Application Platform

Take the "C" out of "CRM" and you're left with Microsoft Dynamics Relationship Management—and that's the right way to think about this product. Microsoft Dynamics CRM is not just about sales and marketing. It's a platform for developing line of business applications—applications that manage and track information and processes around realworld objects. The object could be a customer, but it could also be a grant, building, or a potential candidate for hire. The key question is: "do I need to track the information and activities related to this entity?" If the answer is "yes," Microsoft Dynamics CRM deserves some serious consideration.

The Real Story Behind Microsoft Dynamics CRM

Imagine that Microsoft Dynamics CRM is an iceberg. The part you see, the CRM application itself, sits above the water and represents only about a fifth of the total size of the iceberg. But 80% of the power and potential of Microsoft Dynamics CRM lies beneath the surface—in the platform layer and the application framework. Once you know the real story behind Microsoft Dynamics CRM, you'll understand how true this is. We can tell you that story,

because we were there.

Originally, Microsoft's plan was to create not a product, but a development platform. In fact, at the very beginning, it was going to be just a platform and API's. Microsoft wanted to build a horizontal platform that developers could use to create their own CRM systems. By providing a really solid, fully-featured platform that was easy to develop on, organizations could create powerful CRM systems tailored to their own needs, and third-party companies could create deep, vertical market CRM applications that they could host or sell.

Microsoft knew they needed a way to showcase this platform. As it so happened, Aaron Elder and I were part of a team that had built an award-winning product called iCommunicate that fell in the same vein. Seeing the product, Microsoft Corporation decided to acquire the company and, more importantly, acquire us as employees, in 2001. So out we moved to Redmond, WA to join the Microsoft CRM development team.

To demonstrate the capabilities of this platform, the team developed a

reference or sample application for general customer relationship management. The sample's goal was to show what a CRM solution built on the platform might look like. As it turned out, the sample application wowed even those on the team. At the same time, it was becoming increasingly apparent that what the world really needed was a 2-tiered approach: a configurable customer relationship management application with a deeply functional platform underneath. We were most of the way there. The sample application acted as a prototype while we built Microsoft Dynamics CRM as a fully featured product.

All the while the core goal of developing a great platform remained. Instead of building an application and its underlying platform, we built a platform and framework with an application on top of it. That's why Microsoft Dynamics CRM remains such a flexible and powerful development tool for line of business applications. This development approach and strict policy of keeping the platform pure is why the product deserves consideration as a business application platform.

Microsoft Dynamics® CRM as a Business Application Platform

Not Just Sales, Not Just Customers

Because it was built as a platform, Microsoft Dynamics CRM had to be metadata-driven. While the application portion of the system includes a Sales module, a Marketing module, and a Service module, the underlying database simply contains a set of entities, each with certain characteristics and relationships. What is called an "opportunity" in the application and is used to track potential sales opportunities could just as easily be called a "proposal" and be used to track proposals and manage grant applications. What is called a "contact" could be a patient in a health and human services system. In fact, the platform allows you to change the names of these core entities to accommodate the function they will perform in your business.

Business applications typically track information about people, money, products, or services; the activities and communications done around them; and the relationships between them. Microsoft Dynamics CRM was designed to capture exactly that kind of information. The people Microsoft Dynamics CRM tracks may all be internal to your organization, rather than customers. Or they might be citizens coming to your public sector agency. Or a combination of patients, doctors, and staff. It doesn't matter. They are people with relationships and around whom actions are being performed and tracked. Here are just a few examples of success stories from organizations for which we've built applications using Microsoft Dynamics CRM as the platforms for non-sales purposes:

A large, state, government-based organization uses Microsoft Dynamics CRM to manage its teacher certification program. In the system, "contacts" are educators. Each educator has areas in the system for tracking their schooling, work history, and certifications. The system also contains a portal so that educators can update their personal information, pay certification fees online, and so on.

