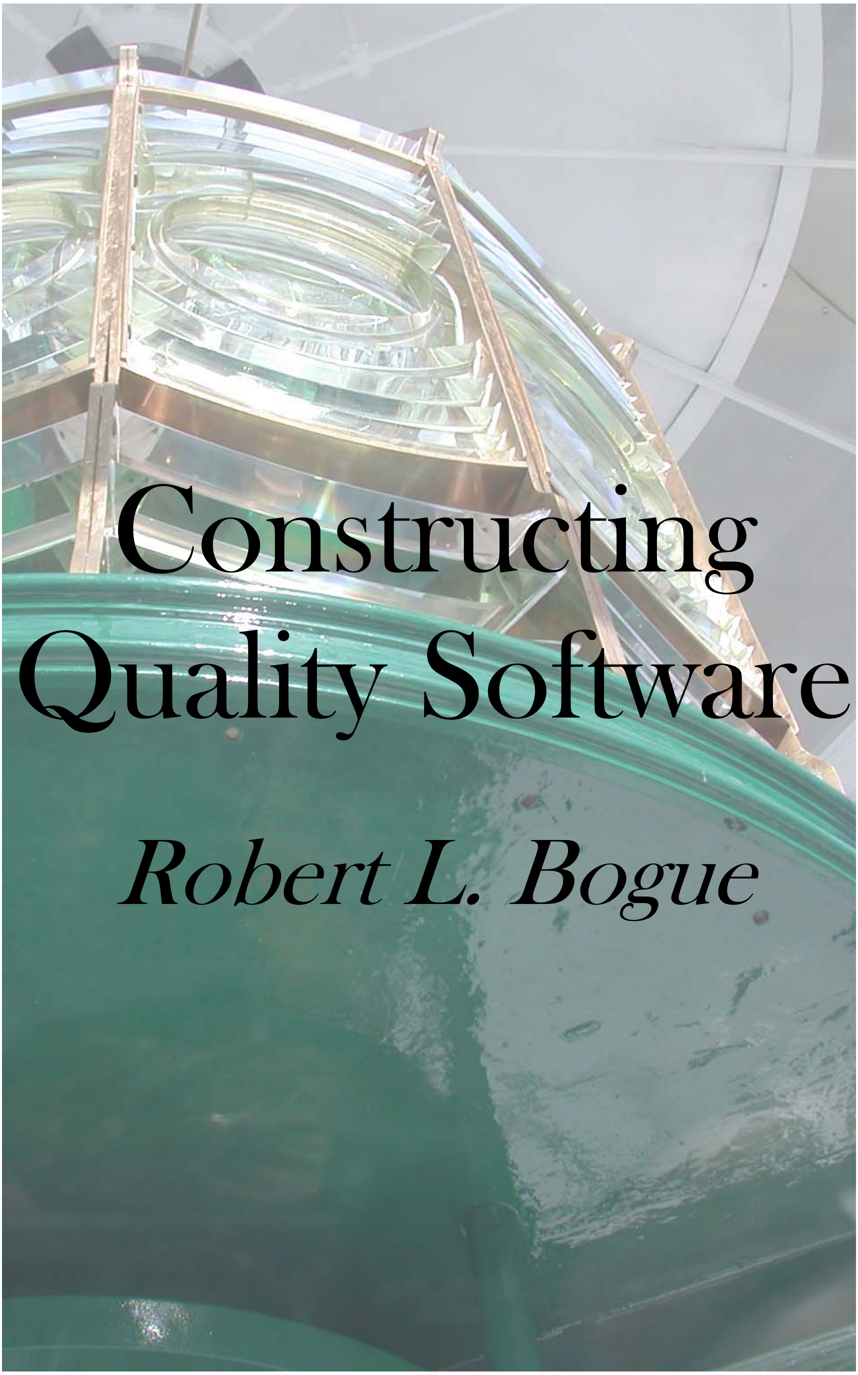


Constructing Quality Software

Robert L. Bogue

Table of Contents

Front Matter i

Introduction.. i

Special Recognition i

About the Author ii

Background on Software Development 1

Standing Out from the Crowd.. 2

Understanding the Differing Goals of Software Development... 7

The Many Faces of a Developer 14

Software Developers—Learn another Language...................... 20

The Best Developers Can Make Poor Architects 26

Software Development Opportunities....................................... 31

Fragile Code.. 32

Handling Exceptions 40

What does Object-Oriented Design Mean to You? 44

Lightweight Objects... 48

The Software Development Process 52

Cracking the Code: Breaking Down the Software Development Roles 53

Anatomy of a Software Development Role: Subject Matter Expert.. 59

Anatomy of a Software Development Role: Functional Analyst 66

Anatomy of a Software Development Role: Solution Architect 74

Anatomy of a Software Development Role: Development Lead 81

Table of Contents

Anatomy of a Software Development Role: Developer 88

Anatomy of a Software Development Role: Quality Assurance 94

Anatomy of a Software Development Role: Deployment 103

Anatomy of a Software Development Role: Training 111

Anatomy of a Software Development Role: Project Manager 118

Anatomy of a Software Development Role: Development Manager 126

What's the Development Manager role? 126

Developing for the Web 133

Design your Web site to be search friendly 134

Developing in Microsoft .NET 139

Create singleton objects in .NET to improve performance..... 140

More efficient coding with advanced user control caching in .NET 154

Application Development: Calling a COM object from a Web service in .NET 158

Front Matter

Cover Photo taken by the author November 2004, Ponce Inlet Lighthouse, Florida

Introduction

What follows is a collection of articles originally written for various publications and on-line sites. The articles have been reedited to fit this book form but are largely untouched from their original form. The articles are selected here because they represent pieces of the software development puzzle.

For many years I've been working in the software development industry and I've been frustrated at our lack of collective progress towards a true profession. There have been exciting projects like the Software Engineering Guide to the body of knowledge (SWEBOK) which have started the process of defining that central repository for software engineering efforts. However, in my day to day life consulting with clients I find that the gap between the state of the art and the state of the industry is so large that it's horrifying.

While I do not expect that it is possible that one more book on software development will radically change everyone's desire to move the industry forward, I feel like it's a necessary step in the right direction towards defining the profession.

I sincerely hope that you appreciate the content here for both its standalone value but also for its value as a collective message.

Special Recognition

I feel that it's necessary to express my gratitude to Brad Jones and Rosemarie Graham. Both Brad and Rosemarie have been partners in developing much of the works collected here. Rosemarie has broken process to publish articles which are much longer than the typical article on developer.com. Brad Jones gave up a great deal of his precious time to personally review many of the articles here. Although he and I did not agree on every point, I believe that the material you see here is better because of his willingness to invest in challenging me.

It is because of both of their support that I was able to create what you see here.

-- Rob

About the Author

Robert L. Bogue has over 15 years experience in the computer industry. During his tenure he's collected recognitions from Microsoft as a Most Valuable Professional for both Windows Servers – Networking and Microsoft Commerce Server. He's also amassed a great number of certifications. (For the interested, MCSE, MCSA:Security, Security+, Server+, Network+, IT-Project+, CDIA+, E-Biz+, I-Net+, A+, and CNA.) He's contributed as an author to 16 books, and as an editor to over 100 more. He currently writes 75 articles or so a year on a variety of topics including software development, Microsoft SharePoint technologies, and IT management.

Robert is a private pilot and lives in Carmel, IN with his wife (Shelley), his son (Alex), and two dogs. He blogs at http://www.thorprojects.com/blog. You can reach Robert via email at Rob.Bogue@ThorProjects.com

Background on Software Development

In this section you'll find a set of articles which provide background information on the software development process. The articles here set the stage for understanding the software development process by examining various specific aspects that are challenging.

Standing Out from the Crowd

Originally published on Developer.com March 7, 2005

The question is often raised about how a developer can become more valuable to potential employers or be recognized by a current employer. The editors at Developer.com and I agreed that there are things a developer can do to stand out from the crowd.

There may be ample positions available today, but that doesn't mean that you'll be able to find a position when you're ready to start looking. Human Resources departments are often inundated whenever an IT position is posted. You need to be able to make your credentials or resume stand out of the crowd somehow — and resume tricks aren't going to do it. At the very least, resume tricks are not going to make you stand out in the interview — the point from which everyone gets offered a job.

Neither will resume tricks help you differentiate yourself with your current employer, helping him or her to understand your value or take notice of your accomplishments. That's where you need to find a way to become unique.

Putting Yourself into a Group of One

Ultimately, the objective in trying to stand out is to create recognition of your participation in a group that has favorable characteristics associated with it. Your participation in the group transfers some of the favorable aspects of the group to you. The more favorable the characteristics and the smaller the group, the more you will benefit.

Take people who have obtained a certification, for instance: it does two things. First, it ensures that certain professional or technical standards have been met. Second, it establishes you as a member of a group. These two characteristics, meeting a standard and being a part of a group, together help to give those reading your resume the impression that you're a member of an elite club. This is in addition to the professional or technical standards that you have mastered. Your goal is to be a member of as many favorably perceived and small groups as possible in order to maximize the uniqueness of your position in the market. It's less likely you'll be competing with someone with the same set of credentials if you are among those belonging to these smaller, more prestigious groups.

Developers can differentiate themselves in a few key ways:

- **Get a certification —** Certification programs exist from a variety of vendors. Microsoft offers MCSD and MCAD certifications for developers. Sun offers a SCJD Java Developer certification. These certifications will demonstrate that you know the languages you're professing to know. Other certifications, such as the IEEE's CSDP Certified Software Development Professional, also exist if you want to show a broader understanding of software development practices. In any case, by becoming certified, you've met a standard and become a part of a group. Of course, it is incumbent upon the certifying organization to maintain high standards for the certification in order for it to be of value to you. It is also important that the certifying organization be known so that others are aware of its value.

- **Get involved with a Users Group —** Users groups, including special interest groups (SIG), are opportunities for professionals to get together and talk. Membership is a great way to show that you're interested in your own development and can be seen as a way of differentiating yourself. If there's not a user group for the development language or development type that you are interested in — start one. User Group leadership positions are a strong statement that you are committed to your own development and the development of others. Presenting to a users group is another way to demonstrate your commitment to the development of others and make yourself standout from the crowd. Participation in a users group at this level can really make you stand out from your peers.

- **Collaborate in a forum —** Become a member of a group of people who speak on forums, public newsgroups, or in other venues. This creates a semi-permanent record of the knowledge you've developed and shows how you're sharing that with others. We live in the age of Google, where it's easy for someone to see the kinds of things that you've been up to. Collaborating in forums leaves a footprint that others are sure to find.

- **Get recognized by a vendor or organization —** Microsoft and other vendors have programs that recognize those that make significant positive contributions to the technical community. In Microsoft's case, it is called the Most Valuable Professional (MVP) program. Due to the relatively small number of professionals in these programs, having a vendor recognize your contributions makes your resume stand out. The MVP program at the time of this writing has less than 2,700 members worldwide. Being a part of the MVP program can really make your resume stand out. Of course, vendor recognition programs don't always tell you how to be recognized. That's why working on all of the other items in this list is useful — they increase your chances of being recognized by helping you stand out in the crowd.

- **Attend regional and national conferences and trade shows —** In addition to the knowledge that you'll gain; you will also demonstrate a desire to stay up-to-date with the latest technologies and techniques. This will help you be perceived as more in touch with what is going on. Learning new techniques and tools helps to broaden your horizons and demonstrates a continuing commitment to learning. Additionally, when you attend a conference, take the time to try to meet people. Most conferences provide the opportunity for attendees to interact with industry leaders and people in high-profile positions.

- **Speak at a conference —** If you think about the audience at a conference, it is filled with hundreds of listeners and only one or two speakers. While being one of the hundreds who attend a conference is a good thing, being the one recognized as the expert to be able to speak at the conference separates you from the others. The process of speaking at a local conference is relatively easy. You need only become an expert in one small area — an area that is interesting to others in your geography. Smaller, regional conferences may not have more than 30 people in a room — no more than most of us had in a speech class at one time or another. Many conferences do a "call for papers". You can respond to these calls by sending in an overview of what you would like to present. In most cases, anyone can submit a paper for consideration.

- **Write an article** — Most people I know aren't thrilled with the idea of writing. The most frequent response when I ask my coworkers if they want to write an article is a groan. However, being published distinguishes your resume from others because there are relatively few people who have ever had an article published professionally. Although getting your first article published isn't easy, it can be within reach if you're willing to make the investment. Even those who have poor spelling and grammar skills (such as yours truly) can become proficient at writing articles. (Editor's note: Writing your first article can be easy. Sites like Codeguru.com focus more on the technical information you can provide rather than on your grammatical skills. As such, it is more likely to get published if you have a valuable technical expertise you can share.)

- **Write a book (or part of a book)** — If the field is narrowed from a floodlight to a spotlight by writing an article, it's laser focused by writing a book or a part of a book. Even though the number of those who have written articles is small, those who have written a book is even smaller. Writing a book shows a more comprehensive knowledge of a subject, beyond just what a single article might show. Writing a book also shows commitment. While an article can be written in a few hours, most books take months to write.

- **Network with others** — Attending a user group or conference, answering questions in forums and interacting in other ways is a start, but take the next step as well — meet the people there. Interacting and networking can lead to connections that will distinguish you. Your ability to develop relationships with relative celebrities can make you stand out of the crowd if for no other reason than making for interesting stories.

Putting it together

The likelihood of you being able to accomplish all these activities, at least initially, is pretty slim. However, they have a cumulative effect. For instance, if you spend just an hour a week working on an article, then within a few weeks you should have a completed article. Keep doing this and within a year you could have over a dozen published articles. I have sixteen books in which I've had author credit, nearly one hundred more that I've worked on, and countless articles. I didn't do all that in a year; rather I did them one at a time. Over time, however, this has added up and now stands out on my resume. In the way, you can flesh out your resume. How quickly this can happen is remarkable in retrospect.

So, take the first step. Find something you can do today to make yourself stand out of the crowd.

Understanding the Differing Goals of Software Development

Originally published on Developer.com November 8, 2004

In preparation for a new series of articles which are forth coming on the different roles within software development, I stumbled across the uncharted territory of how different kinds of companies approach software development. During the process of preparing to discuss various software development roles, I realized that the team dynamics are very different between corporate developers, software company developers, and consulting developers. This article will help you understand the different objectives of software development within organizations.

Before we delve into the goals of each type of organization, it's important to understand that these are generalizations. There are some organizations that don't behave according to the norms that I describe here. The objective is simply to explore how different types of organizations typically have differing objectives and how some of those goals may be detrimental to the software development process.

Corporate Development

Most developers have experience with the corporate development environment. This is because there are so many organizations with developers. If you're working for an insurance company, a manufacturing company, or a healthcare company, you are part of the world of the corporate developer - complete with all of its disjunctions.

In reality there are two kinds of corporations that take part in development: large organizations with structure and processes, and smaller organizations who are trying to develop software with few resources. Let's explore the process of development in a smaller corporation first.

The typical development situation for a smaller corporation is that there is very little support of the process from more senior staff or no routinely enforced software development practices. In smaller corporations, there is generally only one, two, or at most a handful of people who are taking part in the process of software development. The result is a great deal of inefficiency because the development needs are too great for the developers to handle. Although it varies from organization to organization, often times the necessary roles to support a developer are missing from the development process.

One of the ways that this problem shows up is in the lack of clearly defined, detailed requirements. We've learned that taking an organized approach to the development of software is the most efficient approach. Whether you're using a traditional waterfall method of software development or one of the newer, rapid application development methodologies, they all start with understanding the problem trying to be solved. However, because development is not a priority for the organization, there is no perceived need to optimize software development processes by defining clear requirements. It's easier for the business user to simply put together some rough requirements to get the developer started. From there, the developer begins to develop the software. When the program doesn't do exactly what was needed, the business user provides more definition. The developer makes the changes and demonstrates the new program to the internal group asking for the program. When it still doesn't meet the desired goal, the program is modified again and again. The result is a huge amount of effort used to develop a program that could have been simplified by developing clearly defined requirements.

In addition, there is generally a large amount of orphaned code which isn't properly removed from the solution in the rush to meet the needs of the internal customer. This leads into a downward spiral of maintainability problems as described in my article Fragile Code.

Another side effect of the corporate development mentality is that there is rarely appropriate project management control on the software development projects. This may start by the lack of clear requirements and move forward into issues around a failure to tightly control testing and acceptance. Normal changes during the course of the project are generally not well planed for or adapted to. Of course, this has the impact of not only increasing project cost, but frustration as well.

There are often few competitive market pressures and little time to set aside to support process improvement and learning. Standard software development techniques, such as object oriented development, are often not fully integrated into the development processes. In the long term, corporate software developers struggle to remain as effective as their peers. They find themselves suddenly laid off with a need to retool their skills before they can find their next position.

In larger corporations, the development process is more defined; however, there are still problems. The larger the organization, the more systems there are in production. As a result, there are more production problems which must be supported. In the larger organization, processing help desk tickets are routinely routed to developers for investigation and resolution. This practice not only prevents the addition of new features, but also creates a great deal of distraction, which is never good for software development.

On the whole, larger corporations have more bureaucracy than other types of organizations. Without market pressures within the IT space to drive bureaucracy out, it often grows within the IT organization to the point that it can immobilize even the best project. There are architecture groups to get the program approved by, steering committees setting the priority of various projects, and an endless array of middle managers who need to approve a project before it can even start.

From a goals and objectives standpoint, the objectives for a corporate developer of any size are generally focused around simplicity and the desire to make sure that the application can be maintained. Making the application maintainable refers to two things: being able to detect infrastructure issues; and future maintainability of the code by subsequent developers.

The final objective in most corporate organizations is getting the solution out as quickly as possible. There is always a long line of other projects behind the current project; therefore, getting the current project done is always followed closely by the next project on the list. Notice this objective is in direct conflict with the situations noted above, which cause the software development process to be ineffective and require more work to get the same set of deliverables accomplished.

The Software Development Company

Software development within a development company is much more progressive in comparison to other corporations. It does, however, have its own challenges.

By and large, software development companies are driven by the technologies the solution uses. An architecture group may have been a part of the large corporate developer's life, but at a software development company, the architecture group typically includes people at higher levels of the organization. The result is that the architecture hurdle is substantially larger in a software development company. Even a simple decision on architecture can take the form of a month's worth of meetings. Because the technology is perceived to be key to the value the organization contributes, it is given high regard - even to the point of becoming a "sacred cow" which developers know that they can't challenge.

Requirements at a software development company are some of the most rigid to be found in any type of software development. Because there are many clients to be satisfied, the requirements that a developer sees are the distilled requirements created from the various needs of the customers.

The rest of the software development artifacts are generally created from the detailed requirements. Detailed design documents are created to map back to the requirements document. Program specifications are created to ensure the programs being developed match both the requirements and the new design. This, in turn, leads to testing plans. All of these steps help to ensure projects are efficient and effective.

In a software development organization, there's generally more focus on having the right people to do all of the roles needed for successful software development. Because of this greater focus, project management control is generally higher than in a corporate development environment.

However, one of the challenges with software development in a software development company is that the processes, procedures, technologies, and techniques are generally more ingrained and, as a result, changes to the process are more fiercely contested, making integrating new process improvements more difficult to accomplish.

The final component of the software development company is that they typically have a system for measuring customer satisfaction with their products and facilitating the capture of that customer satisfaction rating system. This allows for a continuous feedback cycle designed to push the software development process to improve, despite the natural resistance to any kind of process change.

The Consulting Company Development

My home ground for the better part of my career has been within consulting companies of various types and sizes. While I've worked as a developer in a corporate environment, shared a beer with friends in both corporate environments and software development companies, my home is really in the consulting company.

Software development in a consulting organization is truly a different animal. Typically, a good consulting company will gather good requirements and will maintain appropriate project management control on the project. What gets interesting is how this impacts the projects.

Consulting companies typically focus on meeting the requirements document almost to the exclusion of all other factors. Although there are performance and reliability objectives in a project, they are constrained to the level at which the client is willing to pay. As a result, issues like performance and reliability are only approached to the point where they meet the criteria established in the contract or requirements. Where a software development company or corporate developer might seek to maximize reliability, performance, and flexibility to the best of their ability, a consulting company development project stops once the minimally acceptable level has been reached.

What the minimally acceptable level is for a given project is defined by the context of the organization requesting the project, the consulting organization, and what is written into the contract. Whatever the minimum is between all of these factors will be what the minimally acceptable values are.

However, because consulting organizations must remain flexible to integrate with the software development practices of a wide variety of organizations, they tend to be much more progressive in terms of adapting new software development practices into their process.

A place where the culture can diverge is in the development of reusable components. Consulting companies are faced with developing the same code over and over again for each client. Nearly every application needs code to talk to a database, present a user interface, manage a configuration, etc. In some consulting organizations, they get quite adept at copying or recreating code from one project to the next. In other organizations, a framework of base code is developed which can be used from one project to another.

In the latter case, it becomes easier to deliver true value to the end client because there's a large base of pre-tested code which can be used. This reduces the overall cost of creation and reflects the organization's investment in core technology.

The consulting company, like the software development company, typically has a customer feedback mechanism to help it understand the satisfaction of its customers and allow it to adapt to client needs and expectations. This system drives some of the process improvement that occurs in the organization.

The Contractor

On a parting note, it's important to note that the contractor - or contracting company - will adapt to their environment. The goals of the contractor may never perfectly line up with the organization's needs; however, they will often take on the same basic characteristics of the existing culture of the organization. Because of this, contractors should really be thought of as an extension of your software development process - for better or for worse.

The Many Faces of a Developer

Originally published on Developer.com November 23, 2004

While working on a series of articles, I was tearing apart various software development structures and roles and came to the realization that we call lots of people "developers." Those developers have many specializations. In my last article, I broke apart how organizations have different software development groups. In this article, we'll explore what it means to be a developer and the different kinds of specializations.

A Word About Specializations

Specialization can be a scary word. When the initial discussions began in the 1990s about moving software development offshore, I was thinking about what role, if any, the software developer in the United States would have. I thought specializing made it easier to eliminate me from my current position. I thought if I had a specialization in something that the organization no longer needed, I would be removed. I thought remaining a generalist with many skills would be more beneficial in my quest to remain employed.

What I've learned from years of being on the other side of the fence is that, although there are times when a specialist has to be removed due to a lack of work, there are an equal number of times when having a specialty has proven an invaluable trait for an employee. A person who possesses a specialty can be the right resource to keep: since they're capable and willing to learn one specialty, they may be willing to learn a new specialty - one that hasn't yet been identified as a need. Because of this, sometimes people with specific skills are more valuable than a generalist.

The General Developer

The general developer is the generalist described above. A general developer is someone who has the ability to do any kind of development, but either hasn't yet developed a specialty, or has decided to stay in a mainstream development role that involves a wide variety of different kinds of development. Some developers spend their whole professional lives as general developers without any movement into other roles or development of specialties.

In general, those who fail to distinguish themselves within a specialty after a few years of development work will stagnate within the position for their entire careers.

The Database Developer

The database developer is focused on database access. This developer is particularly adept at creating mechanisms for accessing data in a database. They typically can take the role of a DBA in a pinch because they understand database technology thoroughly and can perform almost any role when it comes to a database. The database developer's ability to troubleshoot performance issues and debug problems can be invaluable.

Initially, the position of database developer may appear to be a dead end job - once you define a database access architecture, what else is there to do? The answer is to assist other developers in the development of reasonable data access patterns, as well as deal with performance issues created by bad queries and indexes. There's an endless supply of new database access situations; the typical developer doesn't have a deep understanding of how a database works or the performance implications of the code.

