Microsoft® Visual Studio®

Microsoft® Visual Studio® 2010
Reviewer's Guide

Contents

This reviewer's guide is designed to give you an overview of the features available in Microsoft® Visual Studio® 2010. It's intended to give you an understanding of how professional developers can be more creative and more productive while building high-quality applications.

The guide begins with an example scenario to showcase the new product features in action, complete with screenshots of the solutions being built. It then covers each of the new features in greater detail. Throughout the document, we've included links to Web sites that offer deeper information about the technology.

This guide will be updated as information becomes available, and is subject to change. For the latest information about Visual Studio 2010, please visit **http://www.microsoft.com/visualstudio/.**

You're going to love what we've done with Visual Studio® 2010.

You've got big ideas. You know the algorithms. The architecture's up on the whiteboard, and you can imagine the user interface. And now, with Visual Studio 2010, you've got the tools to turn your ideas into solutions.

The new Microsoft Visual Studio 2010 simplifies the entire development process, from design to deployment. New prototyping, modeling, and design tools help unleash your creativity and bring your vision to life. Coding is faster than ever, since you can leverage existing code and skills to write applications targeting different platforms, including SharePoint® and Windows® Azure™ applications. What's more, developers and testers will work better together and save time with integrated testing and debugging tools. You'll find and fix bugs quickly to make sure your solution is at the highest possible quality.

Take a look at what's been done, and see for yourself how this edition can boost your team's creativity to deliver quality applications through a more simplified process.

Microsoft® Visual Studio®

VISUAL STUDIO 2010 > **In action.**

This release of Visual Studio has plenty of compelling new features — and some key updates that will make every developer using it a whole lot happier with their favorite IDE.

But seeing it in action — and actually getting your hands on the product and using it — is what'll convince you that this edition's new features and tools are truly breakthroughs. Not only will they change the way teams develop applications. . . they'll change how businesses do business.

Until you have a chance to see and try Visual Studio 2010 for yourself, the following scenario will show you how Fabrikam Consulting helped open a new channel for **Coho Vineyard & Winery™.**

Microsoft® Visual Studio®

Meet Coho.

Coho Vineyard and Winery™ is a small-scale, local winery that has a lot of good stuff going on. They're passionate about growing the perfect grapes. They've got their harvesting and fermentation down. They innovate new blends every year. And their local marketing efforts and customer service are outstanding.

Connoisseurs, restaurants, and local foodies love Coho wines. . . but they're the only ones. No one else can get their hands on this tasty local flavor, since it's sold through only three channels: directly at the vineyard, through a Web site that serves the local market, and at a few high-end local restaurants.

And then. . .

“Leading critics give boutique winemaker’s Coho Cabernet a 92!”

Thanks to some remarkable nationwide press, demand goes through the roof. Coho’s site simply can’t handle sales of more than five bottles per user. (Not to mention handling the traffic this single review has generated.)

It’s clear that there’s a new opportunity to expand their distribution channel to include specialty shops, large local chains, and national retailers.

Demand? Check. Supply? No problem. Technology is the only thing holding Coho back during this opportunity for explosive business growth. **They simply don’t have the systems in place to manage the orders and handle the distribution.**

Coho is no stranger to technology, mind you. For years, they've had a sufficient system in place and are ahead of their competition in the local wine industry.

They have:

+ **An eager sales manager** who has personal relationships with customers.
+ **A point-of-sale application** they use at the vineyard for in-person purchases.
+ **A simple e-commerce solution** on their site, which works fine for buying a few bottles.

To take their business to the next level, Coho needs a system in place to capture the holiday market — the hottest season for wine retailers. That means the solution has to be up and running by September 1. That's just six (short!) months away.

A few internal meetings quickly prove that they're potentially in over their heads. Within six months, they'll need:

+ **A better Web site** to keep their current customers and support the new wholesale market.
+ **An ordering system** to process sales for industry buyers — restaurants and distributors — who want 10+ bottles (or cases) at a time. Ideally, it should offer dynamic volume discounts and accept foreign currency.

The folks at Coho just aren't sure how to get the site and ordering system up and running in such a short timeframe. Is it even reasonable? They have a great idea and can imagine the future. **But clearly, Coho is going to need some outside help.** The search is on!

Coho's current home page. Definitely room for improvement!

The product pages and ordering system need to handle higher volumes.

Enter Fabrikam.

After assessing a number of options, Coho hires a firm that has retail and wine industry experience: Fabrikam Consulting.

Fabrikam is a large local firm that's been in business for six years. The core team dedicated to Coho is made up of 12 technology pros.

For the Coho project, they've got a short timeline and a tight budget. It's going to be more important than ever to streamline the application lifecycle and make sure everyone is collaborating on the solution.

They've been working with Visual Studio 2005 ever since it came out, and Fabrikam might be ready to upgrade. But before committing to new licenses across the whole company, the management team decides to test run it for the team assigned to the Coho project.

This plan gets a mixed reaction. The QA team is anxious to get their hands on the new testing tools they've read about. The project manager is on board once she realizes how much easier tracking progress will be in Visual Studio Team Foundation Server 2010. Developers primarily care that they'll be able to repurpose code they've crafted for past projects. And the refreshed IDE is going to be HOT to work in.

Determining what Coho really needs.

During the discovery and research phase, Fabrikam's project manager dives deeper into Coho's business needs to create a strategic recommendation.

The project manager and the client at Coho collaborate to finalize a list of business requirements:

+ High-volume order capture and processing
+ Shipping and tracking systems
+ Variable credit terms across multiple currencies
+ High-end brand experience
+ Real-time reporting for order status, credit limits, shipping, payables, and receivables
+ Integration with inventory system
+ Cross-selling tools
+ Marketing support for events
+ Scalability for future growth

Fabrikam's project manager captured all this info in Microsoft Excel®, and heads back to the office to kick off the project with the team.

The Coho project team at Fabrikam

+ 8 Developers
+ 1 Designer
+ 2 QA Testers
+ 1 Project Manager

Fallout from discovery process — the solution takes shape.

The top-level business requirements make it pretty clear to the team that they're looking at creating a fully revamped Web site that has a robust e-commerce and inventory system.

The core components of the site include:

- **A video player:** Adds high-quality interactivity and can showcase the vineyard's stunning vistas and grounds.
- **New e-commerce solution:** To support high-volume orders, mix-and-match cases, etc.
- **Customer and distributor profiles:** Allow return customers to get a higher level of support than what's currently offered.
- **Variable pricing models attached to the profiles:** Each distributor will have unique pricing arrangements as determined by its contract.
- **An event-planning tool:** An online form would help manage the deluge of phone calls they get from event planners. A simple calendar and interest form will let customers easily check availability.

Digging into design.

Fabrikam has identified as many of the needs as possible at the beginning of the project. Now it's time to start drafting the functional requirements for the solution before the team gets to work on the project parts.

The project manager and QA lead collaborate on capturing the requirements. He's familiar with designing use cases using Unified Modeling Language (UML), so he spends some early cycles creating the use-case diagrams to ensure everything is captured early on.