A large, financial institution wanted a system to manage the process of reviewing and interviewing candidates. Using Microsoft Dynamics CRM, we're building a system that not only keeps track of the candidates and their personal documents (resumes, cover letters, recommendations), but also manages and schedules all of the activities during the recruitment process, such as the initial screening, the manager review of the resumé, the interview, and so on.

A major command within the United States Air Force uses Microsoft Dynamics CRM to receive, disseminate, and track the chain of organizational tasks and resulting sub-tasks within their organization and from Air Force Headquarters at the Pentagon. In this system, "accounts" are organizations and "contacts" are users and action officers charged with the coordination and response to policy, procedures, Congressional requests, requests under the Freedom of Information Act, and other Air Force business matters.

The Agile Organization

Choosing a platform isn't just about solving today's problems. It's about building for the future. The platform you choose will be with you for some time to come, so you need to make sure you have a solid foundation on which to build your business—one that is able to respond as the business changes.

It's critical these days that businesses respond quickly to change. Therefore, organizations prefer not to spend years developing a line of business (LOB) application. Solutions need to be implemented in months, not years. You can't spend a year rolling out a CRM system. You can't spend a year rolling out an asset management system. You can't spend a year rolling out any kind of business application. If you spend six months trying to figure out what to build and another six months building it, the business you deployed it for will have already changed. You need to develop quickly, but you still need to develop a solid, usable, and fully-featured system.

It's also important to get the application out there quickly so you can get people using it and get their feedback. Then you can change, improve, and release again. That kind of development allows your organization and your users to be productive more quickly, to develop more usable systems, and gets your users invested in the tools as well as their success.

Selecting a platform that allows for rapid development and an iterative process is one of the keys to building a successful long-term LOB strategy. The right platform helps ensure business agility.

It Doesn't Pay to Grow Your Own

Often, people in organizations say to themselves, "Our company is so unique and specialized that nobody will have a system that addresses our business. We'd better build our own system—or get someone else to build a custom system for us."

avanade®
Results Realized

Microsoft Dynamics® CRM as a Business Application Platform

It's true that every company is different. Quite often, an off-the-shelf product won't meet your needs. But the mistake these companies make is that they don't take into account the total cost of ownership for the system over the lifetime of the company.

Building completely from scratch is a costly prospect. You have to design everything from the ground up and build it out. Building the security system incorrectly means opening your company to attacks. Building a UI framework that accommodates all the future possibilities is another challenge. And once it's built, you have to maintain it, update it, and of course, extend it. All on your own, and all at your ongoing expense.

When we meet with potential clients, the IT folks sometimes tell us, "Well, we'll build a platform that can be used for the long haul. We can build our own framework and build it so it can be reused for future applications." It's good in theory, but rarely occurs in practice. It's hard and expensive to build a general-purpose platform. Furthermore, developers leave. The architects of the original platform may not be around when you go to build your second or third LOB application. When that corporate knowledge departs, it can be time-consuming and costly to replace.

And lastly, if you've worked with or managed developers—or you are one—you know that nobody's code is ever as good as your own. Developers hate maintaining other people's code. They'd much rather write something new. Your developers will find all sorts of reasons why the current platform won't work or won't work well enough for the new application. Selecting an external platform settles the debate. You've got a solid, all-purpose platform that will

be maintained by a dedicated group of external developers. So, your devs can write all the cool new code they want for the LOB applications your business depends upon, and leave the maintenance to Microsoft.

Keys to the Family Car
Keep in mind that Microsoft Dynamics CRM is the platform that the Microsoft development team used to build the CRM application that you see. Microsoft's CRM development team uses the same platform and the same Software Development Kit (SDK) as you will. That's not the case with all products. A more typical approach for companies that build CRM products is a closed or black box approach in which the product is self-contained. When customers inevitably come back wanting to extend and hook into the system, the CRM manufacturer creates an SDK that provides access to 20% or 30%—maybe even as much as 50%—of the functionality of the core product. Not enough. Oh, and that's for external use only. The internal developers don't even use the SDK.