The Technical Developer

The technical developer is the one who is at home with technical algorithms and the advanced techniques of data processing. Although not needed on a daily basis, the skill to create highly technical solutions is an important part of the developer spectrum. Technical developers fill that critical niche for solving problems that are more complex or that need a higher degree of technical sophistication.

Technical developers are most at home with memory management schemes, tokenization and compression, encryption, optimization algorithms and the like. These needs spring up in some of the most obvious projects; specialization in delivering technical solutions is a rare and invaluable quality.

I was reminded of this in a recent eCommerce project. I needed technical solutions for XML serialization beyond what the language supported inherently, an encryption scheme which allowed us to manage encryption keys for sensitive data, a memory management and thread handling solution that maintained a minimum memory footprint while optimizing performance across multiple threads, and dozens of other smaller but essential components of the overall solution. I handed the problem off to one of our most technical developers who handily met all expectations.

The Business Rules Developer

Expressing the complex set of ever-changing rules of an organization in a programming language or visual designer tool is an art form. This art form is one that the business rules developer has mastered. While reading requirements and speaking with customers, internal or external, they instantly identify paths that have not been explored, potential disconnects, and other logical loops which would make implementing the logic into the technologies available difficult or impossible. They are constantly seeking to understand the rules behind the exceptions and the exceptions behind the rules.

The business rules developer is in constant demand as more and more processes are automated and previously automated processes change. The value of the business rules developer lies in their ability to rapidly translate these processes into usable code. This individual typically must learn more about the business and how it works - an important consideration if one wishes to move into management.

User Interface Developer

The business logic coded into a program can be very valuable; if the program isn't used, then the logic becomes worthless. It's the job of the user interface developer to make the program interface so intuitive, friendly, and aesthetically appealing that everyone will want to use the program. It takes part research, part art form, and part inspiration to develop truly world class user interfaces.

User interface development is often considered trivial, mundane work done by junior or inexperienced programmers. This is sometimes the case; however, some of the most creative and experienced developers are tapped for tasks in programs where the user interface is very important to the program's success.

The Reporting Developer

One of the truisms of information technology is that most organizations are drowning in a sea of data with little information. The reporting developer is one who has become adept at facilitating the conversion of data into information. In addition to learning the somewhat specialized report development tools, they learn how to evaluate the requests of business users and convert the reams of data into usable information.

Two important skills are necessary in any good reporting developer. First, a solid understanding of on-line analytical processing (OLAP) systems is a must. This skill should be present even if the organization doesn't use any OLAP technology - it helps in the construction of reports from a completely different perspective. The second skill is that of aesthetics - not perhaps to the fine tuned level of a graphics designer, but an understanding of the simple rules that make reports more readable.

The Installer Developer

A rare breed, the installer developer is somewhere between the worlds of software development and systems integration. They must work with the developers to learn what the application needs to run, develop their own special program to perform the installation, and work with the systems integration department to determine what is possible in the environment.

Installer development work is deceptively simple. Copy a few files. Add a few registry entries and you're done - or so it seems. Learning how to create MSI files that can be delivered to the organization through automated deployment tools is a challenge that few will master.

Many organizations aren't large enough that this specialty has emerged within their ranks. However, it is one that has its own set of tools, techniques, and knowledge.

The Testing Developer

Extensive testing is performed in larger projects and in those with higher business impact. Eventually there comes a point where the level of testing becomes too cumbersome to execute manually. In those times, a test script, or program, is needed to exercise the program, just as if a user were sitting banging away at it. The test script can repeat the same boring test over and over again to seek out memory leaks, can be run at high quantities to load test the solution, and can be run exactly the same from one build of the system to another to ensure that no new errors were induced. The testing developer specialty also doesn't exist within every organization; however, it is a specialty that can have a huge return on investment when needed.

What Specialties do you want to pick up?

Up to this point, each of the specialties has been examined in a vacuum. That rarely happens. Many developers take their general development skill and layer on it a technical specialty and a database specialty - or a few other specialties; this is perfectly normal and healthy. The more specialties you develop, the more valuable you can be to your next project. Once you've developed a specialty, you have to decide whether to deepen that specialty or branch out into another.

Getting the Opportunity

One of the frequent questions I get is "How do I get started?" The answer to that question is to get the ground work that is necessary to move into the specialty. If you decide that you want to be a database specialist and don't know the first bit about how SQL databases work, you may need a database book or a friendly coworker who may be able to take you under his or her wing and show you the ropes. You'll have to create the opportunity through your desire and drive - don't wait for the opportunity to knock on your door.

Software Developers—Learn another Language

Originally published on Developer.com March 17, 2003

As a high school student, I studied French. I didn't have any great plans to become a translator for the UN, but it was a requirement for the honors diploma that I wanted to graduate with. You had to have a minimum of two years of a foreign language.

During the summer after my second year, the French professor took us to Quebec City, Canada. It wasn't a trip to Paris, but it was much more affordable. Because very few people in Quebec actually speak English, the experience was roughly the same.

I remember that my French was barely passable. I struggled with even basic sentence construction. I'm sure that, somewhere along the line, I said things that were completely unintelligible.

The truly amazing part of this, though, was that the people to whom I was speaking were helpful and we did communicate. The communication wasn't the most straightforward, but it worked. We used a few hand gestures and they allowed me to speak in English if they could respond in French. Generally speaking, we found a common ground that allowed us to communicate as we needed to.

When I think about the relationship that exists between systems integrators and software developers, I am positive that we speak different languages. I am equally positive that some common ground is necessary for us to communicate with one another. Without a basic understanding of each other the communications breakdowns can be catastrophic.

In the world of software development, we focus on languages, frameworks, APIs, and design patterns. Our thoughts are about algorithms and how to solve problems by the arrangement of mathematical and logical operations.

In the world of systems integration, we focus on packets, bandwidth, ports, and vulnerabilities. We talk about five 9s of availability, about managed switches, RAID, dual power supplies, and a series of other terms and technologies that make your head spin.

There's not a lot of common ground here for us to communicate with one another on. In many ways, the software developer doesn't think the same way that a systems integrator does. The software developer wonders how he can build the system. The question isn't whether the pieces are available or not, but just what they should be made of.

The systems integrator wonders how he can fill the gaps left between different technologies and solutions. In many cases, the systems integrator knows the pieces needed to put together a system. The only question is how the pieces will fit together.

This difference in thinking causes us to communicate in different ways. The software developer is always looking to build a solution where the systems infrastructure staff is looking to pull the pieces apart and look in the cracks. Neither approach is wrong. In fact, these differences can help solve the problem quicker if there is good communication; however, good communication is difficult with such radically different perspectives.

It is because we have this difficulty communicating with each other that it's important that all of us try to learn enough about the other perspective to communicate. This means that software developers should understand the basic components of systems infrastructure. It also means that systems integrators should get a primer on software development.

One of the interesting things about my career is that I've done a fair amount of both systems infrastructure work and a fair amount of work on the software development side. In fact, many of my current consulting engagements rely upon this fact. I am all too often called in to organizations where projects are off track and getting worse. Invariably in these situations, the problem is not that there are not enough talented people who know how to solve problems. The problem is that the software development team doesn't know how to communicate with the systems integration team and vice versa.

Anyone who's worked on a complex system long enough can tell you that eventually something will break down, and when it does, you'll have to be able to track down the problem. This is one of the most difficult things to do. This is particularly true when the problem is so complex that no one person has the answer. It is in those times, it becomes critical that everyone learn to communicate with each other as effectively as they can.

How to Learn about Systems Infrastructure

There is no magic in systems infrastructure, just like there is no magic in software development. There are no pills that will allow you to learn everything about systems infrastructure. Because of that, you'll have to accept your learning about Systems Infrastructure by either buckling down on your own by reading books that describe systems infrastructure or by listening to the systems infrastructure staff at your organization.

The latter method is preferred because your ultimate goal is not to learn everything that there is to know about infrastructure. It is, instead, to be able to better communicate. The best way to invite this kind of learning is to simply ask what really happens when you do something at your workstation, or in your program.

For instance, you might ask about the process that happens when you try to log in. To the software developer, it's just a system call that returns a handle to the user's security token. To the systems infrastructure person, it might be a whole series of communications that take place over the network until a server is found that can authenticate the user name and password as given, or no servers can be found.

On the surface, knowing what is happening when a logon occurs may not seem to be that big a deal. However, the next time that you go to write a program that needs to log the user in, you'll consider how the request to log on might impact the network. It also means that the next time there's a problem with your application trying to log in, you'll have more ideas for things that you can test to validate what's truly happening.

Similarly, consider all of the functions that you call from your code, or those functions that are being called through the objects you are using and ask yourself how much you really know about how the system is able to return that answer to you.

In another example, getting the system time is something that we all take for granted when creating timers and loops. How does the system know what the time really is? Windows 2000 comes with support for time synchronization via a Network Time Protocol. How does that work? What can cause it to break down?

In another example, how does your computer contact the server? How is its name resolved? How is the IP address resolved back to a hardware address? How does the network card control whether it's talking to one workstation on the network or every workstation on the network?

In some cases, you may be asking questions that your systems infrastructure staff doesn't know either; but, in many cases, you may find that they can help you better understand what your system is doing under the covers.

Curiosity may have "killed the cat." It is not, however, a bad approach to learning more about systems infrastructure and building a rapport with the systems infrastructure staff at your organization.

What Every Developer Should Know

Despite the fact that there are a limitless number of things that you could choose to know about the network environment that you're software works in, there are some things that you should definitely know about your environment. In particular, you should be aware of all of the dependencies that you're creating with the software that you're developing. These dependencies are important whenever you go to recreate the solution again.

The most common time that software developers experience the need to recreate a solution is when the software is moved from a development environment to a production environment. In most organizations, these environments are separate. The software that was working in the development environment rarely works on the first move to production.

All of the dependencies that were created during the development of the software, including service packs, data sources, DLLs, SDKs, and Registry edits, need to be understood in every development project. These dependencies can also take the form of infrastructure. For instance, if your program needs to resolve names, the program is dependent upon the proper configuration of DNS, WINS, or both.

Without understanding the dependencies that your program requires it is difficult, if not impossible, for the systems infrastructure team to create an environment for your code to live in that works reliably.

I was recently called in to put together some servers for an organization where I didn't do the software development. When I presented them with a question of what they needed I got three answers: IIS, SQL, and the .NET Framework. While it could be that these are the only things that they needed it is quite unlikely. It's more likely that they needed some COM+ packages, the updated MDAC client installed on the IIS box, and potentially even network load balancing. Additionally, these boxes will need to be secured before deployment. It's unclear what SQL client protocol they will be using between the systems—so I'll be hard pressed to truly lock the system down as it should be locked down. Time will tell what was missed on this project—don't let this happen to you.

The Best Developers Can Make Poor Architects

Originally published on Developer.com May 12, 2004

I am frequently reminded of the quote "In a hierarchy every employee tends to rise to his level of incompetence" from The Peter Principal. I am reminded of this as I watch technical people get promoted into areas of management. I have a handful of examples of managers that I have run into who were promoted to (or above) their level of incompetence that I will not subject you to.

Recently, I have seen crystal clear examples where this occurs even within software development. Someone does not need to go outside of software development and into management to be promoted to his or her level of incompetence. There are plenty of opportunities to be promoted to a level of incompetence within the software development ranks.

Software Development is often viewed as having one constant path from the junior programmer to the most senior architect. This is not, in fact the case. Each role is not just a more complex version of the last. In some cases the skills and perspective required for the next role is radically different than the skills and perspective of the previous role.

Still, many of us need constant feeding of our egos. We want to climb the ladder in the software development hierarchy. I, myself, have to find ways to feed my ego in healthy ways that do not get in my way. Many people feed their ego with promotions to the next level in the hierarchy. They do this whether or not that level is a place where they succeed today or where they can succeed in the future.

There are too many people who learn too late that they have been promoted into a position that is beyond their current capabilities and beyond their true hearts desire to become effective at.

Keep It Simple, Stupid

The fundamental rule of any good architecture is simplicity. Good architecture relies upon a few well-used basic principals, which are tied together in the right way. The problem with the best developers is that they are good at so many basic principals that they find that they cannot contain themselves to just using a few basic principals.

They may start out with an object oriented design and add in a recursive algorithm for some hierarchy. There may become a need to improve performance so they reach towards tools of moving business logic into stored procedures in the database. The next problem is a way to represent information in a way that is user configurable so XSLT is added to the solution. Next comes a new programming language and so on. The architecture may perfectly leverage each of the strengths of dozens of technologies making it look on paper to be the best solution to the problem that there could be.

However, the truth is often far from this. The architects, a group in which I include myself, often forget that the developers on their team are not nearly as broadly skilled or even as talented as they are. They forget that not everyone understands how to write a stored procedure. They overlook the lack of XSLT skills on the development team. They delude themselves into thinking that the next technology, practice, or procedure will not make that much of an impact in the learning curve necessary to create the solution.

Further complicating this is that most architects do not view their solution in terms of how it will be supported over the long term. It may be possible to find developers today that have all of the necessary skills, but what about tomorrow? The developers today may be able to understand all of the technologies well enough to develop with them but will they be able to work with the infrastructure team to troubleshoot problems in the production environment once the code goes live.

The best developers do not register complexity at the same level as their peers. There are more things that can fit into the solution before it becomes complex. This is dangerous because each additional technology and each additional approach creates more opportunities for the solution to fail. The more opportunities that there are for failure the more fragile and unreliable the solution will be. Obviously, the more unreliable the solution the less desirable it is.

A Confession

I have a confession to make. Many times I have broken the rules that I have laid out above. In working development projects as well as infrastructure projects, I have allowed myself to create solutions that were unnecessarily complex because I could.

One time in particular that I recall was many years ago. I was a network manager at the time. We had an AS/400 system, which we connected to our LAN. Because of this we used Token Ring. It was just before the time that the Ethernet adapter for the AS/400 was made available. Token Ring worked well for us until we needed to deal with an electrical problem.

For some rather technical reasons it became necessary to use fiber to connect two buildings. When confronted with a price of $4500 to connect the other building with Token Ring, I opted for a cheaper, Ethernet solution, which cost $2200. The solution seemed fine to me. It saved a few thousand dollars in a very tiny budget.

Sure, Ethernet was a different technology that none of the staff in the company knew, however, it was not that hard. It was a different connector, but it was just electrical signals flowing across a wire. What I did not realize is how different it was. I did not realize that the technical staff would struggle with the solution for years to come.

First there was the difference in cabling. Then there were the special quirks to the arrangement of pairs of wire in Ethernet. Then there was learning about broadcast storms. The learning went on and on. Eventually, because Token Ring died in the market, the company converted completely to Ethernet. Despite this, however, they spent many years struggling with having two different kinds of networking technologies.

Making the Transition

Not all of the best developers make poor architects. There is no automatic path of doom when a developer makes the transition to being an architect. There is no automatic path for success either. The pitfalls are hard to avoid, however, there are some guidelines that can make the process a bit safer.

- **Focus on different developer profiles in the architecture.** Each developer operates within a level. They have a certain amount of skill that they can call upon. When developing an architecture pause to think of each type of developer that you will be working with so that you can ensure that the architecture makes sense for each of them.
- **Do not think fancy. Think doable.** Architects tend to look for elegant solutions even when elegant solutions are not called for. Architects tend to think in terms of some grand plan even when there are pieces that do not fit. Focus on solutions that work instead of solutions that are elegant on paper.
- **Seek input from others.** It seems obvious that an architect would seek the input of others as they are new to the role, however, often times their ego gets in the way and prevents

that from happening. Seeking input from other developers and architects is an important part of adapting to the capabilities of the group you are working with.

Software Development Opportunities

This section is a collection of articles which describe particular areas of challenge when trying to construct software. The ideas expressed here are not necessarily state of the art ideas. Instead they are ideas which are not yet state of the industry. In other words, state of the art is what we know to be the best practice. State of the industry is what most people continue to do today – either because of ignorance that a better practice exists or because it is too difficult to change the ingrained ways of doing things.

Fragile Code

Originally published on Developer.com August 7, 2002

Traditional metrics for the development of a software project watch cost, defect rates, and schedule. All are measured, of course, against the project scope document created at the start of the project or at least measured against the expectation that was set when the project was started. While there are still far too many software development projects that don't meet these metrics there are other less visible factors to the long-term success of the project.

We are all aware that most of the time and effort spent on a software development effort is not spent within the design and construction phases. Most of the time and effort is spent supporting, maintaining, and enhancing the system. Subtle differences in the code can make these parts of the software lifecycle painful or relatively pain free.

The Problems

The problem with software is that some systems are infinitely more difficult to maintain than others. The additional difficulty in maintaining the software far outstrips any benefit that is gained by negotiating with vendors, maintaining tight schedules, or any of the other project management techniques that are typically used to control project costs. The testing process, which removes bugs may not necessarily improve the ability to find bugs, accept configuration changes, or make enhancements. The reason for the difficulty in maintaining systems can be broken down into three key areas.

Documentation

System maintainability starts with the ability to understand the existing system and part of understanding the system is a proper set of documentation. Unfortunately, too many systems are delivered with missing or inadequate documentation. Often this is because developers don't know the kind of documentation necessary to maintain a system. They develop documentation that is of little or no value.

Every system should have an architectural document that describes all of the major components, how they fit together, and how they communicate between each other. This helps to form the framework for understanding the application. When this document is missing it means that understanding how the entire system fits together will be difficult and therefore systemic problems may difficult to find. This is the document that helps you understand the forest before you go hunting for a tree.

The next piece of documentation isn't a separate document at all. It's meaningful comments in the source code. The comments that should be in the code include the header comments that describe each function, it's parameters and returns, and comments which describe **why** the code is being used.

Everyone has seen comments above functions, which describe what they do and what the parameters are. These are some of the easiest comments for the developer to generate. They need only cover the standard pattern of name, usage, parameters, and return.

The more difficult, but necessary, type of comments are the ones that explain why not what the code does. The extreme example of this is a line that subtracts one from a variable. A bad comment is the one that say 'Removes one from the variable.' Anyone who can read the syntax of the code knows that.

The good comment is the one that explains why one must be removed from the number. For instance, 'Convert from base 1 to base 0.' That comment explains not what is being done but why it's necessary.

Architecture

No amount of documentation can overcome a bad architecture. The architecture of the application, no matter what the language, has the most profound impact on the ultimate maintainability of the application. Applications with well thought out, flexible, and structured architectures are the easiest to maintain because they are the easiest to understand and the easiest to extend.

Most architectural problems are caused by lack of forethought. Whether the development project was started with no design phase or whether it was a design phase that was excessively constrained, the result is the same. An architecture, which cannot be expanded to support potential new needs, is difficult to maintain because the architecture itself may need to be changed during the maintenance phase. In some cases this can be likened to pouring the foundation after the building has been built. While it's technical possible, it's never the easiest way.

It's also painful from a psychological standpoint that a project that has just recently been completed must be reworked to support what are perceived to be minor changes. The barrier that exists within corporate management is that they felt like they were done with the large expenditures on the project. Selling why it is necessary to re-architect a part or all of an application is a very difficult sale.

Error Handling and Logging

The final area that will determine how difficult or easy a project is to maintain is the error handling and error logging that occurs. When a problem is found the most intensive and time consuming process is determining what the root cause is. This typically takes substantially longer than the process of fixing the problem itself.

Generally problems are simple to solve once the problem is fully understood. Identifying exactly where the error is caused is the key in understanding the problem. Applications that are designed from the start to identify the exact cause of the error are much more likely to be maintainable than software that was not built with this mandate.

One of the things that is most often added to code after it's been delivered is better error handling and logging facilities so that problems can be resolved quicker and easier. Take Microsoft Word for instance. Eight versions of the product came out each one with, arguably, better error handling. Version 9, Word XP, now allows you to send an automatic bug report to Microsoft for analysis. This is a much higher level of error logging that takes advantage of the logging technologies available today.

Fragile code tends not to have good error handling or logging. The tendency is for the application to receive a general protection fault and have the operating system shut the application down. The problem with this is that this doesn't help identifying the cause of the problem. Without modification to add or improve error handling and logging it may be almost impossible to make progress on problem resolution.

Coping Skills

If you've inherited some code that is fragile you'll need a set of coping skills that will allow you to harden the code that you have even if you can't totally re-architect it. The best place to start is in shoring up the error handling, logging, and adding the ability to trace the application.

Error Handling, Logging and Tracing

The best way to help code become more stable is to install error handlers throughout the code. This means providing a place for the operating system to go when the program encounters an error and it means testing conditions that were never tested.

In languages like VB and C++ you have the option of actual error handlers. They allow for a place, within the function, that the operating system can go to when an error is encountered. Even if you're not working with one of these languages you can add some basic error handling. First, you can register critical error handlers with the operating system when the program starts up. This allows the program to receive critical errors rather than the user receiving a generic operating system error.

The key to the error handler is to log whatever information is available about where the error occurred and whatever other conditions can be captured. For instance, logging the call stack and global variables before exiting may provide clues to what happened to cause the error.

The next step in shoring up fragile code is to add testing for all of the parameters passed to ensure that they are valid. The process of adding code to test parameters is relatively noninvasive. Although with any change in fragile code there is some risk, it is the least invasive was to get the most information. By checking every parameter before it enters the existing code you can identify problems caused between functions. Statistically speaking most problems occur between two functions rather than in the middle of a single function.

Finally, adding a set of logging statements that indicate when execution enters and exits a function you can determine what the entire call history is for an application. Obviously, you need the ability to turn off this logging so you don't impact performance when you're not debugging. Think of this logging as a general ledger for an accounting system. It identifies everything that happened. Further more each function that starts should end - just like every credit has a debit in a journaling system.

In some environments the extreme fragility of the code may lead you away from wanting to add the additional statements to support an increased level of error handling and logging. However, in the long term the number of problems caused by adding this additional testing will be far outstripped by the number of subtle errors that are detected and logged.

Documentation Skills

If one of the signs of fragile code is poor documentation then it would stand to reason that one of the ways to help reduce the fragility of code would be to generate documentation for it. Unfortunately documentation must be done with a certain amount of knowledge of the code itself. An amount of knowledge that is difficult to recover once the project is done - and even more difficult if the resources that were used to build the code are no longer available.

The key with documentation is a mixture of automated tools that can convert the code itself into meaningful documentation. For instance, a tool or set of tools that allow you to build the capability of determining where a function is used or a call tree that indicates what functions a function calls.

Automated tools convert the code into useful information about how the solution is architected. The time that can be saved by condensing and converting the code into useful information can reduce the challenges with understanding what the code does. Documentation will continue to be a source of struggle. However, every opportunity to add meaningful documentation should be exercised.

Preventing the problem

Coping with fragile code is a good reactive stance. However, there is also the proactive approach to consider. It's one thing to cope with fragile code but quite another to prevent fragile code in the first place. This involves being aware of the things that result in fragile code during the development phase.

Make Time

The most challenging thing for most professionals is to make time in the schedule of a program being developed to include the necessary "checks and balances" that prevent fragile code from occurring in the first place. With the pressures to deliver code as soon as possible with as many features as is possible it's easy to see how it might be difficult to maintain time in the schedule to ensure that the code is built soundly. It's important to create an awareness of what "fragile code" is and how removing checks and balances increases the risk of creating a system composed of "fragile code"

Set Standards

Before the start of development agreeing on a set of standards for documentation, comments, error handling, parameter testing, trace logging, etc. will simplify the development process and help to ensure that the code has an even level of resilience to problems. It's a simple, easy step that is often overlooked in the rush to get started on a project.

By providing standards it become clear what is expected of all of the developers and causes an increased awareness of the core concepts of software development. This in turn helps to develop better code with minimal rework.