This step is crucial. They've got the business goals defined, and the personas are pretty clear-cut: wholesale distributors, individual customers, event planners, job seekers, and members of the press. From there, they can get to work on the user stories, how each audience will interact with the site, and think through every possible use scenario.

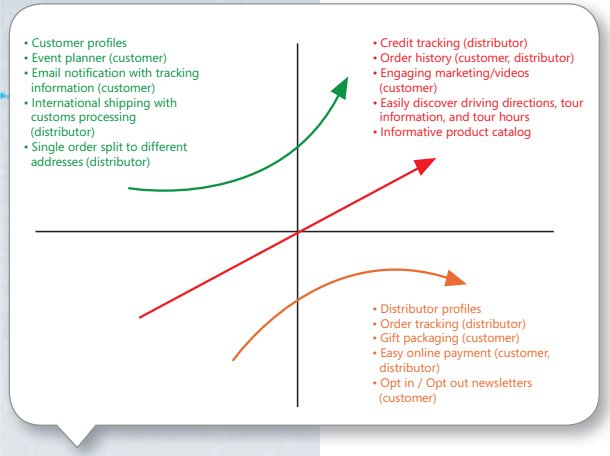
Are there tax or shipping implications?

How are overseas requirements different?

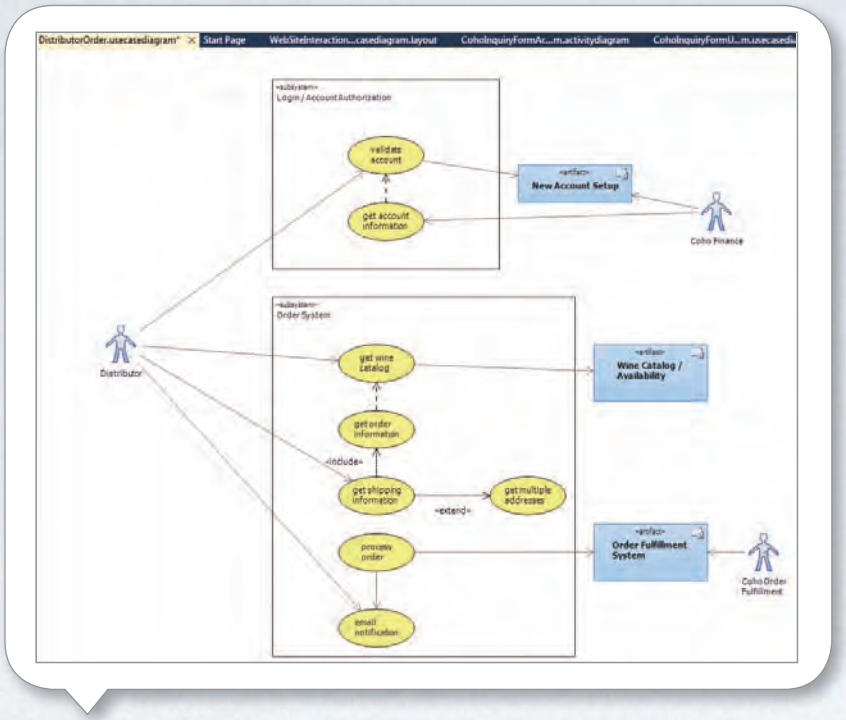
What's the best way to verify ages to comply with federal laws?

And on and on...

The team captures all the potential user stories.



A Kano Analysis shows what features matter most.

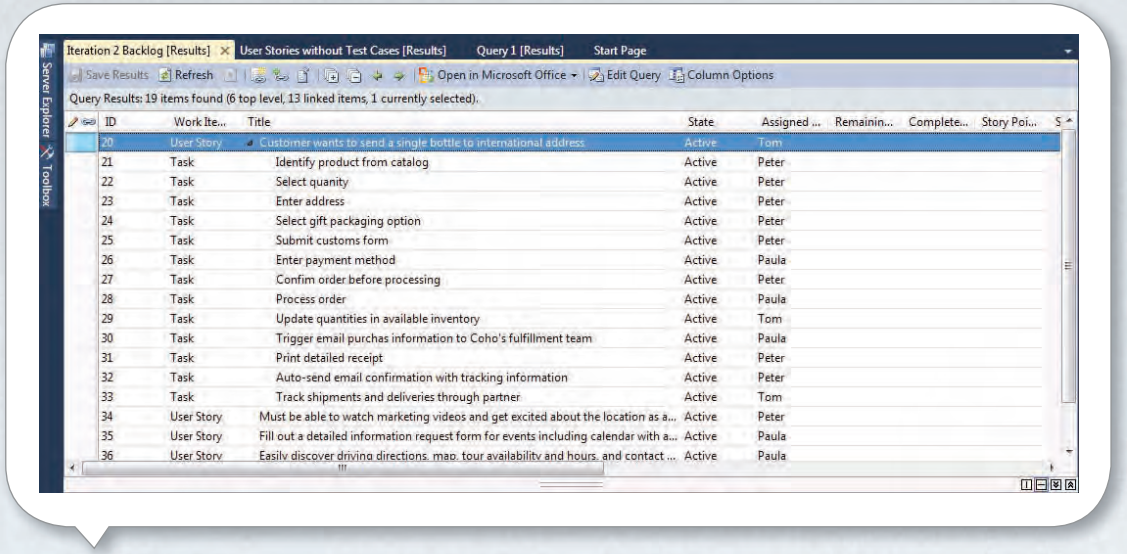


A use case diagram illustrates the user's interaction with the system.

Once they're done brainstorming, they come up with quick Kano analysis for the features to help determine what the priorities are — must-haves, satisfiers, exciters, and indifferent. Kano charts only take a few minutes to create, and Fabrikam includes this step as one of their best practices to keep the team focused on the end-users' needs in addition to the needs of the client.

Then Fabrikam dives deeper into the requirements for each of the scenarios — including quality-of-service requirements — captures it in Excel, **then syncs it all with Visual Studio Team Foundation Server 2010.**

The Fabrikam team now has access to the requirements behind upcoming task lists, and their clients at Coho will have transparent visibility to the requirements too. That means fewer surprises — and fewer last-minute changes. Team Foundation Server dramatically reduces the complexity of corralling all business requirements, especially when the client keeps remembering new needs throughout the early stages of planning.



ID	Work Ite...	Title	State	Assigned ...	Remainin...	Complete...	Story Poi...
20	User Story	Customer wants to send a single bottle to international address	Active	Tom			
21	Task	Identify product from catalog	Active	Peter			
22	Task	Select quanity	Active	Peter			
23	Task	Enter address	Active	Peter			
24	Task	Select gift packaging option	Active	Peter			
25	Task	Submit customs form	Active	Peter			
26	Task	Enter payment method	Active	Paula			
27	Task	Confim order before processing	Active	Peter			
28	Task	Process order	Active	Paula			
29	Task	Update quantities in available inventory	Active	Tom			
30	Task	Trigger email purchas information to Coho's fulfillment team	Active	Paula			
31	Task	Print detailed receipt	Active	Peter			
32	Task	Auto-send email confirmation with tracking information	Active	Peter			
33	Task	Track shipments and deliveries through partner	Active	Tom			
34	User Story	Must be able to watch marketing videos and get excited about the location as a...	Active	Peter			
35	User Story	Fill out a detailed information request form for events including calendar with a...	Active	Paula			
36	User Story	Easily discover driving directions, map, tour availability and hours, and contact ...	Active	Paula			

Top-level requirements are broken down into smaller work items, all in Team Foundation Server.