In the case of Microsoft Dynamics CRM, the SDK you use is the exact same web services SDK that is used internally by Microsoft's developers. And, of course, the platform is the same. So, in theory, if you had the same money, caliber of people, and all the specifications for the Microsoft Dynamics CRM application, you could actually build the exact same product as Microsoft did. If the product out of-the-box is not what you want it to be, it's infinitely modifiable and extensible at virtually any layer. It's like Microsoft just handed you the keys to the family car. You are not relegated to Microsoft's vision of what CRM was when it was built. You are free to drive off and create the future of what Microsoft Dynamics CRM can be.

The Plumb ing and the fixtures
At a very high-level, building business applications requires creating two major elements: the core plumbing and the custom-tailored functionality. The plumbing includes the data model, the security model, the relationships between elements in the system, the presentation layer, and so on. The functionality includes the specific business processes and problems that the system addresses and the particular ways that the system addresses them through the user interface.

In most cases, the plumbing can be the same for all the LOB applications in an organization. If it's done right, you can use the same data model, security model, UI paradigm, and so on to build your HR management system, your recruitment management system, and your professional services automation system. In fact, there are some big advantages to using the same plumbing for all your internal applications—but more about that later.

The functionality, on the other hand, varies for each application. You'll be storing and presenting different information to the user. You'll have different business processes and neds tha each system solves. In every case, the user interface, the features you expose, and the requirements you implement have to work for that particular application's users, as well as for the organization.

The plumbing is an underlying layer that, if done well, can be written once and reused for all your applications. The functionality is a highly-visible layer that must be customtailored to fit the organization and its users' needs. The functionality is where the majority of your development time

Microsoft Dynamics® CRM as a Business Application Platform

and energy should be spent, because it's a highlytailored area and getting that piece right is the key to success.

The functionality is where the majority of your development time and energy should be spent... Unfortunately, most organizations spend most of their time on the plumbing...
They waste their money reinventing the wheel.

Unfortunately, most organizations spend most of their time on the plumbing. They develop their LOB applications separately and at separate times. The platform is often reinvented or rewritten each time because it wasn't originally designed to meet the needs of multiple applications.So organizations end up with a hodge-podge of complex systems, all requiring regular maintenance.

Moreover, building the infrastructure and building it right can be time-consuming and costly. And it has to be done first, before the functionality can be implemented. That means that the functionality, the most important piece, gets whatever time is left in the schedule, and usually far from enough. So, organizations invest the most time and money in the area that requires the least unique and specific solution. They waste their money reinventing the wheel.
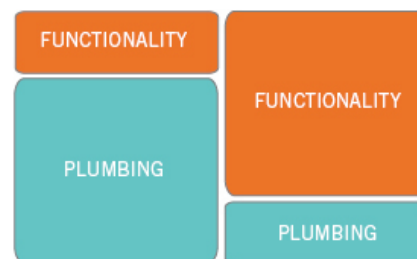
Using Microsoft Dynamics CRM as your platform means that the plumbing is done for you. You don't have to create a data model or a security model. Even much of the user interface is created for you. Moreover, you can use the same infrastructure for all of your LOB applications. By building on this common platform, you can eliminate the majority of the development work required to get an

application off the ground.

Using Microsoft Dynamics CRM as your platform means that... you can use the same infrastructure for all of your LOB applications.

We've found that using Microsoft Dynamics CRM as your platform generally saves 50% to 70% of the development time. Effort you would have spent on the platform can be put towards the harder problem of identifying the right functionality and providing it in the right way. You can spend your time and money on the more important problems of determining exactly what information to track and manage; what processes to put in place; and where, how, and to which users to expose different functionality—the more challenging work that will determine the application's success.