Code Reviews

One of the best but frequently painful ways to ensure that code isn't fragile is to involve multiple parties at every level of the software development process. While it's typical in even the most harried environments to involve multiple people during the architecture phase it's fairly rare for projects to maintain a formal code review process during the development phase.

The code review process is not a punitive process designed to punish those developers who do not possess the greatest skill. It's a teaching tool designed to help all of the developers remember the standards that they've agreed to meet and to learn techniques from one another.

Code reviews need not be long, but they should be done because of their power to help prevent fragile code from being written.

Conclusion

Fragile code is expensive. The additional maintenance costs associated with fragile code will quickly eliminate any gains created during the construction and development phases. Spotting fragile code is easy if you know what you're looking for. Perhaps more importantly fragile code can be prevented.

Handling Exceptions

Originally published on Developer.com June 18, 2004

Although few, if any, software developers really believe that the code that they write is bug free, few spend any time looking at what their code will do when it does encounter a bug, or an unexpected condition. Some developers blindly let the language use its own default error handlers; this is generally referred to as letting the application puke. The other common way to handle an exception is to report little or no information and blindly continue on. However, neither approach is appropriate or helpful.

Most of the money spent on software is not during its development. It is spent during the maintenance phase of a project when responding to slowly changing environmental conditions can cause a wide variety of subtle problems that can be difficult to detect without the proper tools. When handled correctly exceptions can be the key to finding and resolving problems.

Troubleshooting

If your application is designed and written correctly an exception means that a problem has occurred which you did not anticipate. This will generally result in some problem, which the system administrator will be looking to solve. If you choose to eat the exception and never log it or raise it to the user you will be destroying data, which is often essential for the administrator's ability to resolve the problem. Even the most basic exception contains information in it that can be useful to the troubleshooting process. The call stack when the exception was thrown and the type of exception can be invaluable at helping to determine what happened.

By focusing on how problems with the system are identified and isolated will help you to understand how critical proper exception handling is. If you doubt this, wander down to the infrastructure group and ask them which applications they like to support the most and which ones they like to support the least. The answer is often found in how easy or hard it is to identify and resolve problems.

Wrapping

Despite the clear value of any exception, what would be even more valuable is to know all the details about the moment the exception occurred including what data was in use at the time the error occurred, or in other words, logging what data conditions caused the exception to occur in addition to the call log. For a divide by zero type exception it may be obvious what caused the exception. However, in most other situations it will not be obvious at all.

The solution is to wrap the exception that you get into another exception that contains all the information from the original exception but also includes other pertinent data, such as all of the fields/properties from the object. This not only allows you to see where the exception occurred and the kind of the exception but it also allows you to get a glimpse of what was going on at the time.

In a language like C# it generally makes sense to do a try catch block in each function and to serialize the object (this) into a string, use this as the text for the outer exception, and finally include the inner exception which caused the catch block to be invoked. This makes sense because most of the data that a method call works on is wrapped up in the object. This means that within a few lines of code you can dramatically improve the meaningfulness of an exception. Take a look at the following code snipit:

```
catch (Exception e)

{
  throw new ApplicationException(this.Serialize(), e);
}
```

Obviously, the object in question will need to have a Serialize method that returns a string representation of the object. However, in one line of code you have just taken an inner exception and wrapped it up with the data context that it occurred in. A solid serialization routine can even walk down through the object and its component objects so that a very detailed representation of the data objects involved are included.

If you have a method that does take parameters or you need more data you can, of course, add additional information to the outer exception. This involves adding the parameters of the call to the outer exception as well. In C# this can be done by picking up the parameters from the System.Diagnostics.Stacktrace class. Of course, it can be done by hand as well.

Re-throwing Exceptions

You may have noticed in the preceding example that the exception is re thrown to the calling method or function. This is done so that it too can wrap up any information that it can offer into the exception. The end result is that the exception is caught and re-thrown with additional data several times before finally reaching the highest level.

One of the common concerns with this approach is the performance required to generate so many exceptions and raise them to the controlling process. Since exceptions should be used only in exceptional cases where no other normal processing is possible, the additional processing time to handle the same exception a dozen times or so is minimal. The offset to that minimal processing overhead is a greatly improved ability to diagnose problems.

Another common objection to this process is that the exception that is generated will become quite large quite quickly. This is true, however, the number of times that these exceptions are being thrown is minimal thus the memory impact on nested exceptions will be minimal as well.

Logging

Of course, the last step is actually handling the fully wrapped and complete exception at the top level. There are really only a few ways to respond to the exception. The first way is to ignore the exception. If you have spent time wrapping up the exception to make sure that you have everything that you need you are not likely to take this approach.

The second approach is to raise the exception to the user. Because of the sheer size of the exception you may not want to try this approach since you could very easily overwhelm a user. In addition, you may find that there is some data in the objects that could be serialized that you may not want to dump back to the user. You might have internal Ids, which they should never see, or private system administrator comments.

The third approach is to log the error and communicate to the user that the error has been logged. This option, by far the best of the three, records the full text of the exception into a database or other storage mechanism, sends the client that an error has occurred and has been logged, and even notifies the system administrator.

Conclusion

Handling exceptions with care can create systems, which are easy to support without a tremendous amount of work put into creating logging.

What does Object-Oriented Design Mean to You?

Originally published on Developer.com March 17, 2004

In today's world, you're just as likely to see a resume that says object-oriented design as you are to see a bachelor's degree. After more than a dozen years of sitting on a shelf, it seems like suddenly people are interested in object-oriented programming. Perhaps that's because Microsoft .NET and Java languages are all purportedly object-oriented languages. It seems like everyone believes that, if they are using an object-oriented language, they're doing object-oriented design. Nothing could be farther than the truth, and here's why.

Object Oriented Languages Do Not Make Object Oriented Developers

Object oriented languages do not automatically make programmers into object-oriented developers. Saying that I'm an object-oriented developer because I've used an object-oriented language is like saying that, because I have a bottle of wine, I'm a wine connoisseur. (For the record, my knowledge of wine is red, white, and blush.)

An object-oriented language merely makes it easier for you to implement the good programming practices that you already know. You know to minimize the impact of changes through encapsulating your data and code together. You know that you should keep the code similar and minimize the dependencies between different sets of code. An object-oriented language facilitates these and other programming goals.

Becoming an object oriented developer means studying concepts and learning to apply them rather than simply learning a new language.

Objects Don't Make an Object Oriented Design

The funniest thing I have ever heard an interviewee say was, "I've created dozens of objects-yes, and I'm an object-oriented developer." It was funny because it made object-oriented programming sound like an accident. It wasn't something he thought about, but it was something that happened to him without his knowledge. That's about as far from being an object-oriented developer/designer as you can get.

In C#, it is mandatory to create objects. However, that doesn't mean that you're doing object-oriented development or design. It just means you're working within the requirements of the tool.

Object oriented design and development is a conscious thing. It's focusing on organizing information and code into meaningful objects that perform meaningful processes on data.

An Object is

One of the fundamental tenets of object-oriented programming is that an object is modeled after a real world concept. For instance, an object representing a dog might have a tail. The dog might perform the action of walking or eating. The object is modeled after its real world counterpart.

This works well with a concrete object; however, it gets more complex when dealing with abstract objects. For instance, catalog objects have products in them, they have methods that search for products, and they may also have other properties and methods that a physical catalog might not have. When you start to layer the properties and methods, the traditional physical objects begin to lose their focus and have trouble assigning those properties and methods.

One of the keys is to think of how the physical item would behave if it wasn't limited to the physical form. Focusing on the attributes it would take on in an electronic form can help you keep focus on where you put properties and methods.

Coupling and Cohesion

Two classic programming problems are coupling and cohesion. Coupling is a factor that, in general, should be minimized. Coupling refers to the interrelated aspects of different parts of code. Whenever you make one object dependent on another for its operation - or one system dependent on another for its operation, they are coupled. One problem with coupled systems or objects is that, when one changes, it exposes the other to potential problems.

When one object looks inside of another for its operation, rather than using defined interfaces, changes in the internal operation of that object can make the other object operate incorrectly. These subtle interactions between multiple objects can be difficult or impossible to locate.

The second problem is that coupled objects need to be changed together. Even if you avoid the debugging nightmare of not knowing which objects are related to one another, they must still change their function when the coupled objects change.

One obvious exception to the coupling rule is collections. Collections are necessarily coupled with the objects that they contain so that they can return specific type information. They are coupled in other ways as well, as they may need to rapidly create large numbers of instances of the objects they contain.

Like collections, there are other special situations when, used appropriately, coupling isn't a bad thing, and in ways that are clear to even the casual observer, coupling can be a necessary part of software development.

The most closely related concept to coupling is the concept of cohesion. Cohesion answers the question, should an object be one object or more than one object? A cohesive object is one that cannot easily be broken up into multiple objects. It is an object that has one clear purpose and one clear entity that it represents.

It is also applied to those situations when an underlying object is needed. For instance, if you have a dog object and a cat object, you have two objects that have many similarities. Without an underlying object that encapsulates that similar functionality, you risk rewriting the same code. Both cats and dogs are animals and mammals. As a result, a dog object may not be a cohesive object - unless it is derived from objects such as "mammal," which in turn is derived from "animal."

Conclusion

Object oriented design is more than a language. It's not some path that you can follow blindly. It's a winding road that requires concentration. However, there are rewards. Quicker development times, fewer defects in your code, and a more maintainable masterpiece can be yours if you can keep from falling off the path.

Lightweight Objects

Originally published on Developer.com March 31, 2004

One of the disadvantages of object-oriented development is that it can, and often does, mean a more slowly performing application. Unfortunately, this has more to do with a lack of care when designing objects than it does with a fundamental flaw in object-oriented development.

Criteria Balance

First, you must realize that every development project requires balance. No project is without certain drawbacks. For a certain investment in time and money, there's only so much quality, so many features, and so much performance that you can buy. There's no way to have the highest quality software, along with the most features and the best performance, without spending a great deal of time and money.

In any software development project, you must focus on which performance characteristic is critical and make compromises on other less important areas. Of course, if you have unlimited time and money, then you will have more flexibility than those of us with time and money constraints.

In many cases, performance is one of the criteria that is considered last - and in today's world of high-speed processors, that may be the right answer. However, it's not possible to completely ignore performance without eventually having performance issues.

Only What You Need

An economical way to address performance is by creating lightweight objects - in other words, objects that only initialize what they need when needed. An object should be a full-featured way for the user to utilize a specific kind of information; however, providing a full-featured solution doesn't mean that you have to completely prepare all of the features. You can wait until the feature of the object is needed and then initialize it.

For instance, a product object may allow you to view the extended attributes of a product. However, it may not be necessary to initialize those extended attributes if the user doesn't ask for them. This can substantially reduce the amount of time necessary to create an instance of the object. This is important when you need to initialize large numbers of objects, such as when you display an object list.

Patterns for Lightweight Objects

Creating a lightweight object involves an internal field, a sentinel value, and an initialization function. These work together to allow you to selectively initialize parts of the object as needed.

The property is what the user of the object uses to access the functionality. This is so that the call can be intercepted with a small amount of code to determine whether or not the functionality is already loaded.

The way that the property determines if the functionality is loaded is to look for a sentinel value in the internal field that, if present, indicates that the functionality has not been loaded. The sentinel value is set during the constructor for the object or during the object's initialization. Thus, if the field has any value other than the sentinel value, it has been initialized.

In some rare cases, you may not be able to determine the state by utilizing a sentinel value. In those cases, a Boolean flag is used to determine if the functionality within the object has been initialized.

If that portion of the functionality hasn't been initialized, the private initialization function is called. This initialization function initializes all of the functionality - or throws an error if unsuccessful. Once the initialization function has completed, the control is returned to the property, which can then return the internal field.

Picking Properties and Methods

Now that you have a pattern for how to make the object lightweight, it's important to decide which properties and methods you want to make lightweight. When making that decision, it's important to consider how many times the object will be created, the amount of effort required to make the request when the developer-user first asks for the extended functionality, and the amount of time that will be saved.

In general, you need to make a property or method lightweight when the object will be initialized fairly regularly as a course of the application, and the performance necessary to initialize the functionality in the object will take substantial resources. For instance, any remote call to get information, such as a database call, is probably worth making lightweight.

Conversely, objects that are only created once or functionality which won't save much in the way of resources are probably not worthy of consideration for becoming lightweight.

Coupling Where Necessary

A final tactic for making lightweight objects is allowing collections to initialize the object differently than any other user of the object. In general, each object is initialized by providing its key identifying information, and the object loads itself from a database. However, this doesn't take into account the efficiencies that simultaneous retrieval of multiple database records can create. In order to accomplish this, collections are allowed to initialize objects with a record from a query result set instead of utilizing the standard methods of construction.

This technique is allowed from the collection since collections are tightly coupled with the object that the collection holds. In this way, the collection can control the kinds of collections of objects and can allow objects to be initiated much quicker than if you were to initialize each object individually.

This technique can be very powerful since it leverages the native behavior of set based database results.

Conclusion

Object-oriented design doesn't have to lead to performance problems. While performance can be an issue, creating lightweight objects from scratch or adapting existing objects can dramatically improve performance with very little effort.

The Software Development Process

The articles that appear here are the culmination of a series designed to tear apart the software development process and reassemble it. By publicly dissecting the process it is hoped that it gives every practitioner a deeper and more concrete understanding of the process so that they can better work with the people filling the other roles in their software development process.

Cracking the Code: Breaking Down the Software Development Roles

Originally published on Developer.com March 22, 2005

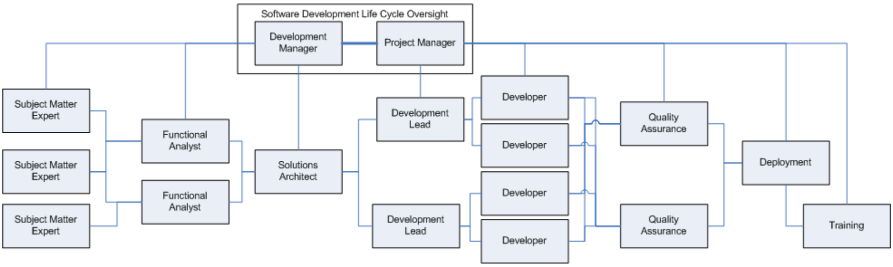
Software development is done differently at every organization, and in every home office throughout the world. The process that one organization or person uses to develop software may work for their specific environment and situation but may fail miserably in another set of circumstances.

It is, in part, the differences in environments which make it so difficult to quantify the process of software development in a single set of terms that all practitioners can agree to. As newer approaches appear on the scene, such as extreme programming and agile development the perspective of the world on what the process should look like changes slightly or dramatically.

However, despite these changes there are some things that remain the same. There will always be a need to understand the business problem, convert that problem into an architecture, convert the architecture into a solution, test the solution, and deploy the solution. Although each of these processes may change to some extent based on the programming models and tools being used, fundamentally there are some roles, which every process has in one form or another. One person may be filling all the roles or a handful of the roles, or one very specific role. Despite this there is a need for all of the roles - each serves a purpose. The organization chart below gives you an idea of how each position fits together within an organization.

Common Roles

There is a series of roles that exist in most software development processes. As mentioned above one team member may be filling many roles and some roles may be suppressed for a specific type of project but all of these roles exist in one form or another in every software development project:



- **Subject Matter Experts (SMEs)** - The subject matter expert is the person or persons from which requirements are captured. These are the people who know what the software needs to do and how the process works. The SME role is somewhat different from the other roles because it is constantly changing as new clients (internal or external) are brought in to help design a solution. SMEs are rarely from IT - except when the solution is being designed to support IT. SMEs are most frequently the person who will receive the benefit of the system.

- **Functional Analysts (FAs)** - Functional analysts have the unenviable roles of eliciting clear, concise, non-conflicting requirements from the Subject Matter Experts who may or may not understand how technology can be used to transform the business processes in a positive way.

- **Solutions Architect (SA)** - The technical architect is responsible for transforming the requirements created by the

Functional Analysts into a set of architecture and design documents that can be used by the rest of the team to actually create the solution. The Solutions Architect is typically responsible for matching technologies to the problem being solved.

- **Development Lead (DL)** - The development lead's role is focused around providing more detail to the Solution Architect's architecture. This would include detailed program specifications creation. The Development Lead is also the first line of support for the developers who need help understanding a concept or working through a particularly thorny issue.

- **Developer (Dev)** - The heart and soul of the process, the developer actually writes the code that the Development Leads provided specifications for.

- **Quality Assurance (QA)** - The quality assurance role is an often thankless position that is designed to find bugs before they find their way to the end customers. Using a variety of techniques ranging from keying in data and playing with the system to formalized, automated testing scripts the Quality Assurance team is responsible for ensuring the quality of the solution and it's fit to the requirements gathered by the Functional Analyst. Sometimes the QA team is known by their less flattering name of testers.

- **Deployment (Deploy)** - The deployment role is the one which packages up all of the compiled code and configuration files and deploys it through the appropriate environments or on the appropriate systems. The deployment role is focused on getting the solution used. To that end the role may include automated software installation procedures or may be as simple as copying the files to the appropriate place and running them.

- **Training** - The training role is responsible for documentation for the system as well as any instructor or computer based training solutions, which are designed to help the users better understand how the system works and what they can do with it.

- **Project Manager (PM)** - The project manager is responsible for ensuring consistent reporting, risk mitigation, timeline, and cost control. The project manager role is a problem solver role. They try to resolve problems why they are small so that they can be handled more quickly and with less cost.

- **Development Manager (DM)** - The development manager is responsible for managing multiple priorities of conflicting projects. The Development Manager role is also an escalation for issues from the team, which it is unable to resolve internally.

Of course, each organization has it's own take on these roles; however, these are the roles you'll see most often in an organization doing development.

Over the course of the next few weeks, articles will be published that delve deeper into each of the above roles.

Critical Skills for Every Role

In the articles describing each role there is a section, much like this one, which is designed to support a bulleted list of items that are critical to the success of the role. During the creation of the series a common set of skills was identified that were essential business skills that professionals in nearly every role needed to consider during their career. Rather than repeat them within the individual articles describing each role, they have been brought together here so that you could consider the impact of these roles in whole no matter where you are in the software development process. The common skills to all roles are:

- **Understanding Business** - Although some roles are focused very specifically around certain aspects of understanding and converting business requirements, every role in the process should have an awareness and sensitivity to the business processes and needs which require technology in the first place. Without this technology may be implemented but it may not solve the real needs and will therefore be considered a failure.

- **Broad Understanding** - Although an understanding of software development is critical there are other areas where an understanding can be invaluable. For instance, understanding how computers work internally including memory, cache, hard drives, etc., can help you learn how to more appropriately conserve those resources. Similarly understanding networking can help in the development of applications, which are compatible or even friendly to the networks that they're working across. SMEs broad understanding of the industry can be invaluable in terms of creating solutions that fit both the organization and the industry. The QA team can benefit the project by a broad understanding by minimizing QA costs while improving testing coverage. In short, a broad understanding can help every role.

- **Multiple Perspectives** - The ability to approach solutions from multiple perspectives is critical to software development. Understanding how each person who is working on a problem views an issue - or how different customers will view the solution is important to be able to find the best solution based on all of the information. There are always multiple ways of viewing - and solving - a problem. The trick is to find the best one from the list of possible options. The larger the list of options (perspectives) the better the solution.

- **People Skills** - Also known as soft skills, the ability to interact with other people and to be a part of a team is essential to nearly every role in a software development project. The lower the overall people skills of the team the higher the likelihood that the project will end in some explosion.

- **Lifelong Learning** - Although some might argue that the perspective of being a life long learner is more of an attitude than a skill, it is a critical part of being in a high-change industry like IT in general and software development specifically. What is learned today will be obsolete tomorrow. The only way to stay ahead of the game is to approach life from the perspective of continuous learning. Each new experience is a new opportunity to learn and each new year brings with it the need for skills renewal.

In the next article, the first detailed review of a role in the series, we'll delve into the role of the SME in the process and what specific skills can make an SME stand out from the rest.

Anatomy of a Software Development Role: Subject Matter Expert

Originally published on Developer.com April 12, 2005

In Cracking the Code: Breaking Down the Software Development Roles I gave you the 50,000 foot view of the human side of the software development industry and the various roles involved. Here and in the articles to follow I will provide you with details on each of the key roles.

The role of the subject matter expert (SME) isn't so much a role that the core information technology person plays normally, however, the role is an important part of the development process. Sometimes subject matter experts are business owners or business users. In most cases they are most often called "Client" Or "user". Figure 1 give you an idea of where the subject matter expert fits into the bigger picture of an organization.

SMEs are the people in the process who provide the information on what needs to be built. They serve in the most important role in the development process - despite not being a part of the permanent development team.

Without the subject matter expert there would be no need and therefore no development.

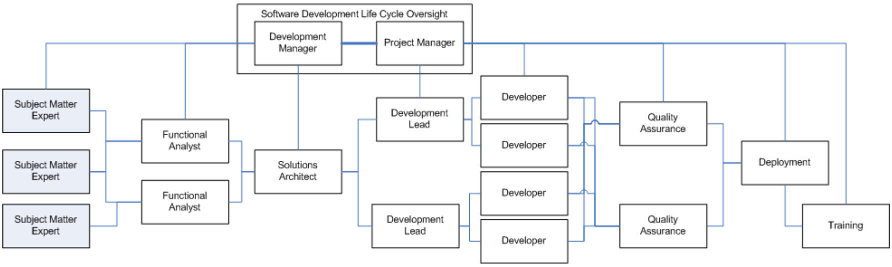


Figure 1: The SME in the larger scheme.

What is a Subject Matter Expert?

Subject matter experts really fall into a few categories. The first category is the business owner who initiates the development process. For internal development this might be the manager who is sponsoring the project. For external development projects it might be the customer paying the bills. For software development companies the SMEs are most often the users of the software who understand what the product is supposed to do the most or at the very least what the product is expected to do.

SMEs provide all of the raw material for the development process this includes the requirements for the system and how it will be used. Their input describes the problem or the opportunity that the software solution will ultimately solve.

Who is a Subject Matter Expert?

SMEs are perhaps the most broadly described part of the development process. Because they can come from all walks of life, all levels of awareness of the software development process, and all levels of interest, trying to describe them is a futile process.

The most defining characteristic of a SME is the fact that, with few exceptions, they won't understand the software development process. They are not a reoccurring part of the process and therefore they won't have experience with what is happening-nor should they. This lack of understanding regarding the process is not a critical limitation, because the SMEs will work with a Functional Analyst who will guide them through the process. It is the FA's job to understand the information that the SME has to share and to guide them through understanding how the process will work.

What Isn't in Their toolbox?

Unlike most roles, which bring extra skills to the table, the SME removes some of the inherent skills that other members of the team possess.

As a part of a development team, here are some skills that you should be careful to avoid assuming a SME has:

- **Don't assume that the SME will clearly communicate what they know.** Most people aren't good at clearly organizing their thoughts on a topic and communicating them in detail. The expectation that the SME can turn on a faucet and start spewing out information in a way that the rest of the team can understand it is unrealistic. Set expectations should be that they wouldn't be able to simply drop their knowledge directly into a document or even into a discussion. It should be expected that things will be left out, some processes may be misunderstood, and that an individual's idea of current priorities might not match that of the company.

- **Don't assume that an SME will understand models Architects, Development Leads, Functional Analysts, and other members of the team often develop fancy diagrams to describe complex program structures.** These structures are readable because the development team has learned how to read them over time. The SME may or may not be able to read technical diagram. Because they are largely outside the development process, they shouldn't be expected to read such documents. Another way to think about this is to think about it is to think about reading a blueprint for a building. While the basic structure may be understandable from a blue print, not everyone will be able to fully understand the meaning of all the lines and numbers on the document.