Before defining the methodology or assigning resources to tasks, the team looks at past projects to see where they can lift code — that way, they won't have to build everything from scratch. (And they've done a lot of other projects for vineyards, so they're bound to find efficiencies.) Projects built in earlier editions of Visual Studio are a snap to pick up and refine in Visual Studio 2010.

Then they'll be ready to finalize the first iteration of the to-do list, building out tasks and test cases.

New

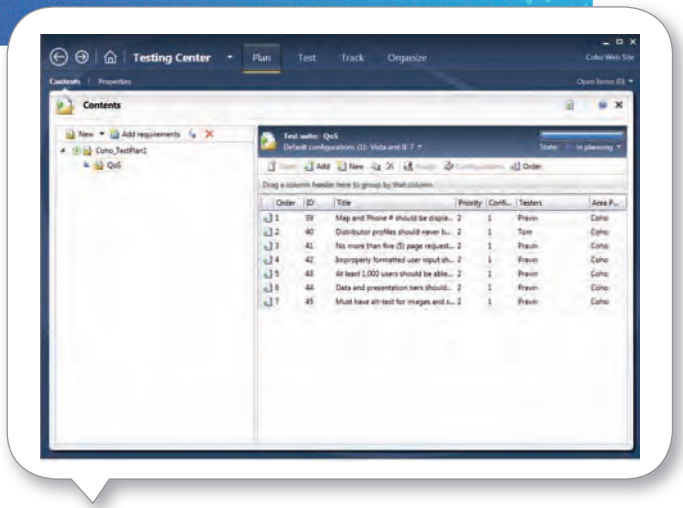
Managing the discovery stage with Team Foundation Server.

1. **Centralized requirements:** Don't worry about managing multiple versions of the requirements doc and risking churn on outdated requirements. Every stakeholder has instant access to a single, central, current requirement list, and team members can pull individual spreadsheets and reports.

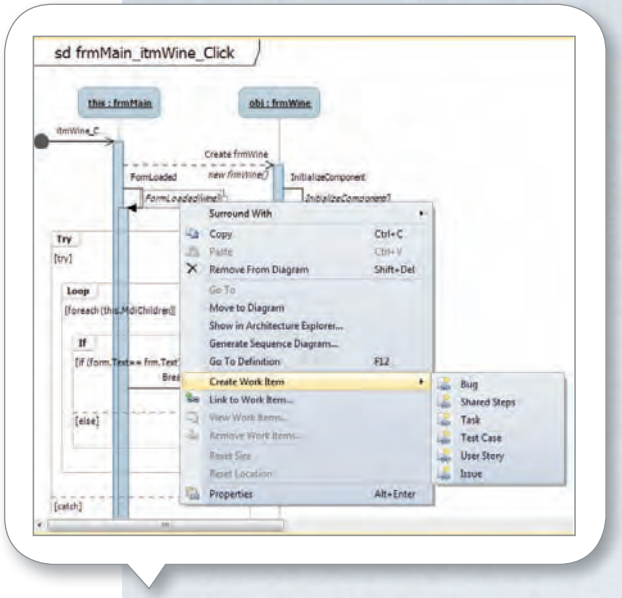
2. **Real-time status:** Each team member can see — in real time — current status, any changes, progress, and quality metrics for all requirements.

3. **Change management:** Centralized requirements reveal which features have been overlooked and which are still in development, so there are fewer last-minute crises. It's as simple as submitting a feature request, or changing the priority status of a task or feature. The client is free to monitor progress, log bugs, and make change requests, too. . . and know that his input is captured and tracked.

Coming up with Test Plans — early and often.



The team creates a test suite with the steps required to execute the test.



The lead developer creates task work items directly from the sequence diagram.

While the developers are detailing out the functional specs of each of the features, the QA lead is tackling the initial Quality of Service Requirements and test cases.

Visual Studio® Lab Management 2010, part of Visual Studio Test Professional 2010, comes with three modes: plan mode, test mode, and track mode. In plan mode, Fabrikam creates the test suite, with the test cases plugged in below. While the tests are added to the tool, they're automatically pushed to Team Foundation Server.

Structuring the solutions.

The Fabrikam team doesn't have a designated architect, but it'll be easy for the lead developer to take on the responsibility on this project. He'll take two approaches to get where he needs to be.

Fabrikam already has Coho's current site to build upon, so they won't be starting from scratch. The lead developer quickly generates a dependency diagram of the current solution to get an overview of how it works. He builds sequence diagrams for the new features. With these diagrams in place, the architect creates work items directly from diagram parts, and can be confident that his design will stay current throughout the development process.

To make sure the functionality is correct, he'll use the class and component UML types that are integrated in Visual Studio 2010. All this work up front ensures from the beginning of the project that the highest level of quality is maintained throughout the project and software lifecycle.

New

Diagramming in Visual Studio 2010.

Visual Studio 2010 provides full support for different types of UML diagrams, including use case, to help you model user requirements and describe the functionality of your system. This planning stage makes sure everyone on the team agrees on functionality and lets the solution set before starting on the code.

Visual Studio 2010 now supports five types of UML diagrams:

+ Activity diagrams
+ Use-case diagrams
+ Sequence diagrams
+ Class diagrams
+ Component diagrams

Making it pretty.

While the functional spec is being refined, the team has enough information to get going on comps. The client is itching to see something — it's been a couple of weeks, and the planning process is getting tedious. What's it going to look like?

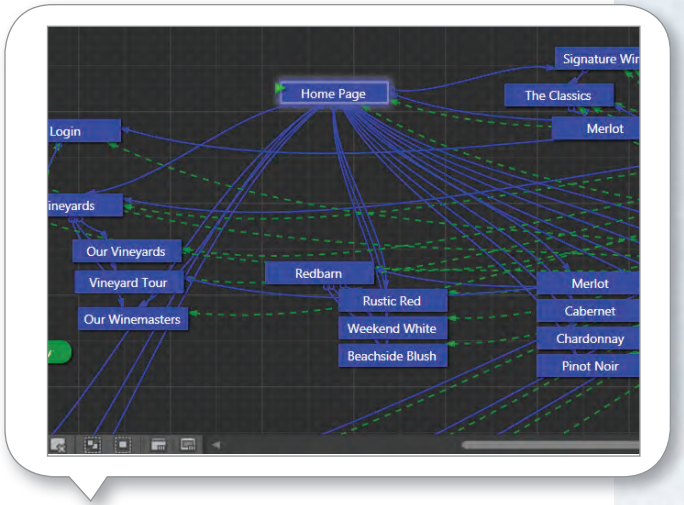
It's time for Fabrikam to step into the shoes of the customer and start designing the interface.

The site is going to need a lot of user experience and graphic design work to polish the look and incorporate the complete feature set.