Fig 1. Custom Application Development vs.Microsoft Dynamics CRM as a Platform

Pay-back
Obviously, when you use Microsoft Dynamics CRM as a platform, you pay some upfront costs. You recoup those costs over time, though. Because you've got that common layer for all of your LOB applications, you never have to repeat the process of building a platform. Thus, the cost of your future applications is drastically reduced, paying back the cost of your initial investment in Microsoft CRM. Every time you add an application on top of

Microsoft Dynamics CRM, you've reduced your base cost via the platform savings on the new application.

This savings also comes in terms of time and quality. Custom business applications are often not extensible or scalable over the long term. They are developed to address a specific need but, due to resource constraints, are typically built in such a way that they cannot be quickly or easily repurposed or extended when new needs arise. With a solid, reliable platform, you have a predictable method for extending and scaling applications.

Lastly, you save money because Microsoft takes on much of the work for you. Instead of having to invest in updating and modifying your platform as new needs arise or technology changes, Microsoft takes on that cost. You have a full-time development and test team dedicated to your platform in the form of Microsoft—one you don't have to manage or incur the costs for. Microsoft bears the burden of updating the platform and making sure that it works with all the other Microsoft products, like SQL Server, Exchange, SharePoint, and Office. To take advantage of the newest technologies, all you need to do is upgrade Microsoft Dynamics CRM.

.NET versus CRM
.NET 3.5 now includes a data access layer and data model. The industry definitely understands what developers want and is moving to provide it. At this point, though, .NET only offers the most basic functionality. It doesn't include a security model or a UI framework and still requires a lot of raw development to get anywhere close to the functionality of Microsoft CRM's full platform and application framework.

Microsoft Dynamics® CRM as a Business Application Platform

Reuse, Repurpose, Repackage
One of the unanticipated benefits of having Microsoft Dynamics CRM as your platform is that it facilitates reuse. Code you used in one application can be reused or repurposed to serve the needs of your latest application. You'll soon have a collection of code and tools that you can leverage for new LOB applications.

Not to mention the fact that your business applications all reuse the same data. So the recruitment management system you built for finding and hiring new employees is tied to the new relocation assistance system you've built. And they both refer to the same central database that stores information about employees.

Benefits to the Business
So far, we've talked about the development benefits of using Microsoft Dynamics CRM, in terms of time, cost, and the added agility it provides. What other benefits does a business derive from using Microsoft Dynamics CRM as its development platform?

Centralization
A central location for storing and accessing data.

Standardization
A standard system with a consistent user experience and all the attendant benefits that go along with that, including easier adoption and lower training costs.

Simplification
Simplified development with the expected cost and time savings—and some unexpected ones. In fact, it's more configuration than development, in many cases.

Fig 2. Platform and Application Costs Decrease Over Time and Usage
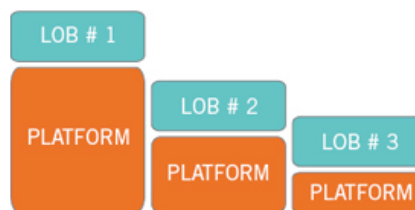
Integration
A system built to be easy to integrate with the organization's existing systems.

Centralization
Using Microsoft Dynamics CRM as the platform for your line of business applications means that they all share a single database. They access information from the same location and in the same way. One of the key business benefits of this centralization is the visibility it provides. All the applications have access to all the information. Obviously, you can choose not to expose all information everywhere—but you get to choose. You can expose whatever information you deem relevant and to whomever needs it, without being concerned about whether your customer service application can talk to your sales force automation tool.

Building all of the applications from the same platform helps prevent the silos of information that develop with disparate systems. Because all of the systems work together, users have one source where they can gain a complete view of all the organization's important data. For example, a sales representative planning to call a customer can easily see the customer's purchasing history, find out if the customer has made any recent service complaints, and verify that the customer's bills are up-to-date. The sales representative can go into her phone call confident, with a full view of any concerns that might need to be addressed and no fear of being blindsided by an outstanding issue.