- **Don't assume that a SME understand or can use defect logging and tracking systems.** Some SMEs won't have the skills to even track the discrepancies from what they want. They'll need help getting their knowledge into the systems used to drive the software development process.

- **Don't assume that a single SME has all the answers.** Each person has their own perspective SMEs are no exception. You may need to work with dozens of SMEs to reach a single consensus based on the amount of knowledge the SME has, the scope of the solution you're trying to create and the lack of consensus in the industry or organization.

Where's the Role Heading Anyway?

There will always be a need for subject matter experts - particularly those who can clearly articulate the needs the organization faces. Although SME is not the primary role that an Information Technology person typically fills it's one that can be a great asset for an organization.

If a SME shows particularly good skills at articulating the business needs then perhaps there's the opportunity to take a part-time or full-time role as a functional analyst (FA). The functional analyst's job is to create clear, precise communication and to support the SME role. This combination makes it a natural path for those SMEs who become hooked on the software development process and want to be more involved.

Of course, the path is not paved in gold. The SME will need to focus on being able to document details, extract information from other SMEs, pay attention to the details, and in general become a more integrated member of the software development process. SMEs who want to make the leap to being a functional analyst will have to accept a greater level of process and technical knowledge while leveraging their familiarity with being an SME and the struggles that a SME has in working through the process.

What Makes a Subject Matter Expert Stand Out?

Being a subject matter expert isn't a career in the same kind of way that being a developer is a career. In most of the roles in the development process the core learning is around the skills and technologies of developing software. The SME is instead developing a deep understanding of a process, an industry, and in some cases an organization. The SME's value is their unique understanding of the problem that the development process is designed to solve or at least help resolve. In this way a SME is focused on being the thought leader and expert for a small set of information.

SMEs stand out from the crowd when they deliver industry presentations that call attention to their complete understanding of how the industry - or one part of the industry works. This process requires a willingness to get in front of large groups to speak and the drive to develop presentation skills that are very good.

In addition SMEs can cause themselves to stand out by writing articles for trade or industry journals. Writing an article is great in itself because it requires a certain level of clarity around the topic being written about. However, the real power is in being published in an industry magazine because there is an implied branding for the kind of quality of person who writes articles magazines. This can be immensely powerful in making you stand out from the crowd.

In a less public way the SME can stand out from the crowd by learning how to interact with different personalities to develop a network of relationships in the organization or industry that they are working in. It is rare for an SME to clearly understand the challenges faced by the producer for the organization, the sales department, the executive staff, and all of the other various departments. The more that an SME understands about the operation and hurdles facing the organization the more valuable they are in their organization and as an SME.

The Good, The Bad, and the Ugly

The Subject Matter Expert role in the software development lifecycle has its ups and downs just like every other role within the process. Here are a few examples of what's good about the role and a few items to watch out for:

- **Good:** The role in the project is generally short lived. Projects tend to come and go

- **Good:** Subject matter experts are a generally well-respected and necessary part of the project.

- **Good:** Subject matter experts have a chance to interact with numerous people at all levels within an organization. This is often great exposure for being noticed within an organization.

- **Bad:** Since software development isn't the primary process of an SME most feel a bit like a fish out of water.

- **Bad:** Although generally bright intelligent people the rest of the software development team may have trouble understanding the business that a SME is describing since they've not been a part of it. An SME may have to explain things from a couple of points of view for it to be fully understood.

- **Ugly:** Participating in a software development process may require more time than you're used to.

- **Ugly:** SMEs may have to interact with geeks and bear through discussions on topics that won't ever help them in their daily jobs.

Conclusion

The subject matter expert is the genesis of the software development process and can be an invaluable member of the team. Because their involvement in the software development process is short lived there is a role to guide them through the process. That role, the functional analyst, is also the next step up for a SME who's looking to become more involved in the often chaotic process that is software development.

Anatomy of a Software Development Role: Functional Analyst

Originally published on Developer.com April 26, 2005

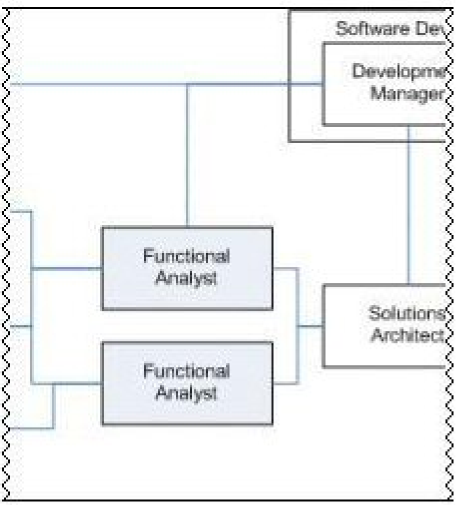
The role of Functional Analyst is one of the keys for successful software development. In Cracking the Code: Breaking Down the Software Development Roles you get a high level view of the software development industry and the various roles involved including that of the Functional Analyst.

The role of the functional analyst (FA) is to capture, consolidate, and communicate the information from the Subject Matter Expertss (SMEs) to the rest of the team. This may seem odd if there's only one Subject Matter Expert; however, the typical case for a sizeable software development project is that it takes several SMEs in order to provide the necessary information to create a solution. Because of this the Functional Analyst is a critical link between the Subject Matter Experts providing the business requirements and the rest of the team trying to construct the solution.

Depending upon the organization that the software development is being performed in the functional analyst title may be called by other names as well. Another very common label for the FA is Business Analyst, or sometimes simply analyst. No matter what the name, the need to help capture, consolidate, and communicate the information from the SMEs to the rest of the team is the critical, bridge-the-gap, role that this person plays. An organizational chart gives you an idea of how each position fits together within an organization.

What's a Functional Analyst?

The Functional Analyst is the facilitator for the Subject Matter Experts. The FA takes their input and transforms it into something that the development team can understand. One of the key components of this is clarifying the intent of the SME. The FA will spend a great deal of time asking questions like "What do you mean by that?" or "How does this fit in with what we were talking about earlier?" Questions like these expose potential, often subtle, differences in meaning between the SME and the rest of the development team. More importantly, these questions expose assumptions regarding business logic and processes that may not be clearly stated - or even stated at all by the SMEs.



FAs are also responsible for identifying and resolving conflicting requirements. If SME number 1 says the sky is blue and SME number 2 says the sky is red, it will be the FA's responsibility to resolve that discontinuity. This may be done by getting the SMEs together to agree or merely in understanding the different perspectives. For instance, perhaps SME number 2 was referring to the sky on Mars. A less abstract example would be if SME 1 considered the way to uniquely identify a customer to be using a person's social security number, where as SME 2 considers a customer to also include people from outside of the United States who don't necessarily have a social security number. If uniquely identifying a client is important, then it is the FA's responsibility to identify this and document the requirements.

Through the software development process a document or set of documents are being developed. These documents, the requirements documents, will represent the contract between the business that wants a solution and the software development team that wants to create the solution. A requirements document is, at the basest levels, a listing of all of the features or aspects that the final solution should have to fully solve the problem that the SME is describing.

The documents need to be understandable both by the SME and the development team. The SMEs will need the document to validate that the requirements for the project are correct in every detail. The development team needs the document so they know what is to be built. To accomplish both objectives the documents must be brief but thorough. They must also be expressed in both business and technical language. Done correctly they are the perfect balance between competing forces.

Getting Started as a Functional Analyst

In Real Estate it is said that there are only three things that matter: "Location, Location, Location." In that is a simple truth that sometimes there is one key attribute that drives all the rest. In the case of the FA that one attribute, or skill, is clear communications. So the starting point for a functional analyst is becoming a great communicator. Unlike Ronald Regan the objective in becoming a great communicator is not inspiring people with a vision. It is, instead, developing precise communication that will allow discovery of inner meaning and inconsistency that is essential.

Learning these skills is best done by working in positions requiring either leadership or detailed documentation. Leadership positions in professional or community organizations is an obvious target for training for the FA role, however, those roles require so many more things that they can often be distracting from the core skill that the FA needs to develop. A better role is the often-neglected role of secretary. While the obvious thought here is that the secretary is simply someone who is taking notes, the role can actually be an opportunity to safely develop the core skills for the FA role. Another benefit is that this role is rarely as contested as the role of leader of an organization.

Precision communication in most professional or community organizations isn't a overt requirement. It is often lacking because of the volunteer nature of the organization; however, most organizations appreciate the addition of the skill. The ability to clearly articulate the situations faced by the organization through precision communication is an immensely powerful tool for helping the mission of the organization. This being said, community organizations are often a safe place for the FA to learn this skill. The secretary role puts the potential FA into the position of being enabled to ask the detailed questions. As the note taker it's generally acceptable to others to ask the questions like "Can someone summarize what we decided?" or other questions that drive towards validating a common understanding. In addition questions such as "Can we clearly define what we mean here so that there are detailed notes of our commitment?" can be asked to drive into more precision communications. This is the exact kind of behavior that the FA will need to use to elicit clear communications from SMEs.

Building a requirements document is usually a key responsibility of the FA. Creating a good requirements document requires an insight into knowing when detail is necessary and when additional detail would only serve to clutter up the understanding. Just as there is no one recipe for creating cookies, there is no one formula for creating an awe-inspiring requirements document. In many ways creating a good requirements document is as much of an art form as it is about the science of capturing specific, numbered requirements.

What's in their Toolbox?

The functional analyst's toolbox is, first and foremost, a toolbox filled with communication and relationship skills. Although the FA has a set of technical skills, their greatest asset is their ability to communicate with others and to work relationships with others in the organization who can help to weave the SME feedback into context.

The FA's technical tool box is not extremely specialized, they largely have to use the same tools as other members of the software development team, however, in their case they sometimes have to be more skilled at the basic word processing, spreadsheet, and general office tools' use in order to support their deliverables.

One of the most important tool to know how to use is a word processing program. An FA may need to know how to use a word processing program at a more advanced level that others on the development team. The FA might need to create templates, develop large documents, and clearly convey requirements in a written format. The FA will need to understand the use of styles in the word processing program to support consistent formatting throughout the document. The role also needs to know how to create indexes and tables of contents so that the documents they produce can be useful reference documents as well as a readable introduction to the problem.

Another tool that the FA might need to understand and use is a spreadsheet tool. They must be comfortable collecting, organizing, and combining a variety of lists including mapping requirements to the design feature points created by the architect and the development leads. They must further then help to map requirements to the testing scripts created by the quality assurance role. An understanding of how to manipulate data in a spreadsheet facilitates all of these mappings.

The last tool I'll mention that is in the toolbox is a drawing program, such as Microsoft Visio, which allows for the definition of use cases, process flows, and other diagrams that would be nearly impossible to express in words alone. The ability to use the program to accurately depict a wide variety of complex ideas is essential to crossing the chasm between the SME's knowledge of the problem and the ability for the software development team to solve the problem. The generally accepted practice for diagramming is becoming UML (Unified Markup Language). UML allows you to describe relationships, states, and other common requirements that are best expressed in a graphical way in a standardized way.

Where's the role heading anyway?

There was once a time when people predicted that everyone would be able to write their own software. They would sit down at a computer and just tell it what they wanted. The computer would write the program from this dialog and thus developers in their current incarnation would no longer be a necessity. This vision is all but gone from the heads of most practitioners. As more became known about what people wanted to do with computer and the rate of growth of their demand it became clear that there would always be increasingly more complex problems to solve.

A part of that realization is the realization that our ability to accurately describe the problem determines the ability for the problem to be solved. Most people are incapable of clearly and precisely articulating - to the level necessary -- the problems that they're trying to solve. This is a problem that is getting larger and not smaller.

This is the very problem that the functional analyst role has been created to solve. The functional analyst's goal is to refine the understanding and communication from the subject matter experts and convert that into the clear, precise vision necessary to create a solution. Because of the growing need to automate in order to be competitive and because of the increasing difficulty for clearly articulating true business needs, the functional analyst's role is more important now than it ever has been.

Standing Out in the Crowd

One way to stand out from the crowd as a FA is to become adept at dealing with differing opinions and conflict. The ability to gather a room full of disagreeable people and getting them to agree is a skill that is difficult to master but one which will show its value relatively clearly. Disagreements about what the problem really is - is common between SMEs. Sometimes SMEs are able to work the problems out amongst themselves. When they can't, it is a powerful FA who can jump in and work through the problem.

Another way to stand out from the crowd is to develop requirements documents that are clear, precise, concise, and meaningful. The ability to develop requirements documents that are clearly understood by the entire software development team is rare even in a role where the primary purpose is to develop requirements documents.

A FA will also stand out when they show that they are keen to listening rather than more focused on presenting solutions. It is not the job of the FA to provide the solution, but rather to document and connect the requirements. SME will recognize a functional analyst that is more interested in hearing from them on what the requirements are, rather than one that presumes to know the answers already. By listening to the SMEs and by taking note of what is being said, an FA can build relationships that can help them in the future.

The Good, the Bad, and the Ugly

As with any role there will be good with the bad and then there will be the really ugly. The functional analyst has the opportunity to set the software development process on the right path by carefully controlling how the process gets off the ground. There are the key points for the role:

- **Good:** Key role in the definition of the solution. Being at the start of the process the FA has the greatest opportunity of any member of the software development team, to get the project started in a way that will create the best solution.

- **Good:** Interaction with everyone An FA has the greatest opportunity to interact with everyone on a project. This includes people on the development team as well as people outside the development team. Often this can include higher-level people within an organization. Such exposure to can be great for building a positive reputation and a strong career.

- **Bad:** Not all SMEs are created equal The quality of SMEs that a FA must work with will vary greatly. Some SMEs will make the FA role easy and others will make the FA want to commit acts of violence.

- **Ugly:** Conflict For most FAs conflict will be a normal consequence of daily work. This can get downright ugly at times - resembling a session of the Taiwanese congress.

- **Ugly:** All fingers often point to the FA. If the FA does their job, then everything should work out. If something is found to be missing from the solution, then the FA is often the role that is blamed first.

Conclusion

The Functional Analyst can be described as the "bridge across troubled waters." Their role is to bring together two very different perspectives. This process is done through conflict management, listening, and clearly communicating. The true armor that the FA has against the slings and arrows that will likely be thrown at them is the armor that they build in the development of a rock solid requirements document that accurately captures and communicates the vision of the SMEs and removes areas of potential misunderstanding by illuminating the dark crevices of detail that hide in everyone's vision of a solution.

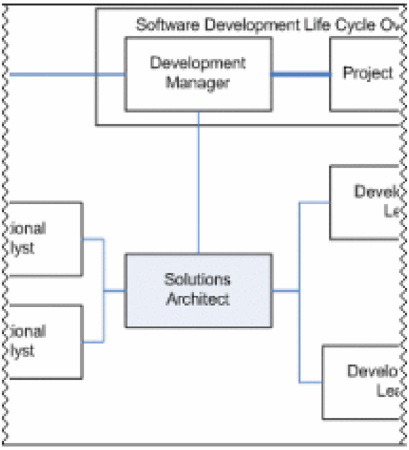
Anatomy of a Software Development Role: Solution Architect

Originally published on Developer.com May 12, 2005

If you've been following this series of articles you know already that every role in the software development process has its unique requirements.

For many developers perhaps the most sought after role is the role of the Solution Architect. The Solution Architect is the person who organizes the development effort. They are responsible for the vision that underlies the solution and the execution of that vision into the solution. The Solution Architect becomes involved with a project at the time the Functional Analyst (FA) is developing requirements. They then remain involved throughout the balance of the project.

What is a Solution Architect?



The essence of the Solution Architect (SA) role is the conversion of the requirements into an architecture and design that will become the blueprint for the solution being created. This conversion is based largely upon the previous design patterns that the SA has been involved with in the past through reading and staying abreast of the latest techniques, or through personal experience.

It is this conversion part of the role -

the role of the SA -that most often is underestimated in its complexity. Just as the ability of the Functional Analyst to create a requirements document is one part science and wrote procedure so is the creation of the architecture. The rest, however, is an art form. Creating effective architectures to create a solution requires the careful balance of dozens of development concepts ranging from "Keep it Simple Stupid" to "Fail to Safe".

In the process of converting requirements to an architecture there are often parts of the SA's role which seem out of place. For instance, there is often a fair amount of research that happens during this phase. The research may be targeted at testing a technology that will become critical to the architecture. For instance, the SA may test to see if USB or serial port access is available from Java if there's a need to read a device without downloading software. This process can either be done alone or depending upon the size and velocity of the project can be delegated to a development lead.

In addition to research on technologies and approaches critical to the architecture, there is often a review of patterns that might be useful to the architecture. Patterns are previously described and validated approaches that can be used to create portions of the solution. Patterns are released through research and can come from places such as Microsoft's software development libraries. Reviewing the pattern allows the architect to refresh their memory on the details of the pattern and to evaluate what additional guidance they will have to provide if they choose to use the pattern.

The final component to the role of solution architect is the motivation and guidance of the development leads. Development leaders need to buy into and accept the architecture, to know how the pieces will fit together at a high level. They must also see the art portion of the architecture to get an appreciation of the subtle nuances of their portion of the architecture. It's the art portion of the architecture that makes it elegant. That elegance helps to maintain cohesion between various parts of the design and encourages simplicity. It is necessary for the lower level design and approach to match the higher-level architecture for the solution to be cohesive. Once the development leader has internalized their portion of the architecture the SA must continuously motivate and reinforce the good work that is being done. They must continue to motivate the Developer Lead(s) to push through tough issues and create the solution.

Getting Started as a Solution Architect

For most people becoming the SA on a large project doesn't just happen. It's not like winning the lottery where one day your name is drawn out of the proverbial hat. It is, instead, a slow steady progression of learning and developing. A person may find their way to this coveted role within only a few years of professional experience but more frequently it takes a dozen or more years to consistently find themselves in this role.

The starting point is generally being the only person on a very small, and sometimes insignificant project. The project may be small enough that a single person may fill every role - including the role of solution architect. These little projects can even be ones where the organization hasn't identified the project as something that needs to be done yet but are items that a member of the software development team realizes that would be helpful.

Another approach to becoming a SA is to become a distinguished Development Lead (DL). The SA role and the role of the DL are very similar in the skill sets needed. The SA skill set is slightly broader and requires a bit more finesse, however, fundamentally the same. The SA lays out the architecture for the overall solution whereas the DL converts that architecture into detailed design. One approach to getting started as a SA is to become a DL and work towards the additional skills that a SA possesses. Most SAs have that ability to give some of their work to DLs looking to step up.

One of the ways to demonstrate an interest in the SA role, no matter what role you may currently be filling is to invest time in learning patterns. Because patterns form the basic building blocks of nearly every architecture, learning patterns makes it far easier to identify where they can be helpful. Also, reading books and articles on different architecture perspectives and new development techniques can broaden your point of view and allow you to see opportunities to create your own small sections of the solution.

The distinction between a development lead and the SA are often subtle. Where the development lead focuses on detailed knowledge of a particular area the SA is very broad. This allows the SA to view the problem from a different perspective. Instead of getting mired down into the details of implementing one specific thing the SA focuses on integrating various parts of the solution into one cohesive network that solves the larger problem.

The other subtle change is in accountability. While the development lead is responsible for their part of the solution, the SA is the proverbial one neck to choke if it doesn't all come together right. The SA has the ultimate responsibility for making the technologies work together. As a result the SA role comes with a requisite level of responsibility for the success of the project.

What's in their Toolbox?

The toolbox of a SA has more tools in it than most other roles. Most SAs have grown up in the software development world and have learned dozens of tools designed to help them be more productive.

Perhaps the most important tool in the toolbox is a visual documentation language, such as UML. The UML structure for describing a variety of different views of the software development problem in pictorial form is the most recognizable visual documentation language for developers. The SA should be familiar with each of the various UML forms and have expertise in the development of use cases, class diagrams, and occasionally state diagrams as well. Mastery of UML allows quicker, easier, and better communication with the DLs and the developers.

In addition to UML, the SA may need to be good at database design. Because most of the time the way that data is stored and retrieved is integral to the success of the solution, knowing how to design a database to hold the information is a critical part of the solution being successful. SAs know how to create databases and optimize them for good performance.

In addition to mastery of UML and database design, it's sometimes necessary for the SA to have experience with specific tools and processes when the organization has decided upon a specific process for software development. The most common software and process is the Rational Rose Unified Process. Other tools and processes exist the ones that the SA will have to master are based on what the organization has chosen.

Perhaps the most critical skills for the SA are the ability to create consensus and understanding around the architecture. While a development lead my need to involve a few people in their detailed design the architecture of the application touches every member of the team and there's a need to get them to understand it and agree with it. Once the SA has created the architecture it's time to communicate and sell it.

Where's the position heading anyway?

There is a great deal of pressure in the US to move development to countries with cheaper labor. While the SA role could be outsourced, , there is some insulation because of the need to work closely with the Functional Analysts in the gathering and organization of requirements. The distributed software of the global world requires more effort on the part of the SA and increases their need.

The overall need for SAs will continue to increase as the problems that the SMEs present are more complex and thus they require more complex solutions. The more complex solutions become, the more SAs will be required to create it.

The software development tools were supposed to reduce the effort of the SAs and therefore reduce their need for the role, however, that increase in efficiency has been far outstripped by new demand.

The Good, the Bad, and the Ugly

- **Good:** Key, High-Value Position - The SA is a key role and one which can provide immense value if done correctly. This generally means a healthy salary.

- **Good:** An SA is likely to get to interact with many of the key members of the development team as well as key members of the user community. This makes it a very visible position.

- **Bad:** Hard to keep up - Being a SA means keeping up to date on a wide variety of new techniques, patterns, and tools. The effort to keep up can be very draining at times.

- **Bad:** Difficult to get right - The role requires balancing so many factors that it's difficult to get right. In other words, it's easy to fail.

- **Ugly:** Requirements - Although a good Functional Analyst can provide great requirements a moderately skilled one may not. The difficulty is that most people, including seasoned SAs, have trouble spotting bad requirements documents before it's too late. The SA must always have to consider that the requirements may require the SA to do a lot more research and legwork into what the client really needs.

- **Ugly:** If a project fails, the SA is at the top of the list for people to blame.

Conclusion

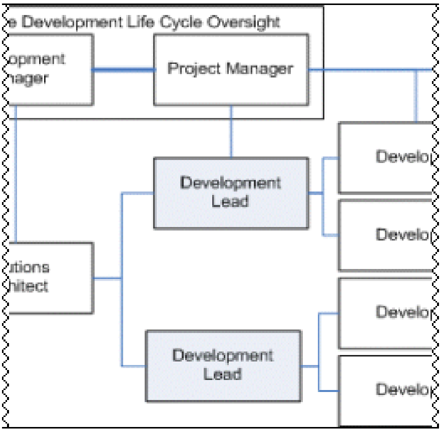
The solution architect role may be the most sought after role in the software development process but it's not without its challenges. Learning the broad array of skills, shouldering the responsibility, and dealing with the consequences can be more than the average mortal may want to take on.

Anatomy of a Software Development Role: Development Lead

Originally published on Developer.com May 26, 2005

There are a number of roles involved in developing software solutions. The development lead (DL) role is the one that glues together the overall architecture created by the solutions architect (SA) and the developers. Figure 1 shows the positioning of the Developer Lead within a project organizational hierarchy. The development lead is the part of the puzzle that bridges the vision of the architecture with the reality of the code. (If you've not been following the series, you should read Cracking the Code: Breaking Down the Software Development Roles.)

What's a Development Lead?