Fabrikam's designer powers up Microsoft Expression Blend™ and uses SketchFlow to bring his doodles to life. It's an incredibly cool, efficient way to quickly map out prototypes for the client.

For the first round, Coho will be looking at a rough, usable prototype — not just JPEG comps — with actionable navigation, controllers, and forms. It's true. . . and the designer will get it done faster than a slick-looking, non-working comp.

SketchFlow creates a working prototype...



...and illustrates the navigational flow.

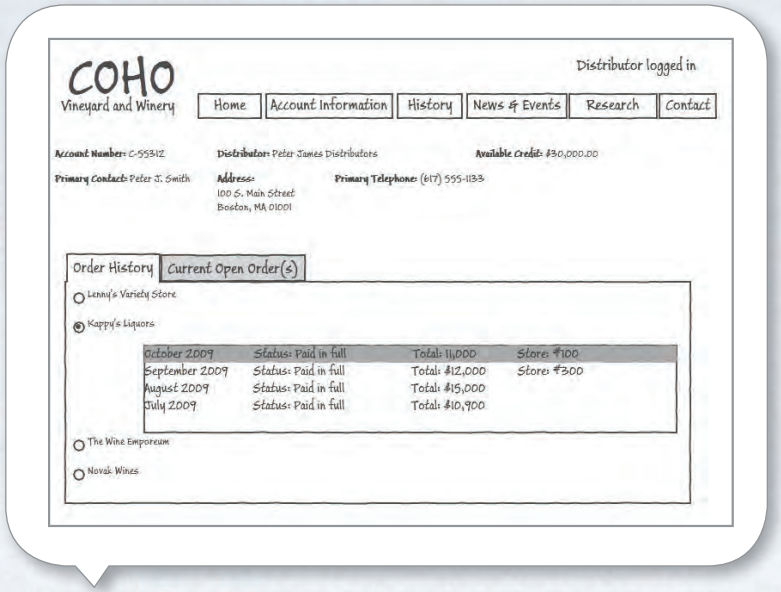
Designers and developers working together.

It's an age-old battle. Designers want projects to look great; developers want it to be simple and work right. And everybody wants more time.

But the tug-of-war might finally be over. Visual Studio 2010 makes a huge step in bringing form and function together. SketchFlow functionality gives designers a shot at ensuring their vision is maintained, by creating strong prototypes from the get-go and importing their work directly into the IDE — independent of the languages used to develop it. **Less back and forth on the little stuff, more time to come up with killer ideas.**

SketchFlow is a part of Microsoft Expression 3. Visual Studio Ultimate and Premium users get Expression as part of their included MSDN subscription. It's also available as a stand-alone purchase.

Stakeholders provide feedback to the team, right on the running prototype.



The prototype functions just like the end solution will, complete with live data.

It's wireframes with action — complete with animation. Pretty hot for a week's worth of work.

His SketchFlow prototypes are dropped into version control in Team Foundation Server to get feedback from the team. Since the feedback is in context with the design, it's far easier for the designer to process the input, instead of tracking e-mail or keeping track of printouts with markups.

After a few internal rounds with the team about the look and feasibility of his designs, he finalizes three concepts to present to the client.

Coho is thrilled. They love the presentation format. . . and don't have to stress about how individual buttons will look or how big the logo is. They decide to move forward with one of the ideas, and the designer can really focus on the look and feel for just one site, rather than blowing out all the concepts.

Designing with agility.

Now that the preliminary work of use cases, functional and quality-of-service requirements, and prototypes has been approved by Coho, the team is ready to decide on a methodology, plan the development, and begin tackling tasks.

To make the best use of the team's time — and their individual strengths — Fabrikam uses the new tools in Visual Studio 2010 to plan three agile sprints. (A linear, waterfall process is supported just as well by Visual Studio 2010, but Fabrikam loves to SCRUM.)

The team will do three 15-day sprints with daily SCRUMs and drop builds for the customer at each milestone. They'll start by building out the infrastructure, then layer on the business logic, and finally pull the whole thing together before deployment.

Sprint 1:

+ Design graphics
+ Build the ordering/e-commerce system
+ Create the profiling system
+ Develop the catalog

Sprint 2:

+ Create distributor approval workflow
+ Create customer and distributor profiles
+ Add Silverlight Media Controls for streaming video
+ Build the event-planning system

Sprint 3

+ Flow in live copy
+ Final bug resolution
+ Develop e-mail confirmations, error messages
+ Final fit and finish feedback from Coho

Agile development will be much easier to manage in Visual Studio 2010, thanks to the new Agile Planning Workbook. These workbooks help the project manager load-balance her team and track progress against the milestones.

Planning and managing agile sprints.

Since Fabrikam is a small team, they prefer to use an agile methodology to stay flexible during the development and design process. This allows them to:

+ Minimize documentation
+ Collaborate more freely
+ Respond to last-minute requests
+ Focus on functionality of unique solution sets
+ Support a flat organization

And it's simpler to track and manage, with new tools that support the agile methodology. Visual Studio 2010 includes a new set of work-item types, link types, dashboards, reports and documents that align better with how agile teams actually work.

Coho Solution 10% complete

Assigning talent to tasks.

Fabrikam's project manager has a detailed list of user stories in Team Foundation Server. She can now assign resources based on their skill sets and experience with Visual Studio 2010. Once team members are attached to tasks, she creates task hierarchies and can identify dependencies and maintain task relationships in Team Foundation Server. The connection between top-level user stories and corresponding developer tasks will help Fabrikam trace requirements and deliver timely updates to Coho.

This used to be an up-until-three-in-the-morning project for her. Not any more.

First, she pulls up the new agile planning worksheet in Excel. Once she enters some basic information about start and end dates for each iteration, she can stack rank the task list.

And here's the best part — she can use historical data in Team Foundation Server to make sure each sprint is realistic based on how long similar projects have taken in the past.

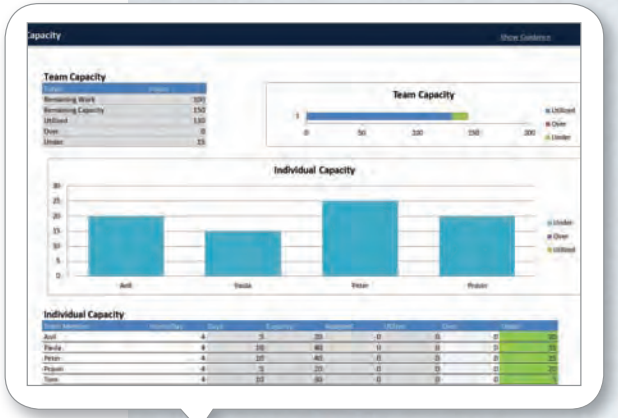
This data gives her a quick capacity graph, which shows that she's overloaded in sprint two, but still has room in the first one. **So she just shifts the event-planning tools up in the timeline to balance out the work.**

Using historical data in capacity planning also gives her a chance now to **add a buffer for unplanned work.** Adding this buffer makes sure the team can take on last-minute requests without cutting any of the features or services they had counted on developing. She also takes the time to define relationships and dependencies between the tasks.