Standardization
Remember when Microsoft OneNote came out? It was a late-comer to the Office family, and that proved a good thing. A new OneNote user was already familiar with 50% of the functionality of OneNote, simply because it was part of the Office family. You knew how to select and bold items, where to look in the menus for Spelling or to create a bulleted list. And what you didn't know, you knew how to look for. It was easy to click and discover features and you felt comfortable exploring because it was Office, an interface you knew and understood.

Likewise, your applications are built on Microsoft Dynamics CRM, they all inherit its clean, easy-to-navigate, web-like user interface. Users can navigate to and through your latest application in the same way they did with the first one. They don't

Microsoft Dynamics® CRM as a Business Application Platform

have to learn a new user interface, meaning your employees are productive right away. No work loss associated with the new deployment.

Likewise, your training and your support costs are lower. The vast majority of calls to help desk on a new application are new users trying to understand how and where to perform tasks. Microsoft Dynamics CRM's user interface is proven easy to use. One of its greatest strengths is the way it surfaces relevant tasks and information at the right time. Your users will need less training, be up to speed quicker, and need less ongoing support. You may find you don't even need to do formal training.

Simplification
Using Microsoft Dynamics CRM as your platform provides a proven, standard architecture for all of your organization's systems. That means less integration and documentation issues. Instead of three different developers building three different systems to track clients, service calls, and finances—a common scenario within organizations—you'll have one system. As individuals come and go, knowledge about the applications' architecture stays with the organization, providing long-term stability and continuity.

On the IT side, there's only one system to support. One platform, instead of three or four, each with their own quirks and needs. One set of hardware requirements. One license cost per user, no matter how many LOBs are built on the platform. New personnel have only one system to learn and your IT people can support multiple applications, because of their common base, rather than becoming specialists in one application or another. Standardizing reduces your IT costs dramatically.

Integration
Microsoft Dynamics CRM was built specifically to integrate. It's a simple web service: a .NET front end with a SQL back end. It's WS-I compliant and supports ODBC, BizTalk, and so on. We haven't found a system yet that we couldn't integrate with. Because it's .NET, you can always write code to connect and synchronize with external systems. And because it's not a custom platform and is built on common, well-supported servers applications like SQL and .NET, you can take advantage of the many tools put out by Microsoft Corporation and others for these systems.

Benefits to the Users
We've talked a lot about benefits to the business, but what about the end-users in the organization? For the most part, of course, it's the applications you build that provide the benefits to these individuals. But the platform plays its part, as well. The value it adds falls into two categories:

Productivity Gains
Users are more productive because of the consistent and high-quality user interface and the many productivity tools that come with Microsoft Dynamics CRM.

Information Access
End-users have access to all the information they need when they need it and in the way they are most comfortable accessing it.

PRoducTiviTy gainS
An application is only as good as the data that goes into it. When users see their own productivity improved by using the system, they are more likely to enter the data that makes the system valuable to the organization. Microsoft Dynamics CRM comes with a number of productivity tools that

add value for users, including personal workflows, automatic to-do notifications, accessibility features like keyboard access, the ability to build quick, custom reports, and seamless integration with Microsoft Outlook.

The success of an application starts with the end user. Ensuring that the platform provides a user-focused experience that is consistent across all applications is essential to achieving the goals of any system. The user interface can be a big barrier to adoption by users in the organization. It's frightening how many times organizations have invested years in developing and deploying large and costly systems only to have the implementation fail because users found the system too difficult to learn and use. Far better to have employees choose to use an application rather than have to mandate its use.

Fig 3. Using Microsoft Dynamics CRM Lets You Increase the Size of the Middle Ground

When is the last time you watched through a one-way mirror as users work with your internal application? Microsoft does it all the time in their usability labs. They have a whole team dedicated to making Microsoft Dynamics CRM's user interface the best it can be. So the users at your company can benefit.

Another reason for failure is that the system ultimately deployed doesn't solve the employee's real problem, so they don't use it. Having Microsoft Dynamics CRM as your platform

Microsoft Dynamics® CRM as a Business Application Platform

allows developers to focus on solving the employees' problems and solving them well, so that they choose to use your application and thereby enter the information the organization requires to meet its needs.