The development lead is the mid-point in the path between being a developer and being the solutions architect. They are still rooted in the reality of the code and the capabilities of the developers they have working on the project. They were once developers themselves and instead of spending all day every day coding their own tasks they now lead and mentor developers as the developers have problems that they don't understand.

The development lead is responsible for fleshing out any of the details in the architecture that the Software Architect didn't spell out in detail and for the creation of program specifications from which the developers work. This process refines the SA's vision and makes it more practical. This process is much like the selection of materials for the building of a house. The SA specifies a sturdy exterior to withstand the elements. The DL specifies brick and which type should be used in the facing of the house, someone else actually installs the bricks. For a development project the DL chooses the methodologies and techniques that will be used by the developers to solve specific problems.

Whether the development lead has formal responsibility for the developers on an organizational chart or not, they will be the ones who will direct the day-to-day activities of the developer. In many organizations the development lead isn't burdened with the formal administrative management of employees, they are instead freed to focus their time on helping developers be successful.

Getting Started as a Development Lead

The best way to become a DL is to become a good programmer, one who is constantly evaluating the right ways to create solutions and one who's able to see the patterns at work in the architecture. The title may change from Developer to Development lead or programmer analyst, however, that's not generally the first thing that happens. Generally it's the process of how coding is approached that happens first.

The DL role is one that starts with the mastery of key skills of a developer and adds to that the ability to convert concepts into deliverable solutions. The developer must look beyond the tasks that they're doing and see the ability to create more reusability in the work they're doing. Developers who are trying to move forward into more efficient and thoughtful ways of doing things will be given more responsibility and will move forward into the development lead role. The transition from developer to development lead may happen in as few as three years but more frequently takes six or more years. This is due in part to the broader range of experiences that the DL requires.

What's in their Toolbox?

The toolbox of the development lead is much like the junk drawer of the average household. It contains a wide variety of implements that can be used in a variety of ways. Most of those implements, however, spend months languishing in darkness only to be revealed when the need arises.

The development leader is constantly creating program specifications. Often the most used tool, the one that may never get put back into the junk drawer, is the word processing program. Because of this there's almost always a copy of a word processor running on their workstation. The specific skill that the development lead possesses - which others in the process may not - is in the ability to rapidly create many small documents. Where others in the software development process are focused on creating larger more far reaching documents the development lead creates a multitude of program specifications for developers that are quite focused in their nature.

One of the other unique characteristics of the DL role is their need to leverage other kinds of media in their documents. Program specifications require flowcharts, class diagrams, Entity Relationship Diagrams (ERDs), and a variety of other visual techniques to clearly convey the DL's meaning to the developer. That means they become adept at adding in Visio diagrams, drawing objects, and any other kind of visual device to help the developer see what is expected.

The development lead spends a great deal of time managing the development process. The most commonly used tool to do this is a source code control system. Solutions like Microsoft Visual Source Safe, CVS, Subverson, and other more robust tools allow the tracking of what code has been checked in by whom and what the differences are from previous versions. The reason the development lead spends so much time in the source control system is that it provides a tangible way to assess progress of every developer that the DL is working with. It also provides a way for continuous code reviews to make sure that the developer is both delivering what the specification calls for and is following appropriate coding standards.

In addition to spending time with source code control systems, the developer lead is also a primary person involved in source code walkthroughs. In fact, they are often the lead person in walkthroughs. These code walkthroughs are designed to help the DL understand the code that is in the system so they can ensure its quality but also serve as an opportunity for the DL to help coach, train, and mentor budding developers in ways of thinking about the code more deeply and more thoroughly. Although relegated to a trivial part of the process in many organization the code walkthrough or code review is an important part in both the quality of the system being built and the long-term development of the developers who are working on the project.

As we start to dig deeper into the back of the proverbial junk drawer we run into configuration management. This is the tool or set of tools that good development leads periodically pull out to validate that their development environment matches their testing environments, which in turn match their production environments. These tools are also used to evaluate whether the environment has changed. They start out with a baseline run and then after that each subsequent run is evaluated for changes against the baseline. This tool is most often pulled out during troubleshooting exercises when it's unclear what's changed from a time when the code was working to now, when the code no longer works. Skilled DLs also pull these tools out on a periodic basis to make sure they understand the changes and work to establish a new baseline.

The final tool in the toolbox isn't a tool at all. It's more of a grab bag of techniques and tools that are used to support the troubleshooting and diagnostic purposes. DLs are often called to support troubleshooting efforts in production systems as well as the continuous barrage of support requests from the developers. A good DL always has a set of tools that they can use to isolate the problem. These range from a working knowledge of the tools built into the operating system, to a set of testing tools that have been built for testing various pieces of the infrastructure, to tools purchased for troubleshooting. These tools include profilers, memory checkers, debuggers, and more. The specific set of tools that each DL has is slightly different, however, each will come prepared to start troubleshooting a wide variety of problems.

One of the characteristics of a development lead is their ability to create their own tools when no tools exist to solve their problem. Their toolbox may be much like a junk drawer in that you never know exactly what will be in it. You can, however, almost always guarantee at least a few of the tools will be their own unique creations.

Where's the position heading anyway?

The path forward for a DL is directly into the SA role. The skills of the SA are very similar to those of a DL and vary only by experience, breadth, and responsibility. Movement to the SA role is simple but difficult. Because there are relatively few SAs in the world, even fewer than there are DLs, the path up is largely about being in the right place at the right time to make the advancement.

Of course, the willingness to take a larger portion of the development lead work, the willingness to help out and support the SA when they get overwhelmed, and a willingness to take on challenging projects will certainly help.

The bad news about the DL role is being squeezed. SAs are sometimes being forced to take positions as DLs leaving fewer positions available for truly qualified DLs to fill. The movement towards global software development is changing the way that the DLs must think and operate in ways that make it more difficult to do a quality job and to distinguish themselves from regular developers.

The market continues to consume more and more developers bringing hope to the need for the position. The resurgence of interest in the software development lifecycle is once again helping organizations of every size to realize the value that a few key roles can play.

The Good, the Bad, and the Ugly

With any position there are good things, some bad, and a few ugly. The development lead position is no different.

- **Good:** Position with advancement - The solutions architect role is just a hop, skip, and a jump up the proverbial corporate ladder. The solution architect role only requires a bit more experience and a few more skills. It's easily within reach of a development lead.

- **Good:** If a developer likes to code, then the Developer Lead role allows them to be promoted and yet still be able to spend some of their time coding if they want.

- **Bad:** Monkey in the middle - The position has the potential to get struck between the grand vision of the solution owner and the reality that is available from the developers on the project. Occasionally, that gap is to far for anyone to bridge.

- **Bad:** Keeping up with new technology, while enjoyable, can be time consuming.

- **Ugly:** When there is more "hands-on" coding to be done than there are regular developers, then the Developer Lead is usually the first person pulled onto coding.

- **Ugly:** The DL can be presented with a great vision but do not have the technology (due to cost or politics) or products to implement it effectively.

Conclusion

The DL role is a pivotal one that is both a natural progression from the developer role and a step away from the solution architect role. With their feet grounded in the world of code and their brains thinking about how to make things better they are the critical link between the vision of the SA and the reality of the project. Despite the challenges the DL role is a very rewarding one particularly as you see projects come together and developers get better at their craft.

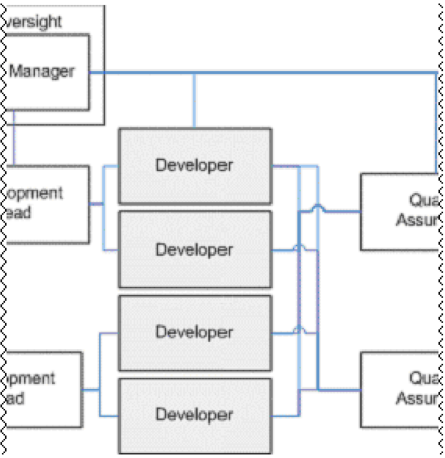
Anatomy of a Software Development Role: Developer

Originally published on Developer.com June 9, 2005

In a previous article, The Many faces of the Developer , I walked through some of the different types of developers that there are - different specializations that developers have worked on. In this article we'll take a look at what the job is in general, how to get it, and what to do once you have it.

What is the developer?

Ask any developer and they will tell you that the heart and soul of the development process is the developer role. Developers are often the huddled masses yearning to create software. They are the workhorses of the whole system creating the code that the other roles only influence. (The exceptions are the development lead and solutions architects who write small amounts of code from time-to-time.)



From a technical perspective the developer is at the most basic level expected to be able to translate algorithms and technical specifications into code that can be executed on a computer system. The language syntax and structures of code must be understood. The language syntax for writing programs is typically the easiest skill for a developer to learn. There are many books, training programs, and tools to teach you the syntax of the language.

But knowing the syntax of a programming language is only the basic requirement to be a programmer. What offsets a developer from this are a number of additional skills. Additionally, the differentiating skills that separate the good developers from the typical developer are many. However, some of the important ones are below:

- Developing Understanding - Nearly anyone can blindly follow the instructions laid out for them, however, good developers make it a point to understand what they're doing so that they can identify potential issues and opportunities for improvement at every turn.

- Structures and Algorithms Mastery - In software development there isn't any one "right" way to do things since the same problem can be solved dozens of ways. However, there are ways that are "more right". Mastering structures and algorithms means that the problem is solved in the most straightforward manner no matter what the language. Algorithms are step-by-step patterns that explain the process of performing an action. Structures relate to this by being the containers for the information that is being transformed by the action. When woven together they form the fabric of an application.

- Specialization - As mentioned in the article The Many faces of the Developer specialization demonstrates initiative to learn more and to grow which can help to separate a developer from the pack.

Getting Started as a Developer

Getting started as a developer may seem like a daunting task. Developers must have that prerequisite syntactical and algorithm skill to begin their professional journey. The question I most frequently get asked by aspiring developers trying to get started is: "What language is the right language to start with?" The answer I most frequently give is "It doesn't matter." Of course, it does matter. I wouldn't want to send someone off to learn COBOL or

FORTRAN or ADA or any of the other languages that haven't really got any real potential for new developers, however, most often the question is whether they should learn C++, C#, VB.NET, or Java. The reality is that any of these languages and associated development platforms has a sufficient following to allow someone to find a job. If you're trying to get started, what language you pick won't be the primary issue in most cases. Although some specialties will gravitate to specific languages the cases where this will be at issue are few Learning the language itself and developing some experience with it will be.

Getting that precious first experience can be key in becoming an employed developer. Despite the initial reaction that there isn't any way to get experience without experience it is easy to find places willing to let you get experience. You'll find that clubs, churches, and other not-for-profit organizations are always looking for someone to donate their expertise. It's a great way to get some experience without having a paying job.

What's in their toolbox?

Developers have a relatively limited toolbox. They are expected to be able to use the development environment including compiler and debugger for the language or languages they've chosen as well as a handful of the common tools that every member of a development team is expected to know how to use. These tools are typically integrated into one platform that functions as both a compiler and debugger. This is typically the same tool that was used to learn the language so learning a development environment isn't typically a big challenge.

In some organizations a basic familiarity with the automated testing tools is also required. This would include unit level testing tools designed to test basic code assumptions as well as tools designed to perform system level tests. Generally testing tools are primarily operated by others, however, developers may need to activate scripts written by others to verify that their changes haven't adversely impacted the build.

Developers who have specialized may find that they need to learn special tools as well. For instance, an install developer will have to learn about the installer software being used. The database developer will need to have use of database management tools.

Where's the position heading anyway?

Another very popular question I get asked is "What about offshore development?" For those in the US, that means the movement of developer positions to countries where companies charge less for coding. In those countries programmers require lower salaries to sustain a good living. While this is a concern for programmers, the reality is that the demand for developers is approximately flat according to the ITAA 2004 survey of the IT workforce (link: ITAA workforce survey). Although the outlook for new developers joining the ranks may seem dim it's important to realize that programmers represent the largest portion of the IT workspace and that positions and opportunities will develop just through attrition and upward movement.

Although there are a number of successful off-shoring case studies there may be nearly as many case studies about how off shoring may not work. The concept has been around for more than a decade and programmers remain one of the largest groups of IT workers in the US. Despite the "gloom and doom" picture that some people like to spout the reality is much closer to a optimistically cautious view.

Once we accept that there are going to be enough development positions for everyone it's important to investigate the trends that will be impacting the developer role going forward and how that can mean.

The Good, the Bad, and the Ugly

No role is perfect and the developer role is no exception. Here are some of the things about the developer role that you'll want to consider before deciding it's what you want to be doing with your career

- **Good:** Upward Mobility - Since the development area is one of the largest in IT and because there are so many different specializations and roles within the software development process there's nearly always the opportunity to move up within the hierarchy. The developer role is the foundation upon which most other positions are built. For instance, you're more likely to grow into a development leader role from development than from any other position.

- **Good:** Stability - Developers are indispensable to the organizations they work for. They are in point of fact the only people who truly understand how the systems do what they do and why they do what they do. As a result programmers have a relatively stable position even in times of cutback. This is most true of developers who are doing maintenance on critical systems but can apply to all developers. This is not to say that developers are recession proof or that they can't be laid off, that happens in every field. However, when compared to other positions in the software development lifecycle things are relatively stable.

- **Good:** Problem Solving - If you like problem solving the developer role may be for you. Developers are in a constant cycle of building and debugging their code. Both sides of that cycle can heavily lean on a problem solving skill. While coding the developer exercises problem solving by figuring out how to get a piece of information that's difficult to get. During the debugging part of this cycle the developer is focusing on identifying the source of the bug or bugs and determining how to eliminate them.

- **Bad:** Cube Land - Developers are often relegated to cube land with little interpersonal contact. Because of the need for focused time during development distractions are likely to be minimized. For those with an extroverted personality this might represent a challenge.

- **Ugly**: Dysfunction abounds - One of the ugliest things about the developer role is that it's at the end of a very long whip. When the software development process works well the developer feels the crunch of a deadline. When the software development process doesn't work well, and it often doesn't, the developer can be crushed by conflicting needs to get the product completed and a series of quality or incomplete feature issues. The truth is that many managers in most organizations do a really bad job of managing the software development process, which can create very painful positions for developers.

Conclusion

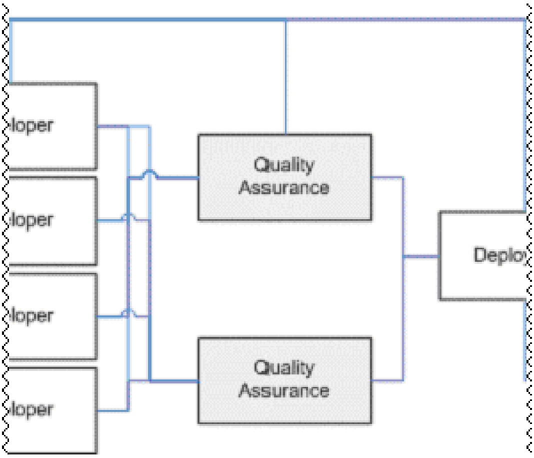
The developer role is the core role into the software development process and is the one that there are the most open positions for. The role itself has its own set of trade offs, however, for the upwardly mobile professional it may be the way to put yourself on the path for the position you really want.

Anatomy of a Software Development Role: Quality Assurance

Originally published on Developer.com June 24, 2005

The Quality Assurance (QA) role is the role responsible for guaranteeing a level of quality for the end client, and to help the software development team to identify problems early in the process. It is not surprising that people in this role are often known as "testers". Of course, the role is more than just testing. It's about contributing to the quality of the final product. (If you've not been following the series, you should read Cracking the Code: Breaking Down the Software Development Roles.)

What's the Quality Assurance role?



The quality assurance (QA) role is one that is focused on creating a quality deliverable. In other words, it is the responsibility of the QA role to make sure that the software development process doesn't sacrifice quality in the name of completed objectives.

The QA role works with the Functional Analyst (FA) and the Solutions Architect (SA) to convert the requirements and design documents into a set of testing cases and scripts, which can be used to verify that the system meets the client needs. This collection of test cases and scripts are collectively referred to as a test plan. The test plan document itself is often simple providing an overview of each of the test cases. The testing cases and scripts are also used to validate that there are no unexplained errors in the system.

The test plan is approved by the Subject Matter Experts (SMEs) and represents the criteria to reach a project closing. If the test cases and scripts in the test plan are the agreed upon acceptance criteria for a project then all that is necessary is for project closure is to demonstrate that all of the testing cases and scripts have been executed successfully with passing results.

A test case is a general-purpose statement that maps to one or more requirements and design points. It is the overall item being tested. It may be a specific usability feature, or a technical feature that was supposed to be implemented as a part of the project.

Test scripts fit into the test cases by validating that case. Test scripts are step-by-step instructions on what to do, what to look for, and what should happen. While the test cases can be created with nearly no input from the architecture or design, the test scripts are specific to how the problem was solved by the software development team and therefore they require an understanding of not only the requirements, but also the architecture, design, and detailed design.

The quality assurance role is split into three parts:

- **First:** The role creates test cases and scripts.
- **Second:** The role executes or supervises the execution of those test cases and scripts.
- **Third:** The role facilitates or performs random testing of all components to ensure that there's not a random bug haunting the system.

In some organizations, the quality assurance role has two specializations. The first is the classic functional testing and quality assurance as described above. The second, is a performance quality assurance role where the performance of the completed solution is measured and quantified. The performance QA role is an important part of the large system development quality assurance process.

The quality assurance role also has within it a wide range of potential titles and specific responsibilities. From the entry-level quality assurance professional who executes and document tests to the QA lead who works with the FA and SA to create the testing plan, cases, and scripts. The role also extends through QA manager position that may take responsibility for the quality of a solution. At this level the QA manager and solutions architect work as peers to ensure the final solution has the highest quality.

Getting Started as a Quality Assurance Professional

Split evenly with the developer role the standard QA role is an entry role into the software development process. The QA can be entered with a basic understanding of the process, and minimal - if any - prior experience.

The entry spot for the quality assurance role is simply running the testing scripts created by another quality assurance professional. This requires no special skills other than the willingness to step through a process one step at a time and to document the results. Once experienced with running scripts other parts of the software testing process will open up and eventually the opportunity to take on the quality assurance responsibility for a part of the system - or future systems may open up.

The role requires an attention to detail so anything that you can use to demonstrate this attention to detail is helpful. It is also helpful to demonstrate any time where you had to keep meticulous records, whether it was for a science experiment or any other kind of task. Since getting started with QA means taking good notes, penmanship is a plus. If this is a weakness, it's possible in some cases to take notes electronically; good penmanship is replaced with the need to be effective at taking notes on the computer.

What's in the Toolbox?

The QA toolbox is filled with things that make validation possible and easier. It includes automated testing tools and the skills necessary to validate applications, database values, and workflows when there is no easy way to validate the correct answer.

The first step is patience. Every test script in every test case must be run by hand. This process is often exhausting but necessary to validate both the software being developed but also the test scripts themselves. Screen capture utilities and screen annotation tools can be very helpful in the process of communicating the errors.

The next step is automated testing tools. These tools allow you to run a test script without the need to be manually banging out keys on a keyboard or clicks of a mouse. These tools come with recording software to allow you to record a set of steps and convert it into a scripting language. The scripting language allows the steps to be customized. By customizing the script from the recorded script it is possible to parameterize the set of steps so that different data can be used. At least one automated testing suite should be in your arsenal if you're a serious QA professional.

Most automated testing tools also extend themselves into being used to test scalability and performance. These tools help to identify how a system will respond under heavy loads such as large numbers of simultaneous users, large amounts of data, and more.

As mentioned briefly above, the larger and more public the system, the more important performance testing becomes. Because of this, the skills of performance testing and interpreting the results of that testing become more valuable as the system's importance increases. It is common to have QA analysts in a functional testing role; this is the kind of role that most people think of. QA analysts are also found in a performance-testing role, which is focused around the performance-testing aspects of a software solution.

Where's the position heading anyway?

Quality in general has taken a beating over the last several years as high profile software failures have made it into the media and because there is a growing awareness that viruses are possible only because of a quality slip in the software that the virus attacks. Despite a growing awareness of software development failures there's little growth in the QA role. This may be due to a variety of factors including: misunderstanding of what the QA role, the lack of understanding of the value, or simple ignorance of the need for quality assurance.

There's a general misunderstanding by a great deal of business people and technical managers about what testing is - the primary focus of the QA role. It is believed that testing is something that the developer does. To some extent this is true, most developers do perform testing (although I've met a few for whom this skill was completely lacking.) However, whether you believe in the psychological concept of objective introspection - the process whereby you investigate your own thoughts, actions, and feelings - or not, it's clear that having the same person who developed some a program test it leads to many more errors to be found later on. This is because the developer makes the same mental mistakes in testing as they did during the development. The result is that the defect slips through until much later in the process when it's more expensive and harder to solve. As a result, even though the developer may run the initial unit tests, it is unwise to rely entirely on the developer for testing - despite this many organizations do. The QA role is designed to validate the developer's tests but also to ensure that the work of several developers fit in together.

In some software the level of appropriate testing is so low as to make it seem completely unnecessary to have a QA role on the project. For instance, the development of a testing harness - code designed specifically to test other code, is rarely done with any testing what so ever. These is little need to test the harness itself as it will be well tested as it's used as a tool to test other things. The thinking of many business and technical managers is that testing has value, but the value it has is low enough that less testing than might be appropriate is done. This misperceived position on quality sometimes prevents managers from allocating enough resources to the QA process.

In some cases when planning for the project the QA need is ignored completely. This is due to complete ignorance that this part of the process exists and that it's a vital part of the software development process necessary to ensure a quality result.

All of these factors play into the future of the role. Luckily more work is being done on educating technical professionals as to the extent of the need for the quality assurance role and how that role fits into the overall needs of software development. Despite its rocky past QA is beginning to be invited as a full member, to the software development dinner table. It's likely that the demand for quality assurance professionals will continue to rise, slowly but steadily.

Standing Out in the Crowd

Standing out of the crowd can be difficult to do when the expectations of the position are low and the role requires a constant tightrope walk between identifying too many non-conforming software components (a euphemism for a bug ridden software component) and not identifying enough problems with the software being tested. However, as difficult as this may seem it's certainly possible with the right passion.

One of the best ways to stand out in the QA role is to have other tools in your toolbox that you're able to use - even if you're not adept at them. The ability to execute SQL statements to identify whether the data from the user interface is making it to the database or not is, for instance, a powerful way to test user interface code. In general, learning other ways to test is a skill that will prove its value as you become more and more respected in the QA role.

Another potentially important skill that can differentiate you is the ability to read and understand the underlying code that is the application. Whether the application was built with Java or Visual Basic the ability to read the code that makes up the application is valuable in helping to identify the specific spots where problems are occurring. This reduces the amount of time the developer must spend trying to sort through the code and find the problem. Obviously the expectation is not that you write code or that you can read the code as well as the developers, however, the ability to convert the knowledge that you have of what you're seeing directly into the spot in the code where the problem lies will make you stand out from your peers.

The final skill which can help you stand out of the crowd is problem solving. (Ref: Problem Solving links here.) Being able to run the testing script and identify problems is what is required of the role, however, the role has the opportunity to dig into the problem and find out more about the problem and create a better understanding of the conditions where it does appear - and doesn't appear. Creative problem solving and the ability to curiously seek out more information about the problem can separate you from the pack.

The Good, the Bad, and the Ugly

Never was it truer that each role is met with good things, bad things, and truly ugly things as it is in the QA role. The role in the right environment is a shining example of how one person, or a few people, can make an impact. However, in the wrong situation it can leave someone bitter, frustrated, and burnt out. Here are just a few of the good, bad, and ugly things about the QA role.