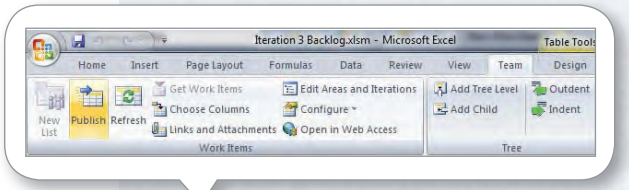
This does more than just satisfy her type-A personality. When hierarchies are defined in Team Foundation Server, they'll be maintained throughout the iterations of the project. This will make it MUCH easier to track when the clients at Coho add functionality later on (which of course they will). The PM, developers, and testers will be able to see how the change affects the rest of the system. When everything's looking good, she clicks a single button to publish it for the team.

Bonus? The project manager can **do it in half the time** and still make it out to happy hour.

Load balancing the tasks is easy to do in Excel – both between team members and across sprints.



The project manager adds a buffer to make sure they can handle scope creep.



Publishing back to Team Foundation Server only takes one click.

Pulling back the curtain for the client.

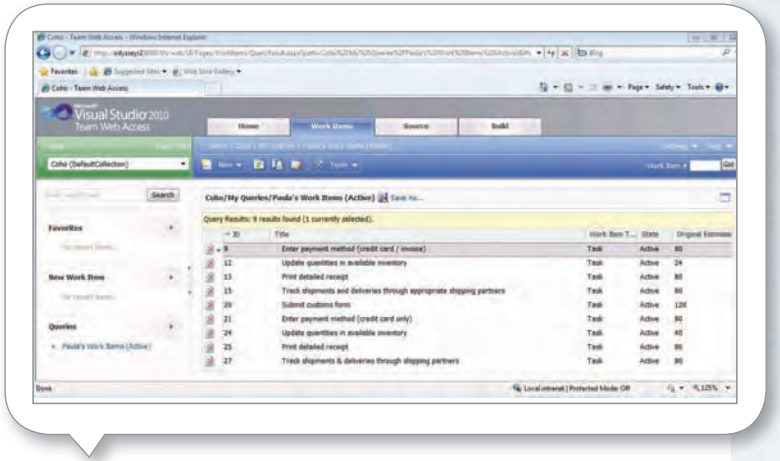
Fabrikam knows that a successful project — especially one on a tight budget and timeline — requires regular and active involvement from the client. The good news is they have a great client in Coho — technically savvy, smart, and fun to collaborate with.

The Coho product manager requires weekly status reports that provide details on bug trending. Easy. *He also wants regular access to builds.* Eeep. This could be a little scary. Can't say no to the client, but how much transparency is too much?

After just a couple of weeks using Visual Studio 2010, the Fabrikam PM has her dashboards and reports customized to give her top-level and detailed views into the status of each branch of the project. She can quickly identify problems and trends, thanks to tools that help her track which changesets are in a given branch. Since Fabrikam is never in the dark about what functionality is in which build, there's nothing to worry about.

They give the Coho client direct access to a Web-based portal in addition to weekly status reports.

Without installing Visual Studio 2010 on his machine, the client will be able to use Visual Studio Team Web Access to interact with the work items. He can also create and update tasks, modify priorities and dependencies, and submit bugs directly as he finds them.



The client can see project progress without having Visual Studio 2010 installed.

Don't feel like sharing?

Every client and every team has different needs. With Visual Studio Web Access, you decide how much information you want your clients to have access to, and set rules for every user who accesses the project reports. From complete transparency to no access, it's up to you.

Sprint 1

Once the project plan is in place, it's time to start the first sprint.

The Fabrikam teams get to work on their individual bits. Essentially, they'll tackle five items:

1. The graphics and user experience
2. The e-commerce solution
3. The profiling system
4. The event-planning system
5. The catalog

Some parts of this sprint are old hat to the Web team. They'll pull a catalog system from another project and repurpose it for Coho.

But past projects haven't given the team the experience they need to deliver a few of the functional requirements. They'll have to come up with fresh solutions for these needs.

The developers want to pick up Coho's existing catalog and shopping cart built using .NET Framework 2.0 in Visual Studio 2005. They have all the assets they can reuse.

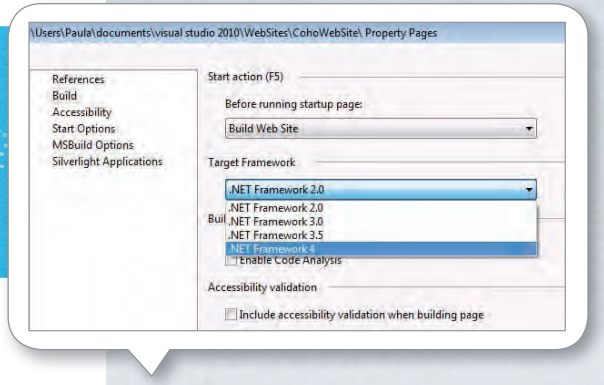
They open the solution file in Visual Studio 2010 and step through the upgrade wizard. In a few clicks, the solution file is converted and open in Visual Studio 2010.

But they want to take advantage of the new features like ASP.NET Dynamic Data, Language-Integrated Query (LINQ) databinding controls, and ASP.NET MVC in the latest version of ASP.NET. All of these features can be used with the current site by simply changing the target framework version and upgrading the Web site to ASP.NET 4. **Using the multi-targeting dialogue, this is a fast and easy step to make sure the old system integrates with their new code.**

The original developer who worked on this solution is long gone. The guy who's working on the new shopping cart enhances the code HIS way — using test-first development.

So first, he writes the unit test for the parts he'll need to enhance the cart experience. In order to keep the pace up, he'll bang out all the code assuming new classes and methods are already implemented. That lets him focus on the logic without getting bogged down with what's required to make his code compile.

Then it's as easy as a right click, and selecting Generate From Usage. Visual Studio 2010 drops in the skeleton code that successfully addresses the logic in the unit test. He then fills in the code required to make his unit test pass. Rinse, repeat.



Old projects can be leveraged just by changing the target platform to unlock the capacities of ASP.NET 4.

New

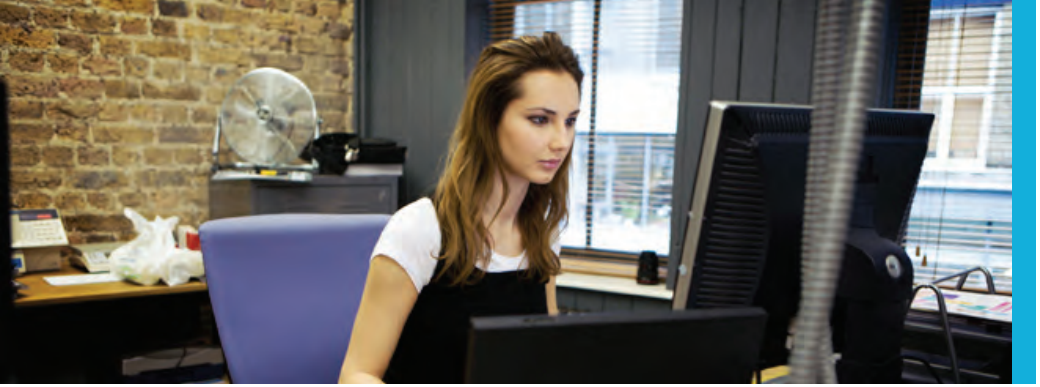
Unit testing during the development process — now better than ever.