Information Access
People need access to data in an increasing number of places and situations. For example, a business traveler might need access to an application on a mobile phone, over the Internet, or on a laptop while flying in an airplane. Microsoft Dynamics CRM allows users to access their data anytime, anywhere, and on the device they choose.

Offline access is also important. Contrary to popular belief, there are times when you don't have Internet access. That field service worker who is literally inspecting a field doesn't have access. Likewise for those housed in the bunker-like buildings of some office parks. And that traveling employee who just hit a dead zone still wants to do her work. Microsoft Dynamics CRM provides an excellent offline story.

Moreover, Microsoft Dynamics CRM lets users work in Microsoft Office applications, such as SharePoint and the omnipresent Microsoft Outlook. The platform is smartly integrated with Outlook, making it easy for users to schedule activities, and send and receive e-mails in Outlook, if that is where they are most comfortable.

Using Microsoft Dynamics CRM as your development platform means that all your LOBs have the same information-access benefit, increasing the likelihood they will be used and used consistently.

Benefits to the Developer
And what about the people actually developing these business applications? How does the Microsoft Dynamics CRM platform help developers? We've already talked about the fact that it allows you to focus on the business problems and processes, and how to solve them through the application's user interface. In our mind, that means you get to work on the fun stuff, instead of the drudge stuff. What else does the platform provide developers?

Latest Technology — You get to work with the latest technology. Microsoft Dynamics CRM allows you to leverage technologies like .NET 3.0, SQL 2007, Visual Studio 2008, and Web 2.0 technologies such as AJAX, Windows Workflow Foundation, and Service Oriented Architecture. Moreover, because Microsoft maintains the platform, as new technologies arrive or existing ones are updated, Microsoft Dynamics CRM will work with them.

Exposed Layers — Microsoft Dynamics CRM's database schema is exposed and easy to understand. The user interface is elegant and obvious rather than convoluted. Microsoft Dynamics CRM isn't a black-box. At the same time, you don't have to be a database analyst to extend the data model. In fact, you don't even have to work directly with SQL, if you don't want to.

Loads of extras — The system provides you with a UI framework and editors for designing forms and lists. You get offline access, the ability to export to Excel, reporting, and role-based security for all your extensions. All of this is free with the platform. So, unlike raw .NET development, you won't need to find or build another data access layer and your own form layer.

A Single SDK — Since you use one platform, you only need to learn and use one SDK. Microsoft Dynamics CRM's SDK provides lots of code samples and tools to jumpstart your development. It is one of the best SDKs out there and is supported by a very active newsgroup community. And there's a great MVP and ISV ecosystem to leverage along with the SDK.

A Responsive Product Team — The Microsoft Dynamics CRM team is very open to feedback from customers and developers. They rely on newsgroup and partner feedback to determine what features to focus on. Customer feedback is often incorporated into interim updates. For example, the ability in 4.0 to display columns from other objects in CRM views and the auto-completion feature in lookups both came from developer feedback. As partners ourselves, that's important to us.

The Skept ical Developer
Ross Lotharius has been a developer at Avanade since the late '90s. When client projects came in and Aaron recommended using Microsoft Dynamics CRM, Ross was the one who always spoke out against it. As he stated, "I had a pure custom development mentality. I always thought, 'I'll just do it myself.'" In meetings, Ross would argue the need to design the data model or the security model from the ground up. Microsoft Dynamics CRM, he thought, is just an application. Building on top of it will require hacking the application to give a client what they want. It wasn't an appealing thought. In short, Ross had yet to see the light.

Then along came Embrace Pet Insurance. The company literally provides medical insurance for pets. They were looking to build an application to support their business,

Microsoft Dynamics® CRM as a Business Application Platform

but the cost they were quoted was vastly more than this start-up could afford. They began looking at systems that might provide some of the functionality they needed and Microsoft Dynamics CRM popped up. So, they approached Invoke with the project.