- **Good:** Quality - The QA professional can take pride in their ability to instill quality in the result of the software development effort. More than any other individual the QA professional can make sure that a quality product is being produced.

- **Good:** Involvement - Because all aspects of a system have to be assured for quality, the QA role often allows a person to be involved in many phases of a project, including the early ones.

- **Bad:** Don't shoot the messenger - One of the challenges is that the QA role is designed to be the bearer of bad news. Because of this it's easy for others on the software development team to develop a negative attitude towards the QA professional. That means a lot of work on the QA professional part to keep everyone in a positive frame of mind.

- **Ugly:** Balancing the amount of testing done - The ugly part of the QA role is realizing that there's only a certain amount of quality that is appropriate for any given program. Some programs, such as those dealing with life or death situations, require substantial testing. However, many of the systems that we operate with today are not life-and-death situations and require only an appropriate level of quality assurance. The ugly part is that no two people will draw the line; on how much quality assurance must be done, in the same place.

- **Ugly:** Testing generally occurs after other things are done. This means that testing is often close to the deadline of when something is due to be completed. If others are late in completing their tasks, the QA person may have their time squeezed to near nothing.

Conclusion

The quality assurance role is one of the most undervalued and potentially impact roles in the software development process. Despite the public failures of software there is little progress in making the quality assurance role takes it's rightful position of importance. Cost constraints are forcing organizations of all sizes to release earlier and earlier betas that encourage users to find the quality problems of the software on their own. In order for the software development industry to gain the professional respect it needs the position of the quality assurance role must be elevated.

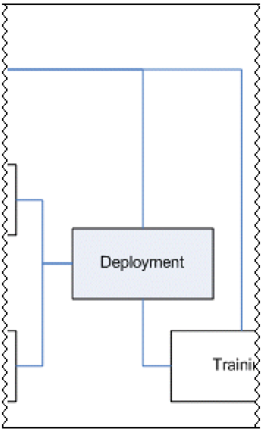
Anatomy of a Software Development Role: Deployment

Originally published on Developer.com July 11, 2005

The deployment role is a role that is often overlooked much to the pain of the users. The actions of this role are the final step before being able to hand over the code to the users for their first real road test of the solution. It is the deployment person who can have the largest impact on the initial perception of the software for the people who are first trying it out. (If you've not been following the series, you should read Cracking the Code: Breaking Down the Software Development Roles.)

Success here can hide quirks in the solution but failures here can give the wrong impression about the software.

What's the Deployment role?



A software solution of any complexity will have dependencies that must be present before the solution can be used. Many of these dependencies go unstated. For instance, a Java program needs a certain level of the Java runtime environment installed to be able to run. .NET based applications require a specific version of the .NET framework and common language runtime to run. In the case of database applications specific versions of the software drivers to connect to the software to the database are required.

In addition to these software dependencies, there may also be hardware dependencies. This could include a minimum amount of member, a required amount of hard disk space, access to multiple machines (such as a database server versus an application server), access to the Internet, and more.

The more complex the solution, the more prolific the requirements are. Solving that problem is the role of the deployment professional. It is the focus of the deployment role to create a program that constructs the operating environment that the solution needs. This includes installing prerequisites, making necessary configuration changes, identifying conflicting software, and manipulating any other components of the environment that the solution may need.

The first step in the deployment role is to identify and test by hand all of the steps necessary to make the software work in an environment. These instructions are often written as complete documentation so that even without the installation program they can install the software. These installation instructions sometimes serve as the early procedure for installing the software through alpha pre-release program and even throughout the life of the software if it is installed to a small enough group of clients.

From the installation instructions, an installation program is constructed to automate the steps identified during the creation of the installation instructions. The program must do more than just install the system, it must also, in today's world, be able to uninstall the software once it has been installed. This must also be done in a way that is respectful of the other software that is already installed in the customer's environment. Figuring out how to be a good software citizen with an installer is one of the most challenging things that any software development team can do.

It is challenging for a variety of reasons but none more important than the need to understand both the software development process as well as an understanding of the operating system and the network infrastructure that may be used to deploy the software. The requirement for understanding the software development process may seem obvious. They're a part of the process so they should understand it; however, the required understanding goes deeper than this. They must also be able to identify inherent dependencies created by the development process that may not readily be obvious.

A command of the operating system (or systems if the software is to be used on multiple platforms) is also necessary since it will be the deployment professional role to manipulate the operating system from their installation program. The subtleties of whether to register a .NET assembly in the global application cache or not is an important thing for a deployment specialist to know. Considering all of the code access permissions and learning how to setup those permissions so that the user will be able to run the application is also important.

The final component that the deployment professional must understand is how automated deployment tools, such as Group Policies in Active Directory and products like Microsoft's Systems Management Server, deliver software to the clients and how the installations must be constructed when being deployed via these mechanisms. This is a critical consideration when the software will be deployed across the enterprise since it's not feasible to go install the software manually on each machine.

The final consideration for what the deployment professional will do is creating patches. Software teams necessarily create updates to their solutions. These updates need to be deployed to systems just as the original program was. Most consumers expect that the patches that they receive today will integrate into the existing installation rather than being another program that must be added or removed. Deployment professionals are entrusted with developing strategies that deliver patches efficiently.

Getting Started as a Software Deployment Professional

With the broad range of skills and the extensive experience requirements you might think that getting started as a software deployment professional might be hard to do. In truth there are two paths that can be used reach the software deployment professional role.

The first way to start is to work on the help desk, servers, and networking side of an IT department. In this role dealing with installation problems, trying to find workarounds for issues, the intimacies of the operating system and the network are revealed. From there it's simply a matter of volunteering to work with the software development team to help software get installed in the environment as well as possible. Since many software development teams, particularly those inside of an organization, don't have a dedicated person filling this role they often welcome the help.

The second path to becoming a software deployment professional is to work as a developer and focus on identifying issues that may become challenging during deployment. That means paying attention to the steps to setup a developer box and documenting them. It means looking for things in the code that may create deployment issues. From here it's natural to get involved in troubleshooting issues when deployments are done. Once you're troubleshooting the problems of deployments the role of the software deployment professional is in easy reach.

The deployment professional role is often combined with one (or more) other roles. Because deployment isn't always what is called for, deployment professionals may fill the quality assurance role, a developer role, or other roles along the way. The deployment role may be one that you'll find yourself volunteering for from your primary role.

What's in their Toolbox?

The toolbox of the software deployment professional is a grab bag of small tools that can help work around limitations and quirks with deployments and a few larger tools designed to construct the installation programs themselves. Following are a few of the tools useful to the software deployment professional.

- **Batch files** - Batch files, which are the legacy inherited from DOS, is the most basic way to string together a set of commands. Batch files are showing their age with minimal error handling, messaging, and branching capabilities. However, they work on nearly every system so they are often a quick way to work through problems and are a quick way to glue together different operations.

- **Scripting Languages** - The inclusion of WSCRIPT and CSCRIPT utilities in the operating system have made it possible to use VBScript as a way to create more robust strings of commands with better error handling and a better ability to message. Many other, proprietary scripting languages exist as well. These languages typically require the installation of a tool on the workstations to be able to run the script. Scripts are used because they are able to do things that are not possible from batch files and are still easy enough to put together that they can be used quickly.

- **Packaging Tools** - The core tool for the software deployment professional is a packaging tool. In Windows these tools create a Microsoft Installation (MSI) files and can create patch files (MSP). These tools create a specifically formatted database that the installer service in Windows can read and perform actions. Extensions in the installer format allows for user provided code to be run. Wise and InstallShield have been the historical leaders in this market. Microsoft has recently released a Windows Installer Xml (WIX) project to open source that is very promising.

- **Package Management Tools** - Occasionally it's necessary to snoop on the package that was created. It may need to be taken apart, monitored while running, or unwound into a set of related files. In the MS Platform SDK you'll find a set of tools for opening, reviewing, and manipulating MSI files. These tools are invaluable for troubleshooting problems with MSI files.

- **Virtualization software** - With software such as VMWare and Microsoft's Virtual PC, the deployment role has gotten easier.

Rather than having to have numerous machines with different configurations to leverage as a test, the deployment engineer can have several virtual environments. This saves time because they can simply fire up the virtual machine and go. It's even more useful in keeping the amount of space required for the deployment engineer down since they won't have to have a dozen or more computers that are used infrequently for testing.

- **Disk Image Software** - Virtualization software have features that allow users to undo changes that can be very useful for testing. It allows you to try different variations without having to reinstall between attempts. If it is necessary to work on physical hardware instead of virtualized hardware then a software which copies the drive image into a file which can be restored to the machine after a test, such as Symantec Ghost, is a must. It substantially cuts down on the time spent rebuilding test systems.

Where's the position heading anyway?

More and more consumers are getting computers. The average level of knowledge on computers is decreasing while the total number of customers is still increasing. Because of this the demand for software deployment professionals is growing. Instead of software being installed with a batch file the program must today be installed through a large number of slightly changing steps. The larger number of installations requires more sophisticated handling. The lower tolerance for problems with software due to lower knowledge requires that installations be more bullet proof. The good news is that this means a greater need for software deployment professionals. It's more important than ever to be able to deploy software quickly and easily.

The difficulty level is going down even as the complexity goes up. The installation tools, often the last thing thought of by the industry, have been making good progress to get better. The release of Microsoft's Windows Installer XML (WiX) toolkit is encouraging since it is making it easier, and less expensive, to develop installation files that the Windows Installer service can install.

The Good, the Bad, and the Ugly

The deployment professional can often times experience the good, the bad, and the ugly of their role all in one day. They can see the software get deployed across the organization - only to find out that here was one application it wasn't tested with, and that one application is one of the mission critical applications for the organization - which the installation just broke. Here are just a few of the good, bad, and ugly things about the deployment role:

- **Good:** High Impact - The role is one of high visibility and high impact to the user community. When done right it's easy to save hundreds of hours time for the users.

- **Good:** Often get to use a number of different systems with a number of different configurations. This can be fun!

- **Bad:** Never Enough time - Being near the tail end of the process the installation often gets rushed. The testing time to ensure that the installation doesn't break other applications is often hard to come by.

- **Ugly:** Difficult to get right - Building good installations is still very technically challenging today. Over time good deployment processionals learn how to avoid big mistakes, however, even veterans are known to make mistakes that has huge impact.

- **Ugly:** No matter how many systems you test your installation against, it always seems that a user will have a system that is slightly different. When something goes wrong with the installation, this person gets blamed.

Conclusion

The deployment role is one that is critically important to the initial impression of the software being delivered. A bad installation experience can sour just about anyone's taste. Because of this the role of the deployment professional is increasing. Although professionals are still often shared with other roles the deployment role is one which continues to grow.

Anatomy of a Software Development Role: Training

Originally published on Developer.com July 27, 2005

Wrapping up the software development lifecycle and turning over the completed product to the users is the training role. The training professional is the last one in the process since they are the ones who get the mass of users to use the software that has been created. Their purpose is to help the users understand how to use the software that's been created. (If you've not been following the series, you should read Cracking the Code: Breaking Down the Software Development Roles.)

What's the Training role?

The training professional first and foremost creates the materials necessary to train users how to use software. For that reason, training professionals are often tapped to create user documentation and help files in smaller organizations.

The training professional is ideally someone who has an instructional design background and therefore understands how to create materials that are effective in helping adults learn. They are also, ideally, someone who can approach the problems that the software solves in a way which makes sense to the users. Click here to see how the the Deployment role fits within the full organizational chart.

Training materials range from slide decks to handouts for training sessions to self-study guides. No matter what the material type the focus is on creating a situation where the user will find it easy to use the software - even if the user interface isn't perfect. The trainer needs to find a way to communicate what the product does and how it does it. Whether the medium is a standard printed manual or an interactive video the goal is still the same - efficiently convey how the user can best use the solution. The trainer's role is often the role of the super user. Because they must instruct on the software's use, they need to have a detailed understanding of the software across nearly every feature.

In addition to the creation of training materials the training role also often delivers training. The training can be instructor lead training like the traditional classroom, an Web cast, or one-on-one sessions with key users. Performing the actual training sessions often provides invaluable feedback to the continuing development training materials themselves. It helps to expose the kinds of questions that are in the minds of the users - the kinds of questions the trainer must be prepared to answer. The author teaches most materials at least once.

In some cases a special form of a trainer may evolve in the software development process. The special form is an evangelist. An evangelist is a passionate, charismatic individual who evangelizes the product in the market. Evangelists are almost universally someone capable of filling the training role but whose passion and charisma make them stand out as advocates for the product. While they can and often do perform roles similar to the rest of the professionals filling the training role, they often side step or short circuit detailed training and answers in favor of engaging the user in the vision of the solution.

Getting Started as a Training Professional

Getting started as a training professional is often easier than other software development lifecycle roles because the primary ingredient to success is communications skills. The ability to communicate complex topics with precision and understanding separates the mediocre training professional from the stellar. The beauty of this is that everyone has experience with communicating. The powerful communicators that are successful training professionals can write as well as speak and have learn how to really connect with the people that they speak with. Getting started doesn't take the awe inspiring communications skills it relies on a strong desire to help others learn about the solutions being created and an enthusiasm for the process that students can feel. The mechanical skill of how to engage the audience comes with experience and coaching. They require as a raw ingredient a passion for learning.

While the role may be about training users in a classroom style setting or more focused on the creation of materials, it's the enthusiasm for helping others learn that will make the next training professional stand out.

Although an understanding of instructional design is greatly appreciated in the training industry it is usually not a prerequisite. If the role is primarily focused on delivering content that has already been created then the ability to speak comfortably in front of a group is often the only requirement. These are roles are often entry roles in an organization. More advanced candidates will have the ability to anticipate what the group of students will need and have the ability to "think on their feet." The training role is filled with twists and turns as students ask sometimes blindly simple questions and other times ask the most arcane trivia. Learning how to control the students is yet another skill that develops for most professionals over time.

What's in their Toolbox?

The training professional's toolbox is a bit more organized than the toolboxes of other members of the software development process. In fact, they often subdivide their tools into content creation tools and tools for instructor lead training.

In the content creation category the toolbox contains:

- **Word Processing** - the same word processing program that was being used by other members of the software development team but perhaps with a deeper understanding.

- **A layout or publishing program** - this type of program may also be handy if the type of solution lends itself to quick guides, keyboard shortcut sheets, or other kinds of highly formatted handouts.
- **Presentation software program** - Presentation programs, such as Microsoft PowerPoint, create the ability to generate slides for teaching the material.
- **Virtualization technology** - Although all training professionals do not use this technology, virtualization technology can be immensely powerful. This technology allows for the program being trained can remain unaffected by changes on the core system that they use. VMWare and Microsoft's Virtual PC are good examples of virtualization technology that can be valuable in creating a repeatable environment for training.

In the instructor lead training side of the tool bag there are a completely different set of tools. They are:

- **Presentation Software** - This may include software such as Microsoft Power Point for creating slides to use for live presentations. It can also include Web casting software as well as video recording and editing software.
- **Telecommunications software** - If the presentation is being done remotely either completely or where some of the parties are not local, then telecommunications software may be an essential part of the toolkit. Whether the favorite software is Microsoft LiveMeeting, WebX, or another one of the presentation and conferencing software getting virtually connected to the audience is essential. Of course, these presentation and video conferencing programs are bandwidth hungry so plenty of Internet bandwidth will also be required. If a face-to-face interaction is required a video camera, which can be attached to the computer, may also be required.

- **Hardware devices** - Trainers are known for their toys. Little gadgets can make their job easier. Here are just a few of the devices that may be found in a trainer's toolbox.
    - **Remote Clicker** - Good instructors don't want to be separated from their audience by a computer. Remote slide advance and mouse movement devices are available to allow a presentation to be run remotely. These devices occasionally have built-in laser pointers that allow the training professional to point out or highlight important information.
    - **LCD Projector** - Whether the room has the LCD projector or if it's brought into the room, the training professional who's leading a class will need a way to display the content. Understanding how to setup LCD projectors is essential if instruction is a part of the training professional's role.
    - **Thumb Drive** - An instructor never knows what's going to happen. It could be that the instructor's notebook fails or the materials that were supposed to be loaded on to a computer in the room aren't available. Carrying a USB thumb/flash drive can get the training professional out of a jam.

Where's the position heading anyway?

There are more training materials today than there has ever been in the history of computers and yet there is still a need for more and better training. Because of the increasing complexity of software the need for good training is becoming substantially more important. Because of this the need for future training professionals is well assured. Add to this the proliferation of different mediums for training including various electronic learning formats and it's easy to see that there's plenty of work to be done.

Training is also a part of the process that will always be personal and therefore often face-to-face. Learning is a sensitive process. We like to learn from instructors that we can trust. In many people trusting in an instructor requires seeing them. In a globally oriented economy the trend is to source talent from the most inexpensive location. However, because training is so focused on conveying information it's often difficult to relocate. In other words, it's unlikely that the positions will be transferred to areas where cheaper labor is available.

The Good, the Bad, and the Ugly

The training role, like all of the roles has its ups and its downs; here are a few of the highlights.

- **Good:** Get to see the results - More than anyone else in the software development process, the training professional gets to see the real results of the work. The training professional gets to see the smile (or frown) on the faces of the end users when they are introduced to the solution.
- **Good:** Get to meet and interact with lots of people.
- **Bad:** End of the lifecycle - Training is at the end of the life cycle. It's the last thing that has to happen to get the system fully functional. Because of its location training is often compressed into less time than it should have both from a content creation perspective and from the training delivery standpoint.
- **Bad:** Thankless - Often the trainer has the thankless job of working with users who are frustrated because their process is changing. It may take a great deal of energy to keep the users happy with the solution that is being delivered to them.

- **Ugly:** A wide variety of users - Training means working with all kinds of users including both highly respectable users and crackpots who will have to be suffered.
- **Ugly:** Some trainers are required to do a lot of traveling.

Conclusion

Training is without a doubt the most extroverted role of the software development process. Even the functional analysts don't have as much contact with people as the training professionals do. Key communications skills are essential to getting the role and staying in front of the best customers.

Anatomy of a Software Development Role: Project Manager

Originally published on Developer.com August 10, 2005

The Project Management role is the first role in the software development process that isn't on the main line. The project manager isn't a person doing "real work." The project management role is one that is designed to help ensure that the software development process works as it is intended. The project management role works closely with the development management role in order to facilitate, encourage and prioritize the process.

The project management role is perhaps the most clearly defined role within the software development process due to the development of project management as a profession.

While the software industry is nascent, the project management industry is enjoying the advancement of a powerful organization in the Project Management Institute . They have assembled a guide to the body of knowledge for the project management profession that is often referred to as the PMBOK Guide. This organization has developed a widely recognized certification, Project Management Professional (PMP), which has both practical experience requirements as well as traditional testing requirements.

What's the Project Manager role?

The PMI's PMBOK Guide defines the project management role via definition of a project manager by defining what project management itself is. Their definition is:

"Project management is the application of knowledge, skills, tools and techniques to project activities to meet project requirements. Project management is accomplished through the application and

integration of the project management processes of initiating, planning, executing, monitoring and controlling, and closing. The project manager is the person responsible for accomplishing the project objectives."

You can go to the PMI website (pmi.org) to download sample content from the PMBOK Guide, buy a copy of the guide, or sign up to become a PMI member.

Within the context of the software development process the project management role is responsible for driving the work through the process and to completion. Starting with the earliest requirements discovery sessions and ending after the training has been completed, the project manager is the one role that should be consistent throughout. Project managers work with both the development team and with business stakeholders to ensure that what is being built will match what the customer expects and that this development occurs within the expected time frame. Click here to see how the Project Manager role fits within the full organizational chart.

In that the project manager is often intimately involved with the business stakeholders they are often working closely with the functional analyst during the early stages of a project. The functional analyst is focused on getting the details of what the system must do and the project manager is focused on determining what the success criteria are and helping to educate the stakeholders about how the process works.

The project management role is generally responsible for status reports, which are frankly more tools about instilling a sense of urgency and demanding definite answers than they are a process for generating reports. They expose resource and scheduling issues as well as provide a communications vehicle for issues as they begin to occur. The project management role is typically spreads across several projects and potentially development teams. The project manager is frequently working on more than one project at one time - unlike many of the other roles in the project who may be dedicated. The project manager need not be an IT specialist, although this is helpful. The nature of their role is to provide broad based oversight.

In the software development process, the project manager is one part prodding and one part smoothing over problems in the process. They're there to make sure that the process goes forward and they use a variety of processes, meetings, and documents (artifacts) to ensure continued forward progress and to identify problems as soon as possible.

The project management role must be capable of executing details with regular consistency. They must also be able to take a step back from those details to look at the big picture to evaluate things like risk and the accuracy of time estimates. All of this takes a measure of restraint and good listening skills. The restraint prevents the project manager from jumping in before it's time and the listening helps to ensure that the real problem is eventually understood - and can be disseminated to everyone.

Project managers are the keepers of the process. Whatever the process is they're responsible for making sure that it executes well. In some organizations this includes the software development methodology. The project manager is the one that makes sure that the process is adhered to so they can get the predictable results they desire.

Getting Started as a Project Manager

Being a project manager more often happens out of necessity than it does out of design. More often than not the project manager becomes a project manager because they see and understand the challenges that are being faced by the software development team and believe that they have the ability to help prevent them. This is the spark that generates an interest in taking a project management path.

One of the paths that can lead to the project management role is the functional analyst. The functional analyst bridges the gap between the business and the software development team. This is a critical skill for the project manager as well. The other skills, such as being able to generate the status reporting and communications, are necessary to maintain control of the project but they can be taught and learned with some coaching.

The project management role is one that is sometimes split into senior project managers and junior project managers. This allows the more senior staff to work on larger, more mission critical tasks, while monitoring the smaller projects through junior staff. If your organization has a project management mentoring process with junior and senior project managers getting started is best handled by approaching a senior project manager and asking what you can do to help. If no such structure exists within your organization then you'll just have to volunteer for tasks that are traditionally project management focused such as taking meeting notes and circulating them to verify consensus.

The preparation that you should do to become a project manager includes reading the Project Management Guide to the body of knowledge (PMBOK) as well as developing your own set of templates and processes which can be leveraged in the management of a process. Templates such as a project charter, which indicate an overall vision for the project and an agreement to get started, are a standard tool that should become second nature to you.

Ultimately getting into a project management role can be easy, in the case where no one is really filling that role or where there's a structured mentoring process, or it can be difficult where there are strong people leading the role of project management but who do not have a mentoring structure in place. If project management is important, you may need to prepare by reading the appropriate material and developing the appropriate templates and then make a change that allows you to try out your new learning.

Occasionally, getting started as a project manager is done completely by accident. A solution architect, development lead, or developer gets noticed for adherence to process or the desire to move the project forward and are encouraged to seek the position. This can be a problem for those who still want their hands on the code.

What's in their Toolbox?

Each project manager, it seems, manages projects slightly differently. They have their own specific way of making sure that it will work. However, there are many common themes that run through the project manager's toolbox. Here are a few:

- **Project Management Software** - Project Management software, like Microsoft Project and Microsoft Project Server are important tools for the project manager who must keep tight control of timelines and budget. Project management software helps document dependencies and status so it's clear where a project is and what impact someone missing a deadline has. Most, if not all, project management software is capable of generating Gantt charts (timelines), and PERT diagrams (dependencies). These are essential tools in their own right.
- **PMI Membership** - Most serious project managers are a member of the Project Management Institute. Their membership magazine and local PMI chapters have a great deal of value in their ability to demonstrate new ideas and provide connections to others when you need advice on a problem.
- **PMIBOK Guide** - The Guide to the Project Management Body of Knowledge provides foundational knowledge that is important for all project managers and can provide sometimes not so gentle reminders when you're project isn't on track as to why it's not.