As developers work on their code, it's critical for them to effectively test it — not only to prove the new code works, but to ensure there's no unexpected impact on other parts of the project.

The new Test Impact View window lets developers view tests that need to be run resulting from any code change. The developer can toggle between two views:

+ Impacted Tests: Shows a list of tests that need to be run and which code changes are covered by each of the tests.
+ Code Changes: Shows code changes and which tests must be run in order to validate each of them.

So developers can run unit tests in less time, which in turn saves valuable development time.



Too many bugs, and most point to one guy.

The team has been meeting every day to track progress. Two weeks in and there's a pattern emerging. It's not a pretty pattern.

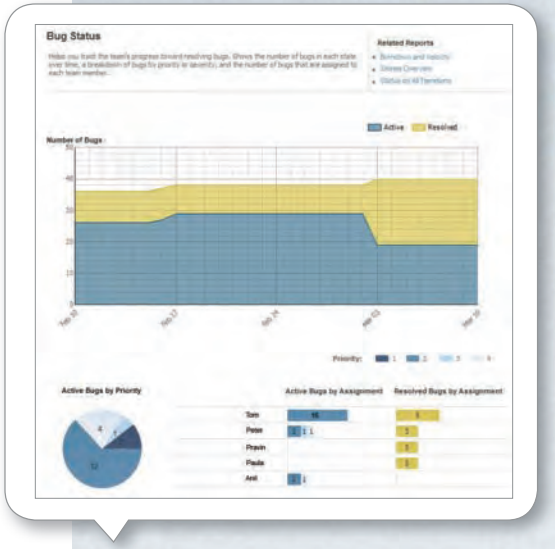
The project manager is reviewing a bug trending report and sees that Tom, one of the developers, is racking up far more than his share of bugs. About 70% of the bugs are assigned to him.

Tom's job? The event-planning system, which should be one of the more straightforward project parts to tackle. She immediately puts a plan in place to a) make sure his code performance improves, and b) protect the repository from his mistakes. The project manager takes advantage of the new tools in Visual Studio 2010 to enforce these policies, so the lead developer doesn't waste time trying to manage this guy.

Moving forward, Tom's code will be:

+ Checked in to a separate shelf set (gated check-in).
+ Reviewed by the lead developer.
+ Unit tested with each check in.
+ Validated by the Layer Diagram to ensure he's not violating the constraints set by the architecture.

But Visual Studio 2010 tools can only do so much. **The project manager is realistic, and begins interviewing contractors to take over Tom's work if he doesn't make it through probation.**



Bug reports make it clear that Tom's not writing high-quality code.

Tip

Gated check-in keeps code healthy.

Team Foundation Server 2010 includes a new feature that lets you define code change requirements, and dictate code is rejected if it doesn't clear policy requirements like architectural validation and passing unit tests.

Changes that DO pass policy requirements are committed to the version control repository on behalf of the user who submitted the check-in. The developer can keep tabs on the progress of this validation process and is alerted when her code is cleared.

By enforcing gated check-in for the whole development team, you don't have to worry about build breaks.

Sprint 2

Overcoming the inevitable obstacles.

The client has reviewed the initial milestone, and with a few minor tweaks, the team is ready to dig into the second sprint.

+ Create distributor approval workflow
+ Create customer and distributor profiles
+ Add Silverlight™ media controls for streaming video
+ Design the recommendation engine
+ Fix the event-planning system

The Silverlight pieces.

Fabrikam is adding enough Silverlight touches throughout the site to make the experience richer and more rewarding for the customer. Visual Studio 2010 has full support for Silverlight, which makes it easier to deliver the polished experience that Coho expects.

The team decides to create a Silverlight video player to allow customers to watch video clips pulled from local media coverage. Silverlight allows for the video to be high-quality, and is easily integrated into the Web site. The team takes advantage of integrated project templates to quickly add a new project to the Web site solution that contains all the necessary code to host the Silverlight content.

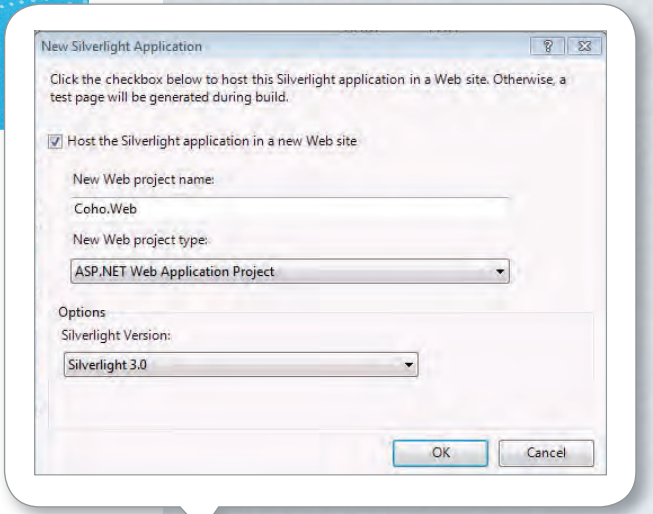
From here, the team starts implementing the video player by dragging and dropping a control from the toolbox, the Silverlight MediaElement, onto the new design surface.

Visual Studio 2010 sets up the project files needed to work on both the Silverlight elements and all the hosting ASP.NET pages. A few things stand out that make this project easier:

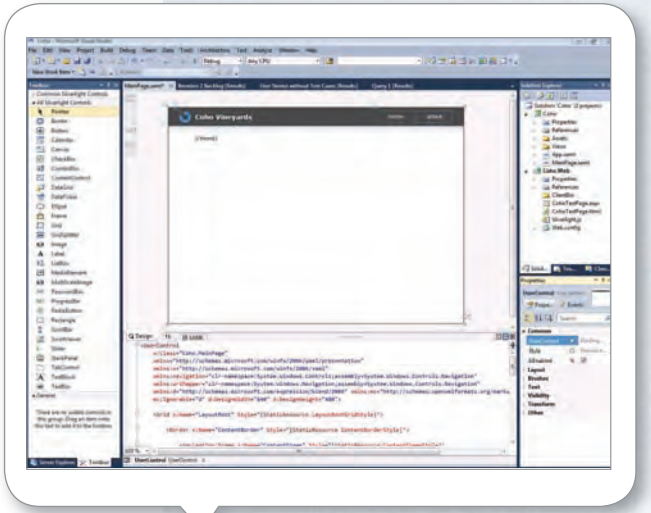
1. **Split-view designer** helps the team visualize changes made to the XAML markup file. Once Fabrikam's designer uses Expression Blend™ to create a stylized video player based on this MediaElement, the developers can continue to make incremental changes through this designer.

2. The **properties window** feels familiar to developers who have worked in Visual Studio before. Fabrikam will enjoy new data-binding wizards that make the integration of data into their Silverlight projects much easier.