Once again, Ross looked at using Microsoft Dynamics CRM 3.0 as the platform. But this time he approached it with a more open mind and began to see the advantages. CRM would provide all the core functionality Embrace needed, like user management, roles and security, and a solid data model. And the customizable entities made it easy to convert opportunities to insurance quotes, contacts to pet owners, and accounts to veterinary hospitals. Soon, and with a minimum of effort, Embrace's customer service representatives had their internal system and were using it to track quotes, policies and process claims.

Skipping all the painstaking platform work allowed Ross and his team to focus on the really juicy part of the project—Embrace's web portal. The portal was a key part of Embrace's business strategy, providing customers with the ability to get quotes and purchase policies online. Ross's team was able to build the web portal on top of Microsoft Dynamics CRM and spend their time getting it right by providing a great Web 2.0 user experience for Embrace customers. The web portal hooks into CRM, reading and updating data behind the scenes. Using the power of Microsoft Dynamics CRM's SDK, web service, and APIs, they were able to finish the project in about six months time and at a fraction of the cost, including the licensing. Ross was converted.

... they were able to finish the project in about six month's time and at a fraction of the cost, including licensing. Ross was converted.

Currently, Ross is a lead architect and director of Software Development. As a director, he's found using Microsoft Dynamics CRM has many other benefits. For one thing, it prevents burn-out on his team. Custom development takes a lot of time and while designing a database and a platform might be fun at the beginning, by the end it's just drudgery. You wind up spending most of your budget on the foundation and the least amount of time coding the most important parts to your client: the functionality. Microsoft Dynamics CRM provides for "Very fast turn-around on our projects. We don't have many projects going on for years," says Ross. "The burn-out isn't as bad because you aren't working on the same projects for years. You get lots of variety." Another benefit is that it cuts down on the tedious platform design work and the many meetings, documents, and reviews that have to take place as part of the platform design process. Microsoft gets to do all of that, while his developers are "getting their hands dirty by coding and creating features."

And now, he extols the benefits of using Microsoft Dynamics CRM as a platform. As an architect for projects, he can't remember saying no to a customer yet. Microsoft Dynamics CRM has all of the infrastructure and the hooks he needs, right out of the box. Its event model and broad functionality mean that hooking in is very, very easy. So when a customer asks, "Can you...?" So far Ross's answer has always been, "Yes."

Data Model
You'll want to understand classic data modeling, but database management and database construction aren't necessary. There's no need to ever write a single SQL statement to build out the entire data model because it's all done using tools within Microsoft Dynamics CRM.

Microsoft Dynamics CRM provides a data structure ready for your use. Contacts, accounts, addresses—they're all already defined. You don't have to come up with a taxonomy or define a structure. And all of your applications will use the same definitions, making them consistent with one another—so you don't have to write code to translate one application's format for addresses to match another application's format.

Business Entities and Business Logic
The extensible business logic layer in Microsoft Dynamics CRM provides for unmatched interoperability and customization options, and relies on technology used by millions of developers. You can inject custom logic during pre- and post-events around platform operations and you can write your platform extensions in any .NET language. That means hooking into Microsoft Dynamics CRM is very easy, so you can extend and integrate to your heart's content.

Microsoft Dynamics® CRM as a Business Application Platform

Security

The security model provides a highly efficient means for establishing and changing security privileges to ensure that users always have the appropriate access.

• The security model works in much the same way that files do in Microsoft Windows, following an Active Directory model.

• Role-based security defines privileges based on roles within an organization rather than on individual users. Users can have one or many roles associated with them, with the highestlevel role taking precedence.

• Privileges across roles are additive, so users can perform many roles within an organization and the system will automatically recognize the correct security privileges.

• Users can share individual records with other users (temporary sharing) and delegate privileges to those users based on business needs that normal roles would have precluded.