- **Templates and Processes** - Project management can become overwhelming with the amount of artifacts (documentation) that is created. Having tools to ensure that the process itself is as efficient as possible is in every project manager's toolbox.

Where's the position heading anyway?

The project management role has gained a lot of ground over the past five to seven years. It's become well recognized within IT circles that project management is a necessary discipline that is a necessary catalyst to keep the software development process running. With the hard work of the PMI the role is becoming more standardized and can therefore be more readily purchased by business users.

In short, the future looks bright for the project management role. As more organizations realize the need for a dedicated project manager and due to the continuing need for software development project managers are dearly needed.

Because of the dual role that the project manager has in managing the software development process and managing the expectations of the business customers there are often two project managers. One who works with the software development team in whatever geographic location they are in and another project manager who is more closely positioned with the business stakeholders. Sometimes the role of the project manager on the development team side is filled by a solution architect or development lead. However, this still means that there should always be project management positions wherever you are.

The Good, the Bad, and the Ugly

The project management role, like other roles has its high point and its low points. Here are some of the key things that this role has to offer both positive and negative.

- **Good: High Visibility** - The project management role is a high visibility role. Because it has a part in so many different projects the role is often involved in reporting to executive management in an organization so it has the potential to demonstrate good work to important people.
- **Good: Key Role** - The project management role is a key role to the project -- one that can have a great impact on the overall success or failure of the software development process. If you like influence this is a role for you. Additionally, the project manager is often considered the person within IT that owns and is responsible for the project.
- **Good: Sense of accomplishment** - Because a project management role is one which works with a large number of people there's constantly progress being made. When a project gets shipped or even when the first beta comes out there can be a great sense of accomplishment in that the project was moved forward because of the role that you played.
- **Bad: Tight timelines** - The project management role is at its most crucial when the project has gone awry or when the timelines are too tight. Unfortunately with the pace of business accelerating it seems like every project has a tight timeline and doing what is necessary to ensure that timelines are met often creates a high degree of stress.
- **Bad: Overworked** - The value that project management can bring to the software development process can't be understated nor can the amount of work that it takes to create this value. The artifacts that drive consensus also require time to create. However, there's not a solid understanding of the amount of work that is required to create the value and because of this project managers often find themselves stretched too thin across too many projects.

- **Ugly: Politics and People** - Perhaps the most frustrating thing about the project management role is that it is focused on fixing political turf wars, keep people working with teach other and moving forward. This means that there's a heavy dose of pop-psychology that must go on to get the project done. The drain caused by so many whining people can be overwhelming at times.
- **Ugly**: The project manager is the one that may have to make the decision to have people work extra hours to meet deadlines.
- **Ugly:** The project manager is the one that has to maintain a projects scope - what will or won't be included. They are often caught in the middle when decisions need to be made on what can be completed within the given time frames.

Conclusion

The project management role isn't for everyone. A strong ability to adapt to an ever changing environment and the ability to work through internal politics are critical for success in this role. However, the ability to right a project that has gotten off track is much sought after and highly valuable skill.

Anatomy of a Software Development Role: Development Manager

Originally published on Developer.com August 25, 2005

At this point in the development process, it is possible to believe that there is nothing left to be done. That all of the roles outlined thus far is all that is needed to make the process work. However, the role of the development manager is critical to the long-term success of the software development team. The role that the development manager plays - particularly when interacting with the project manager - is essential to a continuously improving process.

What's the Development Manager role?

The development manager's role can be described as "everything else". Although accurate this description is not very illuminating.

The development management role is the role whose purpose it is to keep the vision on track. This is much like a CEO, who sets the vision for an organization. This of course differs from the COO, who-like a project manager-ensures the day-to-day operations. While it's the project manager's goal to get the project to the finish line, it is the Development Manager 's responsibility to look ahead to make sure that the finish line is the right finish line to be reaching. While the project management position is a management position, the development manager role is a leadership position. Click here to see how the the Deployment role fits within the full organizational chart.

The development management role works closely with the project management role to ensure that the projects are completed. They also spend time working with the business owners on planning and preparing for new projects that will soon be consuming the group. They're constantly making small changes in the current project to get a better product, develop a better skill set in the group, or just generally preparing the development team for the next hurdle that they must face.

While the project management role is dutifully executing the current process, the development manager is examining, evaluating, and assessing the impact of potential changes in the market. The development manager is constantly evaluating ways to improve the skill set of the group. That can be through the use of a new tool or technique, or additional training on fundamental skills that the group already knows but doesn't execute consistently. The project manager's short-term focus allows the development manager the ability to focus on longer-term objectives.

In addition to the long-term objectives the development management role is intimately involved with prioritizing multiple, nearly always conflicting, priorities across software development projects. With the help of the project managers, the development manager intertwines the activities of multiple projects to improve the efficiency of the team.

In some organizations the development manager is also responsible for evangelizing to business leadership. While the training role was responsible for evangelizing to the "rank and file," the development manager evangelizes the solutions, the process, and the team to business leadership and customers. This portion of the role is more marketing than development, but is an essential part of ensuring that the business is aware of the value of the software development team.

The development manager also is often the cheerleader that pumps up the development team, encouraging them to remember the vision, to be a part of greatness, and generally be excited about the work that they're doing. Great teams work hard but have fun with each other and believe in what they're creating. It's the development manager's role to create that shared destiny that the group needs to be truly powerful.

In non-software development or consulting companies, the development manager role is often played by the IT management. Often times the IT manager leverages the strengths of a project manager or a solutions architect to fill the needs of the development manager role, retaining tasks such as cross prioritization of projects.

Getting Started as a Development Manager

The development manager role is at the top of the food chain, so to speak. Because of its position at the top of the IT organizational chart, it's a role for which there is generally great competition. One of the most important parts of preparing for this role will be differentiating you from the competition. (See Standing out from the crowd)

People filling the solutions architect role or the project management role have the best chance of moving into a development manager role. The development manager who comes from the solutions architect position generally leads from technical strength. Conversely, the project manager who moves into the development manager role leads from a process strength. Neither is better than the other, however, the approaches to the role will be different. Understanding that both can be successful is important.

To prepare for the development manager role, you should be primarily concerned with learning how to communicate with the business leadership and learning how to adapt software development priorities to match the business goals. Offering to get involved with business management meetings in lieu of the development manager - or in addition to the development manager to take notes or document action items - exposes you to the kind of discussions that the development manager must be able to be effective at.

Communication is more than just one directional, however, while learning to understand the business goals and the impact on the software development process is important, it's also important to be able to communicate with business management about the state of the software development group, good or bad. This includes the state of software development projects. Volunteering to present or prepare status reports helps to encourage this kind of communication.

Ultimately, becoming a development manager is demonstrating these communication skills, gaining the respect of others, consistently delivering, and a little bit of luck since development manager positions don't come around that often.

What's in their Toolbox?

The toolbox of the development manager is less about the kinds of "hand tools" that most other roles are intimate with. The development manager's tools are more about "power tools" and "automated machinery" which the development manager leverages to transform the group. Here are some of the tools the development manager will use:

Methodologies - Development managers have their own methodologies for developing software. The ability to execute that methodology or their methodologies is a great tool. Some development managers may use existing methodologies derived from the Capabilities Maturity Model Integration (CMMI) framework created by The Software Engineering Institute (SEI) at Carnegie Mellon University in Pittsburg. The SEI web site at http://www.sei.cmu.edu/ has a wealth of information about the metrics for ensuring that the processes that are in place at an organization are effective.

Practices - Development managers need more than just broad methodologies they need a practical guide to understanding what does and doesn't work in Software development. The Guide to the Software Engineering Body of Knowledge (SWEBOK) contains the high level overview of software development best practices and is freely available at http://www.swebok.org. The development manager can also use the guide as a tool to identify gaps in the team's knowledge and develop strategies to fill those gaps.

Career Development Process - Development managers are responsible for developing the group. That means a firm understanding of how to develop people in the organization. The development manager may be familiar with one or more than a dozen approaches to developing the careers of the people that work for him (or her).

Scheduling Tools - The development manager is adept at scheduling multiple conflicting priorities and as a result is skilled in the use of a resource scheduling tool. Whether a simple Excel spreadsheet or a deployment of Microsoft Office Project Server 2003, the development manager knows how to ensure that resource management is well taken care of.

Where's the position heading anyway?

Even the staunchest of organizations are beginning to realize that IT is necessary to function and it is software which allows them to leap frog their competition in terms of features and efficiency. Growth for development managers is steady but not at a breakneck pace.

Development managers are being asked to become more a part of the business management to better understand the business drivers that the organization is facing. This brings with it the opportunity to better learn the organization and the industry. This leads to other opportunities outside the software development world.

As technologies become more advanced and the options become greater it will be necessary for the executive management to have a liaison between the business and the technology to help them understand what is possible. The software development manager role is often the key person to fill that role since they have the broadest understanding of how the organizations' systems work.

The Good, the Bad, and the Ugly

The development manager role has a great deal of perks but still a few downsides. Here's the good, the bad, and the ugly:

- **Good: Develop People** - The development manager, more than anyone else, has the ability to help Shepard people through their careers. This can be an immensely rewarding experience.
- **Good: Hiring Authority** - Often the ultimate decision of who to add to the development teams resides with the development manager. It means that more than any other role the development manager can shape the dynamics of the team.
- **Good: Company Impact** - Done right the development manager role can have a huge positive impact on the organization's growth.
- **Bad: Reports** - The development manager is often saddled with creating and presenting reports of various types on the development process. The project manager may help but the development manager often gets stuck presenting the reports to executives.
- **Bad: Responsibility** - The development manager is often the key person responsible for a project's success or failure. If a project fails, it can potentially cause the termination of the development manager's job.
- **Ugly: Letting People Go** - No one likes to admit that they made a mistake, but ultimately someone's got to rectify it. The development manager often gets the unenviable task of eliminating people from the team when they don't achieve

the team's standards.

Conclusion

Development managers come in all shapes and sizes. Each background brings a unique flair to the role. However, learning how to communicate and evangelize to the team, to management, and to the rest of the organization can help you find your way to the development manager role.

Developing for the Web

This section focuses on topics that are particularly important when developing web based applications. The topics here are less about software development in general and more about the specific domain issues that appear when developing software for the web.

Design your Web site to be search friendly

Originally published on Builder.com January 11, 2005

Google just works magically. Searching is a simple magic tool that works without any effort or thought—at least that's what the magicians who develop search engines want you to believe. Realistically, there are some skills that you can learn to help your Web site be a better partner with search engines and help users find the information that they need.

In this article, you'll learn the basics of how to control searching and techniques to both ensure that all of your pages get indexed and how to make indexing more valuable for the users trying to find your information.

Search engines, whether developed internally using a commercial search tool designed to be deployed with a Web site or with a public indexing and searching service such as Google, obey the same basic rules. Developing a site to return meaningful results for the internal search can also provide more relevant results for external searches as well.

Meta tag it

The most basic thing that you have to do to control a search engine, whether internal or external, is to write a Meta tag with the name attribute of ROBOT and a content attribute which contains INDEX or NO INDEX and FOLLOW or NO FOLLOW. This simple tag tells a search engine what it should to do with your page. Both internal and external search engines obey this META tag's instructions on what to do with the page.

<META NAME="robots" CONTENT=" noindex">

INDEX means to include the page in the index the search engine is creating. NO INDEX tells the search engine to not include the page in the index. It is this index that the search engine uses to find user search results. If the page isn't added to the index then it will not be found. A good example of where you might want to use the NO INDEX setting for the ROBOT Meta tag is when you have a discontinued product on your eCommerce site. You still need to keep the product in the catalog so that users can review their orders; however, you don't want anyone just randomly stumbling across the product. Products in the catalog which have not been discontinued would typically have an INDEX setting.

FOLLOW indicates the search engine should follow the links on your page, and NO FOLLOW tells the search engine not to follow links found on the page. The NO FOLLOW setting can be used to prevent search engines from following links that you don't want them to follow such as cases where you're indexing a discussion forum and you don't want your internal search engine to go off and index the links to other sites that might be contained within postings. In other situations, found below, the whole purpose of the page is to provide a set of links for the search engine. In this case, the content will likely be NO INDEX, FOLLOW so that the search engine doesn't index the page itself but does follow the provided links.

Make a list

One of the key challenges for creating a search friendly site is helping the search index know what pages it needs to add to its index. Traditionally a search engine is pointed at the root page in a site and is allowed to wind its way through the site until it has followed every link. This works well for sites that are always outputting their links as anchor (A) tags with Web-based HREF attributes. However, many sites are now using JavaScript-based links to connect one page to another. The result of this is that the search crawler won't be able to follow the links in the site. So the search index may get only a handful of links, which it was able to pick up from normal links on the home page.

The solution is to create a page which contains all of the links that you would like the search engine to follow. This page might include links to all of the products for an eCommerce site or every discussion in a community site. The singular purpose of the page is to try to create a HTML page which contains a large number of anchor (A) tags which lead to all of the content on the site. The page is not special in that it must be written in some specific scripting language. It is only special in that it attempts to quickly provide all of the links needed.

In some cases this technique can be a quick and dirty way of enabling a site index even if the structure of the site itself doesn’t lend itself to that. It is possible to create a program which creates a listing of all of the files on the site which you want indexed by literally walking through the file system or through the IIS virtual directories. By providing a link to each it’s possible to add every page to the search index. This has the negative effect of causing the search index to include pages and files which may have been orphaned from the main site a long time ago.

The search crawler start page is set with a META ROBOT tag which tells the search engine to follow links but not to index the page itself. As we saw above, the contents would be NO INDEX, FOLLOW. Because of this page, the search engine will be able to index every listed page in the entire site.

Some search engines, particularly internal ones, will allow you to point the search engine directly at this page of links. However, there are cases where you don’t have the luxury of controlling the starting point for the crawler. In this case, you need only create a link to your search crawler page on your home page. Because your intent is to allow a search engine to follow the link, you do not need to put any text in the anchor tag. The end result is a link that will help search engines reach your search crawler indexing page without users even knowing that it is present since there is no text to highlight inside of the tags. This might look something like:

<A HREF="./searchcrawler.aspx”></A>

Keep it off the page

Once you have managed to get all of your pages indexed, it’s time to focus on making the search results more meaningful. The first step in this process is to eliminate items on the page which are distracting for the search engine. For instance, menus are not useful to a search engine since they will appear on every page and will contain the same words. Another example is a promotional item which is being recommended to customers based on their interest in a specific product but whose text is not directly related to the focus of the page. Inclusion of these things into the search index only makes searching more difficult because the search term used may occur on every page where the menu exists.

By inspection of the user agent, which is coming in and doing a case-insensitive search for the string ROBOT in the user agent string, it’s possible to determine whether a request is coming from a search engine or not. Although there are some search engines that do not include ROBOT in their user agent strings, most of them do. Once you’ve identified a request as coming from a search engine, you can simply not draw the menus, promotions, and other non-related information found on the page by preventing it from being indexed.

The net result is that the search engine gets only the information that is relevant to the page and doesn’t accidentally index information which isn’t core to the page and therefore return search results to the user which are not useful.

Tag it

As a parting note on search engine optimization be sure to include a specific correct title on all of your pages. Most search engines display the title of the page when it’s in a search results set. Similarly, the use of a META tag with a name of KEYWORDS is useful in encouraging higher ranks for pages when the user searches for those keywords.

Developing in Microsoft .NET

This section deals with specific development opportunities or issues that are available in Microsoft.NET.

Create singleton objects in .NET to improve performance

Originally published on Builder.com February 3, 2005

One singleton can be an extremely valuable tool for enhancing performance. This is particularly true when there is always just one physical object. In traditional programming, you would instantiate a class as an object every time you need a specific instance of an object. For instance, you would create your own product "12" object if you needed access to the information about product "12."

However, in many cases, such as the case in a product catalog, there is one, single persistence of the object in the database, and therefore there only needs to be one instance of the object in memory at one time. In this article, we'll explore how to convert your objects into single instance objects to improve performance and reduce resource usage within the .NET development framework.

The problem with construction

The idea of a constructor for a class has been around since the dawn of object-oriented programming. The object must have a chance to initialize itself before it's handed back to the calling program. This is great since it allows internal data structures to be populated and configured.

The challenge is that the compiler or the runtime environment has already allocated memory for the object. It has, in essence, already created the new object and is just waiting to be configured.

This prevents the constructor from being used with a singleton. The object has already been partially instantiated by the time the code is called and because of this, we have to stop the developer from using the constructor. The constructor must be made protected, private, or internal. Once this is done, you'll be able to call the constructor from within your class, but the outside world won't be able to call it directly.

By creating even one constructor, even if marked protected, private, or internal, the default constructor isn't created. This ensures that no code, except the scope you've specified, can create the object.

The role of static methods

If the developer can't use the constructor, she can't create her own object, which will make using the object very difficult. It's hard to use something that you can't create, which is where static methods come in. Since static methods operate directly off of the class, an instance isn't required. This allows the developer to call the methods before he has an instance of the class.

The role of the static methods becomes the process of looking up some internal cache to determine if the requested instance of the object exists. If the instance exists, the calling program is provided a reference to that instance. If the instance doesn't exist, the static method automatically creates it, stores it in the internal cache, and returns a reference to the object.

The collection

We now know how to prevent a constructor from being created and how to allow developers to create instances through static methods. However, the missing piece is the cache which holds the instances of the class that you want to save and offer up to future calls. That's the role of the collection. The key to the whole program is being able to look up and find previous instances of the class that you've instantiated. Collections can store objects and index them for the lookup the static method will need to do.

A collection is a necessary component of being able to cache the objects, and therefore is a necessary part of creating a singleton class.

Putting it all together

The prerequisite pieces are in place for creating your own singleton class. Listing A shows a Method class, and Listing B shows the MethodCollection class. These two classes cause a single instance of any given data to be created in the Method class. This code, which was extracted from a larger IIS Log Parsing application, has been designed to take the relatively few request methods that are encountered in a log file and prevent literally millions of instances from being created. The Pseudo code comments in the listings are in place to remove unnecessary details, such as database access.

The collection design has two basic retrieval methods. These methods, both of which are GetMethod() overrides, fetch a method by either its integer ID or the string representation. At this level, a null is returned if the method isn't found. On that are layered two GetOrCreateMethod() overrides. These call GetMethod() but create a method if one is not found. This allows the calling code to not worry whether the method exists in the database or not—if not, it is automatically created.

Each of these functions has a corresponding static counterpart. These static methods work on a single global list of methods. This works by a static method which returns or creates a global instance of the Method collection. The static prefixed methods use the global instance and call the methods described above.

The net effect is that the collection can be used directly. This is in case there is some need to have a subset of the methods available, but it also has a global instance, so it isn't necessary for the developer to maintain his own list if he doesn't want to.

On the Method class side, it also supports a set of GetMethod static functions. These are the ones that the MethodCollection calls. These either return the method, or they will return null. An oddity you may see in the Method class is that there's an internal constructor which takes in a data row. This is designed as a performance enhancer. It allows the object to be instantiated with a pre-populated row of data. This isn't a big enhancement with such a small class, but in larger classes with a potentially large number of instances, it allows the creation without another database query and does improve performance.

The challenge of multithreading

For the scope of this singleton class, multithreading would be a problem because it would be possible to create multiple instances of the method name as objects—even in the database. In many cases creating a singleton doesn't require much thought towards multithreading; however, when it does, you'll need to add your own critical section handling to prevent two simultaneous attempts from running the same addition code, which will prevent the addition of duplicate records.

Listing A

```
using System;
using System.Data;

namespace LogParser.WebLog.Values
{
    ///
    /// The request method (GET, PUT, SEARCH, etc.)
    ///
    public class Method // Pseudo Code: DataAccessBase base
class reference
    {
        ////////////////////////////////////////////////////////////////////////
        // Constructors
        ////////////////////////////////////////////////////////////////////////
        #region Constructors
        ///
        /// Creates a method from it's ID
```

```
///
/// The unique ID # for the method
internal Method(int ID)
{
	_ID = ID;
	// Pseudo Code: Read();
}

///
/// Creates a method from it's name
///
/// The name of the request method
internal Method(string method)
{
	_method = method;
	// Pseudo Code: Commit();
}

///
/// Creates a method object from a data row
///
/// The row to create the method object from
internal Method(DataRow dr)
{
	_ID = (int) dr["ID"];
	_method = (string) dr["method"];
}

#endregion

////////////////////////////////////////////////////////////////////
// Fields and Properties
////////////////////////////////////////////////////////////////////
#region Fields and Properties
///
/// The ID number for the record
```

```
        ///
        protected int _ID;
        ///
        /// Public accessor for _ID
        ///
        public int ID
        {
                get { return (_ID); }
        }

        ///
        /// The string description of the method
        ///
        protected string _method;
        ///
        /// Public accessor for _method
        ///
        public string MethodName
        {
                get { return (_method); }
                set { _method = value; }
        }
        #endregion

        ////////////////////////////////////////////////////////////////////////
        // Static Methods
        ////////////////////////////////////////////////////////////////////////
        #region Static Methods
        ///
        /// Returns a method given the ID
        ///
        /// The unique ID for the method object being
requested
        /// Method object if found, null otherwise
        public static Method GetMethod(int ID)
        {
            // 0 is an invalid ID
            if (ID == 0) return (null);

            // Pseudo Code: Get rows from database,
            // Pseudo Code: if a row is found call
constructor, else return null

            Method method = new Method(); // Provide
row from database call as parameter
            MethodCollection.StaticAdd(method);
            return(method);
        }

        ///
        /// Returns a method object for the given name or
null if not found
        ///
        /// The method name to locate
        /// The method object for the methodName if found,
null otherwise
        public static Method GetMethod(string
methodName)
        {
            // Some quick error checking
            if (methodName == null) return (null);
            if (methodName.Length == 0) return (null);

            // Pseudo Code: Search the DB for the
method
            // Pseudo Code: If one record, call
constructor and add to static list
            // Pseudo Code: else return null
            return(method);
        }

        #endregion
```

```
    }
}
```

Listing B

```
using System;
namespace LogParser.WebLog.Values
{
    ///
    /// Summary description for MethodCollection.
    ///
    public class MethodCollection // Pseudo Code: Reference
any base collection you have or
                  // Pseudo Code:
System.Collections.Specialized.NameObjectCollectionBase
    {
        ////////////////////////////////////////////////////////////////////////
        // Constructors
        ////////////////////////////////////////////////////////////////////////
        #region Constructors
        ///
        /// Create a method collection
        ///
        private MethodCollection()
        {
            _idMethods = new FrameworkCollection();
        }
        #endregion

        ////////////////////////////////////////////////////////////////////////
        // Fields and Properties
        ////////////////////////////////////////////////////////////////////////
        #region Fields and Properties

        // Pseudo Code: Create the _idMethods as a
collection that supports string keys
```

```
                // Pseudo Code: and object values
                #endregion

                ////////////////////////////////////////////////////////////////////////
                // Indexers
                ////////////////////////////////////////////////////////////////////////
                #region Indexers
                ///
                /// Returns the method item given the methods ID
                ///
                public new Method this[int ID]
                {
                        get {
return((Method)(base.getItem(ID.ToString()))); }
                }

                ///
                /// Returns the method item given the method's
name
                ///
                public new Method this[string methodName]
                {
                        // Fetch the ID from the names collection
                        get {
return((Method)(base.getItem((string)(_idMethods[methodName])
))); }
                }

                #endregion

                ////////////////////////////////////////////////////////////////////////
                // Methods
                ////////////////////////////////////////////////////////////////////////
                #region Methods
                ///
                /// Adds a method to the internal collection
```

```
        ///
        /// The method object to add
        public void Add(Method method)
        {
            base.Add(method.ID.ToString(), method);
            _idMethods.Add(method.MethodName,
(object) method.ID.ToString());

        }

        ///
        /// Get a method from the collection or database if
necessary
        ///
        /// The ID of the method to locate
        /// The method object if found, null otherwise
        public Method GetMethod(int ID)
        {
            Method method = this[ID];
            if (method == null)
            {
                // Not found, check DB
                method = Method.GetMethod(ID);
                if (method != null) Add(method);
            }
            return (method);
        }

        ///
        /// Gets a method from the collection or database if
necessary
        ///
        /// The name of the method object to create
        /// The method object matching the methodName if
found, null otherwise
        public Method GetMethod(string methodName)
        {
            Method method = this[methodName];
            if (method == null)
            {
                // Not found, Check DB
                method =
Method.GetMethod(methodName);
                if (method != null) Add(method);
            }
            return (method);
        }

        ///
        /// Gets or creates a method based on the
methodName
        ///
        /// The name of the method to get or create an object
for
        /// The method object read from the database or
created
        public Method GetOrCreateMethod(string
methodName)
        {
            Method method =
GetMethod(methodName);
            if (method == null)
            {
                // Not found, create
                method = new
Method(methodName);
            }
            if (method != null) Add(method);
            return (method);
        }

        #endregion
        //////////////////////////////////////////////////////////////////////
        // Static Methods
        //////////////////////////////////////////////////////////////////////
        #region Static Methods

        private static MethodCollection _methodCollection;
        ///
        /// Returns the global instance of the
methodcollection object
        ///
        /// The global methodcolletion object
        private static MethodCollection
GetGlobalInstance()
        {
            if (_methodCollection == null)
            {
#if DEBUG
                Console.WriteLine("DEBUG:
Creating global MethodCollection instance");
#endif
                _methodCollection = new
MethodCollection();
            }
            return (_methodCollection);
        }

        ///
        /// Get a method by ID
        ///
        /// Get the method by ID
        ///
        public static Method StaticGetMethod(int ID)
        {
            return
(GetGlobalInstance().GetMethod(ID));
```

```
        }

        ///
        /// Get a method by it's name
        ///
        /// The name of the method to fetch
        /// The method object matching the string
        public static Method StaticGetMethod(string
method)
        {
            return
(GetGlobalInstance().GetMethod(method));
        }

        ///
        /// Gets or creates a method based on the method
string
        ///
        /// The text name for the method
        /// A method object representing the method string
provided
        public static Method
StaticGetOrCreateMethod(string method)
        {
            return
(GetGlobalInstance().GetOrCreateMethod(method));
        }

        ///
        /// Adds a method to the global instance collection
        ///
        /// Method to be added
        public static void StaticAdd(Method method)
        {
            GetGlobalInstance().Add(method);
        }
```

```
                #endregion
        }
}
```

More efficient coding with advanced user control caching in .NET

Originally Published on Builder.com January 25, 2005

If you've not yet learned about user controls, you’re missing a powerful feature in ASP.NET. User controls are page fragments that can be inserted into Web pages just like a Web control. The difference is that user controls retain their visual design in Visual Studio. This means that you can create reusable controls without having to try to write the HTML output yourself (as you would with a Web control).