3. Starting the **debugger** in Visual Studio reveals an integrated debugging experience that enables Fabrikam's developers to set breakpoints and step in/out of their application's C# or VB code.



Fabrikam sets up a new Silverlight project.



The programmer can visualize the changes as he makes them with the split-view designer.

The boy who cried "no-repro bugs!"

Testing is going fast with the new Visual Studio 2010 tools. Tom's bugs are adding up the quickest, but he doesn't seem bothered. He starts rejecting bugs left and right, claiming that he can't reproduce them on his machine, so the QA team must have made some mistake.

The QA team disagrees. Aggressively. They say Tom's being lazy. Now the whole team is frustrated by this drama.

People are starting to get a little snippy and passing blame back and forth. Everyone's complaining that they need more help or they'll be working weekends for the next three months. (And it's summertime — vacations have already been approved!)

This is just one more issue that points to Tom, and the reports are still showing that he's mostly responsible for slowing down the job.

Fabrikam has a couple of options. They can slip the schedule or recommend killing the feature Tom's been responsible for. The client can make informed decisions rather than last-minute reactions to cut business requirements.

But early on, the Kano Analysis showed that the event-planning system is definitely a must-have for the client — and visitors to the site — so there's no way they'll recommend cutting it from the feature list. **The event planner stays — but Tom isn't working out.**

On-boarding a new team member.

Six weeks in and things start to get a little stressful. The initial excitement around the new project has worn off. The project manager agrees with the lead developer — she needs to fire Tom and get someone else on board.

Tom is let go, and the project manager knows one thing. . . Unless they get another developer on board — fast — they're not going to be able to hit the September 1 deadline.

Armed with detailed progress reports, the Fabrikam PM makes a case to the management team and gets instant approval to add a new developer. They bring in a top-notch, experienced developer to help drive the project to completion. The new blood will bring energy to the team, and he'll serve as the release/build manager as well.

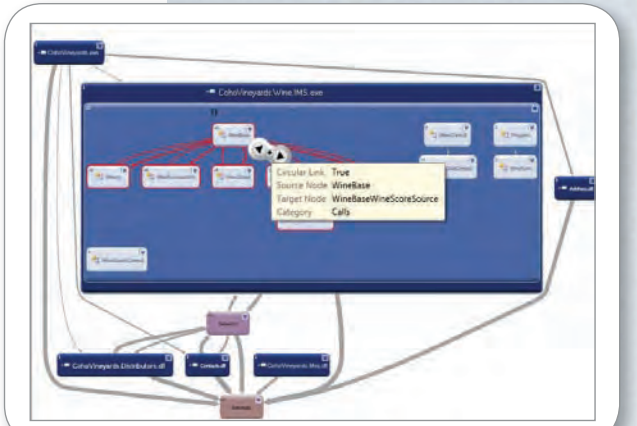
It'll only work if the developer can jump right in and get started coding.

Even though he's got the chops to hit the ground running, it's always hard to work with a new team — and that much harder to jump in partway through a project. The developers are head-down in their work, and no one wants to slow down to show the new guy the ropes.

Not a problem. He can figure out what he needs to know on his first day. Before lunch.

First, he jumps on the team portal to take a look at the requirement rollups. This gives him a sense of what project parts still need the most attention, and what bits have already been created. It's obvious that the event-planning system is critically far behind.

Now what he really needs is a top-level view of how the site is structured. He pulls up the new Architecture Explorer to access the artifacts of the solution. He browses the layer diagrams to see how Fabrikam has been structuring their code, so his new work can map to their logical architecture. So, instead of spending two hours with the lead architect at a whiteboard getting stepped through the solution, it's baked right into the code.

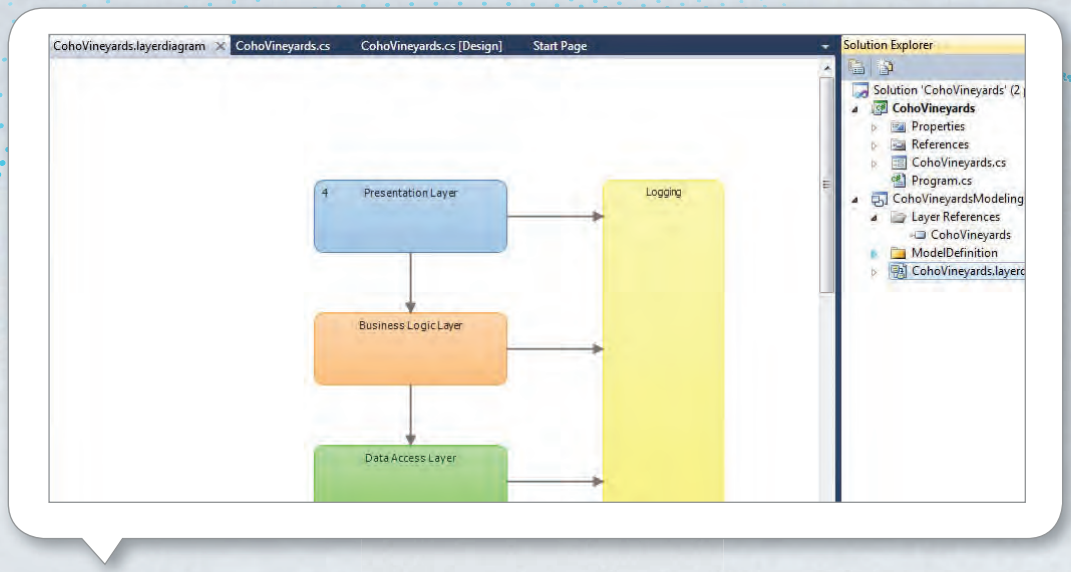


The new programmer opens Architecture Explorer to navigate the relationships between the various classes in the project.

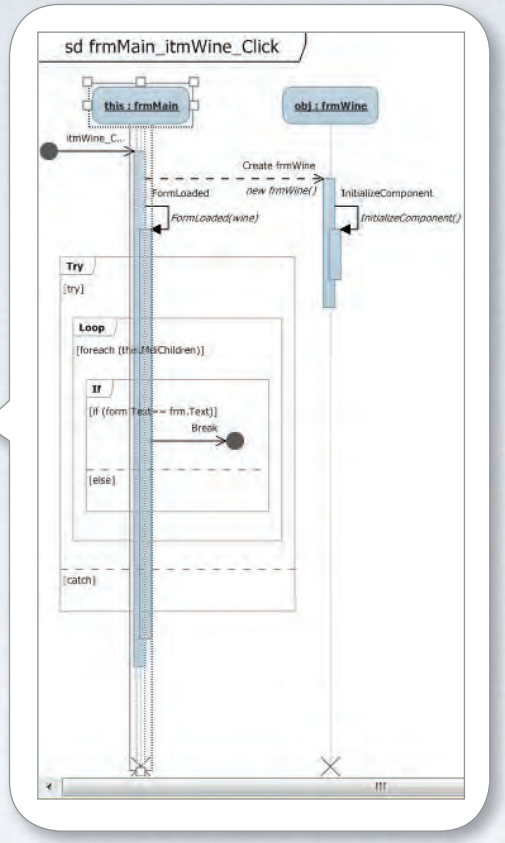
New

At last! Multi-monitor support.