• Filtered Views can be used to write queries or reports directly against the database, while adhering to the security rules of the user executing the query.

Workflow

By leveraging the Microsoft Windows Workflow Foundation, Microsoft Dynamics CRM provides efficient methods for organizations to increase productivity and automate tasks between different systems.

• You can construct workflows that can be executed across other applications because Microsoft Dynamics CRM leverages the same workflow engine as other Microsoft products, such as Microsoft Office SharePoint Server.

• Workflows can be executed manually, when records get created or deleted, when records'statuses change, or even when a particular data value in a record changes.

• You can use workflows to automatically create activities, update records, send email on a user's behalf, and extend workflows with your own custom workflow actions.

• Custom workflow actions are .NET assemblies that can take data from Microsoft Dynamics CRM, pass it to another system, get a result, and use that result as a decision factor throughout the rest of the workflow.

• You can define workflow rules within a simple web user interface that is similar to the Microsoft Outlook rules interface. For more complex workflows, you can open a rule inside Visual Studio to leverage the full capabilities of the workflow engine.

• Workflows can be defined at the organizationallevel to apply to everyone, or by individuals as personal workflows. For example, when new leads come in, one user can set a workflow to automatically send an email using a template that user created, while another user can have a workflow automatically schedule a phone call activity for the user to call the new lead.

Services

Microsoft Dynamics CRM services break down the barriers between systems and facilitate a broad range of customizations:

• It's completely metadata-driven, and data model modifications are automatically represented in the dynamically-generated web services.

• The entire platform is exposed through web services, which is the exact same collection of web services that Microsoft developers used in building Microsoft Dynamics CRM.

• Web services allow interaction with entities that ship in the box as well as with any custom entities that are created during configuration of the system.

• Microsoft Dynamics CRM for Outlook with Offline Access has a local copy of the web services so you can write against both the server and the offline clients.

• Programmability against these services is the same whether the user is connecting to the actual server or to the local data store on an offline client.

UI/Users

The user interface components enable you to give end users a proven, consistent and dynamic experience that maximizes the value of the user interaction:

• Base-level data model changes bubble up all the way through the UI layer so no coding is needed to enable basic functionality such as forms, searching, data views, and so on.

• You can bind to events on forms to build dynamic user experiences when users are filling in data. Microsoft Dynamics CRM exposes onload and on-before-save events at the form level and on-change events at the field level on the forms.

Microsoft Dynamics® CRM as a Business Application Platform

Security

The security model provides a highly efficient means for establishing and changing security privileges to ensure that users always have the appropriate access.

• Custom buttons/actions can be added to forms and data grids, and custom navigation panes and I Frames can be added into the application and forms to allow composite applications. Of course, developers can extend the basic functionality of the product with custom dialogs and code.

Let's Go!

So, now you know all the reasons why you should choose Microsoft Dynamics CRM as your development platform. Want to know more? There are several ways you can learn more about Microsoft Dynamics CRM. Our solution architects at Ascentium would be happy to talk with you about the benefits of Microsoft Dynamics CRM and how it could be used as a development platform in your organization. We deeply believe in Microsoft Dynamics CRM as an agent for change in how organizations develop software and want to help organizations understand the benefits, whether or not it results in work for us.

Microsoft is another good source of information. Specifically, see: http://www.microsoft.com/dynamics/crm/product/default.mspx the Microsoft starting point for learning more about CRM. We also encourage you to ask other users about their experience developing on Microsoft Dynamics CRM. The CRM newsgroups are a great resource. The site http://www.microsoft.com/dynamics/crm/community/default.mspx provides links to a number of forums you can explore.
Finally, to obtain more technical information, you can download the SDK at:
http://www.microsoft.com/downloads/details.aspx?FamilyID=82E632A7-FAF9-41E0-8EC1-A2662AAE9DFB.

Once you've explored the possibilities and the pros and cons, we're confident that, like us, you'll conclude Microsoft Dynamics CRM is the best choice for line of business application development.