What’s more, the user control supports built-in caching, which can help improve performance by storing a pre-rendered copy of the control’s output. This means that the control can be displayed without even running the code. This can greatly improve performance, especially when the same information is hit repeatedly.

How to cache user control

The process of caching a user control in .NET is very simple. You put an OutputCache directive at the top of the user control’s ASCX file. This directive takes a duration attribute that indicates the length in minutes that the control will be cached. This alone is a great feature of ASP.NET; however, the functionality goes much farther by allowing for multiple copies of the control to be cached based on the parameters that the user control receives.

The duration attribute is joined by a VaryByParam, VaryByControl, or VaryByCustom. VaryByParam creates different cached copies of the control based on the query string and form post variables provided to the form. The VaryByControl attribute allows you to enter a string that will be used to differentiate copies of the control in the cache. The VaryByCustom attribute causes ASP.NET to call HttpApplication.GetVaryByCustomString to return the string that will be used to separate one cached copy of the control from another.

An OutputCache directive without any variance in parameters will look like this:

<%@ OutputCache Duration="60" VaryByParam="none" %>

An easy way to test the directive is to create a user control that displays the current time, copy it into two different user controls, and put the OutputCache on one of the two controls. Once you've done this, you can assemble a page that contains both controls. If one of the controls doesn't update the time when it's loaded the second time, you know that it's been cached.

Of course, varying by parameter is fine when you are getting the parameters from a query string; however, if the parameters to change the output of the control are not coming from a query string, this will not work.

Vary by custom

The designers of ASP.NET decided, wisely, that they may not be able to anticipate every method of determining which cached version of a control developers might want. Because of this, VaryByCustom was added. It allows you to vary, for instance, the cached copy to display information from a user's profile. It might be the list of benefits that a member of a particular class should receive. A control that displays those benefits would need to know what class the user is, but this information is probably not going to be in the query string or a part of a form post. In order to control the different cached copy of the control, you'll have to resort to the VaryByCustom attribute and set up a GetVaryByCustomString() member of the global class.

The member function uses two parameters, the first being the HttpContext of the request that generated the call. This is necessary if there are variables in the session that must be considered, such as a user profile object. The second parameter is the argument string provided as the value for the VaryByCustom attribute in the control. This string is the complete value, not broken up or processed in any way. The expected return value is a string that will uniquely identify this particular instance of the cached copy of the control.

Your code should evaluate the incoming argument and provide the appropriate return string. For instance, the code in Listing A searches the user's session for the variables listed in the VaryByCustom attribute and returns the values of those session values as a part of a custom string:

Clearing the cache

Most of the time, the caching that you set up may be appropriate, but there may be times, particularly when you're trying to debug problems, when waiting for the cache to expire may be painful (that is, at least if you're as impatient as I am). For that reason, it's possible to clear the output cache for a user control (as well as a page). The Response.RemoveOutputCacheItem() is the method that will remove the item from the cache. However, there's a catch. RemoveOutputCacheItem only takes one parameter—the path of the item(s) to remove. Although the static method indicates by its name that it's clearing one item, in reality it clears all of the cached copies of the user control (or page) for a given path.

While this is typically not a problem in clearing items for debugging, it can be potentially challenging if you were expecting to use the function to clear a single version of the control. That control may have a few hundred different cached copies, and each of them would have to be regenerated.

Listing A

```
override public String GetVaryByCustomString(HttpContext
context, String arg)
{
  string[] splitString = arg.Split(',');
  StringBuilder returnString = new StringBuilder();


    foreach(string str in splitString)
    {
      string singleValue = context.Session([str.Trim()]);
      if (singleValue != null)
       {
        returnString.Append("~").Append(singleValue).Append("~");
       }
     else
        {
         returnString.Append("~~");
        }
      }
  return (returnString.ToString());
}
```

Application Development: Calling a COM object from a Web service in .NET

Originally published on Builder.com February 18, 2004

Creating a Web service in .NET is easy. Setting up a call to a COM object in .NET is very easy as well. However, putting the two together can be difficult when the COM object uses a single-threaded apartment model. You see, Web services are inherently defined by a multithreaded apartment model. However, making a Web service call a single-threaded, apartment-model COM object is not impossible; I will show you how.

What is a threading model?

Windows, like most operating systems today, allows you to have multiple programs running at the same time. Every program is essentially a process (or in rare cases, a collection of processes). A process is a collection of memory address space containing code and data, as well as other system objects and handles that a process may need. Every process in Windows has the ability to have multiple threads of execution. A thread of execution, or more simply a thread, is a path of execution in the code. When the process starts, one thread is started as the main function for the program. From there, the main thread can spawn other threads as needed. Most programs are developed for only a single thread of execution. In other words, the main thread does not ever spawn any new threads.

Another way that threads are spawned is if multiple requests come in for the same type of object. For instance, when a second request is executed to a Web server, a new thread is spawned to service that request. With a Web service, each new request to the Web service is automatically spawned as a new thread in the Web service process. Because of this, Web services are inherently multithreaded.

In COM, the way that a COM object handles multiple threads is important, so you should be clear on how the outside world should communicate with the object. Most COM objects developed by programmers in languages like Visual Basic and Delphi are single-threaded. In other words, they do not internally understand the concept of multiple threads operating at the same time within the process.

But the options for COM objects do include single-threaded or multithreaded. Basically, a single-threaded COM object will enforce the running of only one thread of execution at a time. A multithreaded object will allow multiple threads to run at the same time. Free-threaded, which you may have heard of, is really another name for multithreaded objects. There is no restriction that you must use the same kind of threading model for every object.

Calling a COM object from .NET

Calling a COM object from .NET is simple enough. You set up the reference in the project and VS.NET creates an interop object for you to use to call the COM object. From there, it is just like creating any other object from a native .NET class. This works well when the threading model of the COM component matches the threading model of the .NET application. However, when this is not the case, you will receive an error when you try to instantiate the object.

The solution is to call the COM object from a thread that has the same threading model as the COM object itself. In cases like a Web service, where you cannot control the threading model of the main thread, you will have to spawn a new thread using the System.Threading namespace. In the new threads that you create, you have the option of controlling the threading model so you can match it to the COM component.

To create a thread, you will first have to create a ThreadStart delegate. If you are unfamiliar with delegates, they are essentially function pointers. Rather than using an actual function name, you can assign a function name to a delegate and then use that delegate to perform the operation. The ThreadStart is, as the name implies, the starting point for the thread. You are telling .NET where to start executing code.

The ThreadStart delegate is for a function that takes no parameters and returns no parameters. In order for this to be of much use, you will want to encapsulate the function in an object (technically making it a method call). In this way, you can initially create an object, stuff the data the thread will need into the object, and then have the method that you use for ThreadStart process the data that is already in the object.

Unwinding the threads when done

Before I show you a fully functional code listing, there are two things that you should know about threads. First, threads may end at any time. They are running independently of one another. However, in most COM object calling situations, you will eventually want to wait on the COM object to finish its work before proceeding. On a Thread object, that is the purpose of a Join() method call, which pauses the current thread (the main thread, in our case) until the thread being referenced has stopped. This is a simple way of synchronizing the operation of one thread to another.

The second important point about threads is that they exist within their own exception space. Exceptions thrown in a thread are not sent back to the thread that spawned them. As a result, you will need to wrap your code in a try/catch block and store the thrown exception in the object that you wrapped the function call in. Then, the calling thread can check to see if the thread has completed

(Thread.IsAlive == false) and then check if there is a value other than null in the internal exception. If there is a value, then you know that your thread threw an exception. If not, then there were no problems.

Seeing it in action

Rather than showing a Web service that would require either using the automatic Web interface or using a specialized tool to call the service, I created a simple console application which performs the exact same steps that you would perform in a Web service.

Listing A (Class1.cs) shows a simple console application class that creates 60 threads by creating 60 instances of a shell class and then creates a thread that starts in the ThreadStart delegate of each of those classes in turn. It sets the threading model for the thread that was created to STA, or single-threaded apartment. This allows even a Web service to call STA COM objects. Once all of the threads are created, it iterates through the array of threads and joins to each one to ensure that it closes properly.

Listing B (ShellClass.cs) is the class that each thread instance lives in. It contains a holder for any exception that is thrown (which the console application ignores). It also contains some basic trace statements and a creation of our COM object. In this case, the object is a Delphi object whose only purpose is to expose the Sleep method; however, any STA COM object will work.

Tying up

Getting a STA COM object to run in an MTA architecture is not straightforward; however, it can be done safely if you are willing to get tied up with a few threads.

Listing A

```
using System;
using System.Threading;
using System.Collections;


namespace ConsoleApplication1
{
    /// <summary>
    /// Summary description for Class1.
    /// </summary>
    class Class1
    {
        /// <summary>
        /// The main entry point for the application.
        /// </summary>
        static void Main(string[] args)
        {
            Thread[] Threads = new Thread[60];


            Console.Write(DateTime.Now.ToLongTimeString()
+ " Program Starting \n");


            for (int Looper=0;Looper < 60;Looper++)
            {
                Console.Write("Spawning Thread " + Looper +
"\n");
                Threads[Looper] = CreateThread(Looper);
            }


            for (int Looper=0;Looper < 60; Looper++)
            {
                // Threads[Looper].Join();
            }
            Console.Write(DateTime.Now.ToLongTimeString()
+ " Program Ending \n");
        }


        static Thread CreateThread(int Looper)
        {
            ShellClass myObj = new ShellClass(Looper);


            Thread myThread;
            myThread = new Thread(new
ThreadStart(myObj.Go));


            myThread.ApartmentState = ApartmentState.STA;
            myThread.Start();

           return (myThread);
        }
    }
}
```

Listing B

```
 using System;

namespace ConsoleApplication1
{
    /// <summary>
    /// Summary description for ShellClass.
    /// </summary>
    public class ShellClass
    {
        public int _ID;
        public Exception myExcept;


        public ShellClass(int vID)
        {
            _ID = vID;
        }


        public void Go()
        {
            try
            {
                DateTime dtStart = DateTime.Now;

Console.Write(DateTime.Now.ToLongTimeString() + " Starting
thread " + _ID + "\n");


                COMTest.TestObjectClass myObj = new
COMTest.TestObjectClass();


                myObj.Sleep(1000);



Console.Write(DateTime.Now.ToLongTimeString() + " Ending
thread " + _ID + "\n" +
                    "Thread took " +
DateTime.Now.Subtract(dtStart).ToString() + "\n");
            }
            catch (Exception except)
            {
                myExcept = except;
            }


        }
    }
}
```

Read binary files more efficiently using C#

Originally published on Builder.com August 28, 2003

It would be nice to think that everything file transfer has gone to XML; to believe that every file format you encounter today is just another XML schema to understand. But that's not the case. There are still a large number of file formats that aren't XML, or even ASCII. Binary files are still flowing across networks, being stored on disks, and passing between applications, and they’re doing it more efficiently than text files.

In C and C++ reading a binary file was easy. Except for a few carriage return/line feed problems, every file that was read into C/C++ was a binary file. C/C++ really only knew about binary files and how to make a binary file look like a text file. As the languages that we worked with got more and more abstract, we ended up with languages that couldn’t directly and easily read the created files. The languages wanted to automate the process of streaming out data—each in its own unique way.

Defining the problem

In many areas of computer science, C and C++ are still used to store and retrieve data directly from structures of data. It is very simple to read and write to a file from structures in memory in C or C++. In C, all you do is hand fwrite() a pointer to your structure, tell it how many are there, and how long the structure is. It writes it directly to the file in a binary format.

This put the structure into a file and meant that reading the file, if you knew the right structure, was easy as well. You passed fread() the file handle, the pointer to the structure, how many to read, and how long the structure was. The fread() function did everything else for you. Suddenly the structure was back in memory. There was no parsing, no object models—the file read directly into memory.

The two biggest problems to be addressed in the C/C++ days were structure alignment and byte swapping. Structure alignment just meant that sometimes a compiler would skip bytes in the middle of a structure because it would be suboptimal for a processor to access those bytes. So the compiler optimized for speed by skipping bytes and reordering the order of the fields. Byte swapping, on the other hand, referred to the process required to rearrange bytes in a structure due to a potential difference in the way that processors ordered bytes.

Structure alignment

As processors have been able to process more information at one time (within a single clock cycle), they've begun to expect that the information they process be lined up in a certain way. Most Intel processors expect that integers (of the 32-bit variety) will align along a 4-byte boundary. They won't work with integers that don't exist in memory at an address that isn't a multiple of four. Compilers know this. So when presented with a structure that would cause an integer to not be lined up on an address that is a multiple of four, compilers have three choices.

First, they can choose to add some nonusable white space into the structure so that the starting address for the integer is a multiple of four. This is the most common implementation. Second, they can rearrange the fields so that the integers are all aligned on a multiple of four. Because this causes some other interesting problems, it's less frequently used. The third option is to allow an integer to be in the structure in a nonmultiple of four and then put code in place to move the integer to and from a scratch space which is a multiple of four. This involves a little extra overhead with each reference, but can be useful when being compact is very important.

For the most part, these are compiler details that you don't worry about. If you're using the same compiler with the same options for both the program that writes the data and the program that reads the data, there should be no problems. The compiler will process the same structure the same way and all will be well. But when you're involved in cross-platform file conversion, it's important to align everything the right way so that information can be transferred. In contrast, some programmers learned how to get the compiler to leave their structures alone.

Byte swapping: Big endians versus little endians

Big and little endian refers to two different ways that an integer can be stored in a computer. Since an integer is typically more than one byte, the question becomes whether the most significant byte is the one that's read and stored first. The least significant byte is the one that changes most frequently. That is, if you continually add one to an integer the least significant byte changes 256 times as frequently as the next least significant byte.

Different kinds of processors store integers differently. Intel processors typically store integers in little endian format, in other words, little end first. Most other processors store integers in big endian format. So when a binary file is read and written on a different platform, there's the possibility that you'll have to flip the bytes around to get the correct order.

This was and still is particularly a problem on UNIX where some variants run on a Sun Sparc processor, some on an HP processor, others on an IBM Power PC, and some on Intel-based chips. Moving from one processor to another means learning when bytes must be swapped so they end up in the order that the processor of the local system expects them.

Challenges with binary files in C#

There are two additional challenges with C# and binary files. The first challenge is the challenge that all .NET languages are strongly typed. So you'll have to convert a stream of bytes from the file into the data types that you want. The second challenge is that some data types are more complex than they appear on the surface and may need some conversion.

Type breaking

Because .NET languages, including C#, are strongly typed, you can’t just arbitrarily read a number of bytes from a file and jam it into a structure. You’ll have to start by reading the number of bytes you need into an array of bytes and then copying them over to the structure while breaking the type casting rules.

Searching back in Usenet archives, you’ll find several postings in microsoft.public.dotnet hierarchy with a set of routines that will allow you convert any object into a series of bytes and back to the object again. They appear here in Listing A.

Complex data types

In C++, you know what is an object, what is an array, and what isn’t either an object or an array. But in C#, things aren't as simple as they seem. A string is an object; so is an array. Because there are no true arrays and because there is no fixed size of many objects, there are some complex data types that don’t fit neatly into fixed binary structures.

Fortunately, .NET offers a way to resolve this issue—you can tell C# how you want your strings and other types of arrays handled. This is accomplished via the MarshalAs attribute. In the case of a string in C#, the attribute should go immediately above the member to be controlled and should look like this:

[MarshalAs(UnmanagedType.ByValTStr, SizeConst = 50)]

The SizeConst parameter should be changed with the length of the string that you want as it will be stored or retrieved from the binary file. This fixes the string length at some maximum.

Solving classic problems

Now that you know how the .NET-introduced problems are solved, it's time to see how easily the classic binary file problems are solved.

Packing

Instead of using compiler options to control how structures are arranged, you can assign a StructLayout attribute to a structure to explicitly state how you want that structure arranged or packed. This is particularly useful when you need different structures to be packed differently. It's much like packing your car. Using the StructLayout is like carefully deciding whether you want to pack everything tightly or if you want to just throw it in and hope it works out. The StructLayout attribute should look like this:

[StructLayout(LayoutKind.Sequential, Pack = 1)]

This causes the layout of the structure to ignore alignment boundaries and pack the structure as tightly as possible. This should correspond with any structure you're reading from a binary file.

You may find that even adding this attribute doesn't completely resolve the issues with your structure. In such cases, you'll probably have to tediously work through the issues by trial and error. One of the reasons we've moved away from binary data structures, particularly for cross platform, are the subtle problems that can be caused by the way different computers and compilers handle things at the binary level. .NET is good at adapting to other binary files, but it's not perfect.

Endian flipping

One of the classic problems with reading and writing binary files is that some computers store the least significant byte first (e.g., Intel) and others store the most significant bit first. In C/C++, you had to manually address this problem by flipping each field one-by-one. One of the great things about the .NET Framework is that the code has access to the metadata for types at runtime so you can read that information and use it to automatically address the endian problem for every field in a structure. The code in Listing B is a basic example of how this can be done.

Once you get the object’s type, you can get the fields within the structure and then proceed to check each one to determine whether it’s an unsigned integer of 16 bits or 32 bits. In either of these cases, the bytes are swapped so, you can swap them back by masking off a byte at a time and rotating it to its new position, and then add everything back together.

Notice that you don’t do anything with strings. Strings aren't affected by the big endian/little endian discussion. Those fields are left unaffected by the flipping code. You also flip only unsigned integers. This is because negative numbers aren't always represented the same way on every system. There is the ones compliment notation for negative numbers and the more popular twos compliment. This makes fixing negative numbers cross platform slightly more difficult. Luckily, negative numbers are rarely communicated in binary files.

Just to make things interesting, floating point numbers are sometimes represented in nonstandard ways as well. Although most systems have settled on the IEEE format for floating point numbers, there are a few, particularly older systems, that use other formats.

Overcome resistance

You can make C# read binary files despite its initial resistance. In fact, C# can be a better language to read in binary files because of the way that it maintains accessible metadata about the objects it works with. Because of this, C# can automatically fix byte swapping problems throughout an entire structure.

Listing A

```
 public static byte[] RawSerialize( object anything )
 {
int rawsize = Marshal.SizeOf( anything );
IntPtr buffer = Marshal.AllocHGlobal( rawsize );
Marshal.StructureToPtr( anything, buffer, false );
byte[] rawdatas = new byte[ rawsize ];
Marshal.Copy( buffer, rawdatas, 0, rawsize );
Marshal.FreeHGlobal( buffer );
return rawdatas;
 }
 public static object RawDeserialize( byte[] rawdatas, Type
anytype )
{
int rawsize = Marshal.SizeOf( anytype );
if( rawsize > rawdatas.Length )  return null;
IntPtr buffer = Marshal.AllocHGlobal( rawsize );
Marshal.Copy( rawdatas, 0, buffer, rawsize );
object retobj = Marshal.PtrToStructure( buffer, anytype );
Marshal.FreeHGlobal( buffer );
return retobj;
 }
```

Listing B

```
public static object EndianFlip(object oObject)
{
        string sFieldType;
        Type tyObject = oObject.GetType();
        FieldInfo[] miMembers;
        miMembers = tyObject.GetFields();
        for (int Looper = miMembers.GetLowerBound(0);
                Looper <= miMembers.GetUpperBound(0);
                Looper++)
     {
            sFieldType = miMembers[Looper].FieldType.FullName;
            if ((String.Compare(sFieldType, "System.UInt16", true)
== 0))
            {
                ushort tmpUShort;
                tmpUShort = (ushort)
miMembers[Looper].GetValue(oObject);
                tmpUShort = (ushort) (((tmpUShort & 0x00ff) << 8)
+
                    ((tmpUShort & 0xff00) >> 8));
                miMembers[Looper].SetValue(oObject, tmpUShort);
            }
            else
            if (String.Compare(sFieldType, "System.UInt32", true) ==
0)
            {
                uint   tmpInt;
                tmpInt = (uint)
miMembers[Looper].GetValue(oObject);
                tmpInt = (uint) (((tmpInt & 0x000000ff) << 24) +
                          ((tmpInt & 0x0000ff00) << 8) +
                          ((tmpInt & 0x00ff0000) >> 8) +
                           ((tmpInt & 0xff000000) >> 24));
                miMembers[Looper].SetValue(oObject, tmpInt);
            }
        }
          return (oObject);|
}
```