You've been asking for it, and now Visual Studio 2010 offers multi-monitor support. Now comparing code side by side is going to be much easier.



Layer diagrams show the layers of the project and its constraints, to ensure that the architect's intentions are maintained.



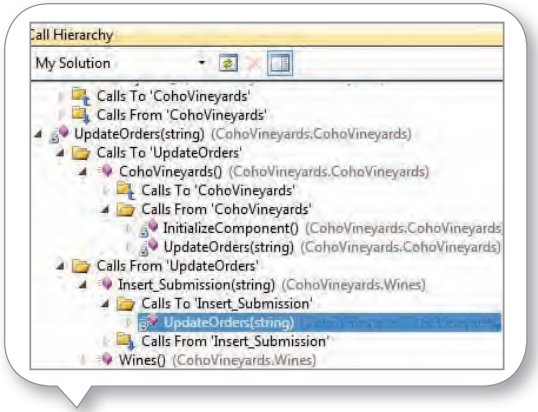
Sequence diagrams highlight areas where code can be optimized.

Now he's ready to dig deeper and make sure that his code aligns with Fabrikam's work on the method level. He snaps to his second monitor and rolls up his sleeves.

The new sequence diagrams feature in Visual Studio 2010 gives him a visual representation of how the code is structured. It helps him quickly understand how it ties together, without reading every line of code. More importantly, Sequence diagrams highlight what's out of place — not that the code is broken, per se — but where he has opportunities to optimize the code that's been written.

With the same code, he's looking at an insert_submission method, which takes the user's inputted information and submits it to the database. He notices that it's being called from twice, which will create duplicates in the database. Not good. He makes a note that one of these will need to be deleted later on.

He also wants to quickly see all references to insert_submission in the code file to further navigate the code and get a sense of what's broken. He just clicks the insert_submission symbol, and all instances of that method are highlighted within the code. This new feature, called Highlight Reference, might seem like a small thing, but it's a huge time-saver when it comes to scanning code without invoking the debugger.



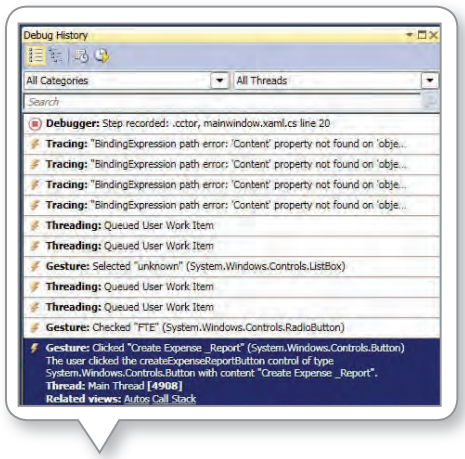
Call hierarchy highlights all of the calls into and out of any given method.

He grabs a sandwich, gets back to his desk, and opens up Team Foundation Server to check out the bug trending. It isn't pretty. . . he can see why everyone's getting stressed out. He asks around and hears about Tom's no-repro issues.

"What!?"

The new guy is surprised the Fabrikam team hasn't used the new Microsoft Test Runner and IntelliTrace™ debugging feature in Visual Studio 2010.

He quickly shows them what it's all about. Test Runner enables manual testers to run through the steps outlined in their test plans while automatically recording their interactions with the user interface. When an error is discovered, the submitted bug contains the execution history of the application, so the developer can see all the events that happened before and after the point of error.



IntelliTrace captures a running application history during debugging…with video.

What's IntelliTrace?

IntelliTrace increases debugging productivity by reducing the time required to reproduce and diagnose an error in your code.

Unlike the standard debugger, which only shows the state of the system at a specific time, IntelliTrace captures and records what the application does when it is running on your local machine or remote server. When an error occurs, you can view the state of the system at any time from the start to the point of the error. While stepping through the events that happened during the application's execution, developers have access to the full debugging context, including watch window, call stack, immediate window, and breakpoints window.

Additionally, because testers can record a historical debugging session in the environment where a bug occurred, Visual Studio reduces the chance of not being able to reproduce a bug.

What's more, while the test is running, Fabrikam's QA team can now start capturing screens AND video during the test. When they find bugs, it's added to Team Foundation Server with the video and screenshots attached — fully indexed with the test steps as bookmarks. The developers can see exactly what happened and start working on a fix.

Protecting builds at all costs.

From past experience, Tom's replacement knows his primary responsibility as release/build manager is to preserve the stability of the code repository. Tom's work put the source code repository at risk, and when he checks in code that breaks the build, the whole project goes sideways. Not only is it a hassle to find what's broken and fix it, the team needs to ensure Tom's shoddy code doesn't make it into the Beta.

To check, he starts using the new actionable diagram of merges to see where Tom's code has been merged and remove it before the Beta is released to the client.

Sidestepping unnecessary tests.

The project is back on track, the team is working together more smoothly, and Fabrikam is actually ahead of schedule. . . for now.

Now that the new developer is up to speed, Fabrikam starts making good use of the new test features in Visual Studio 2010.

Before he left, Tom had written dozens of unit tests. But the team doesn't need to just throw them out — and frankly, they're not all bad. Visual Studio 2010 automatically determines whether or not the unit tests apply.

When the developers make and save code changes, the new code is compared against the last set of unit tests that ran on the server. Based on the comparison, a list of recommended tests is returned in the Test Impact View windows. The developer reviews the list and runs only the necessary tests. **Time isn't wasted, and the Beta stays stable.**

The Beta is released to the client. High-fives all around, and two for the new guy.

Milestone two complete, and the client is thrilled! Well, mostly.

Coho is delighted with Fabrikam's work, and they don't know a thing about the drama around replacing a team member.

They're so happy, in fact, that they are starting to panic. *How is the company going to handle all this new business?* It's nice that some of the process is automated, especially around adding distributors. The problem is, it's going to be four times as much work to manage all the distributor orders. E-mail and a fileshare server aren't going to cut it any more.

And of course, the client is going on vacation.

Before he goes, he asks the team to take on another project for Coho. They have the budget, thank goodness, and the challenge is this:

Create a system to manage the immediate growth and scale for the future.

The project manager doesn't bat an eye. Fabrikam can create a SharePoint® solution in a matter of weeks. Since she used the capacity-planning tool to determine how much of a buffer they may need, they have 10 business days to play with — just the amount of time the client is taking off.

She plans a quick 2-week interim sprint to focus on the new need. It won't be that hard, and the team has plenty of experience.

Fabrikam proposes a SharePoint solution that:

+ Upgrades Coho's backend systems — leveraging current and previous data — to give management better insight into finances and forecasting.
+ Gives real-time reporting for the Coho management team, complete with a customized KPI dashboard.
+ Has a central management system for their business-critical information. This will give the management team an up-to-date, accurate visibility into the company's financials, marketing plans, costing and forecasting, and more.

This would have been a 4-week idea, but Visual Studio 2010 tools helped make it significantly easier. . . and a whole lot faster.

The SharePoint® Sprint

Gotta love last-minute requests.

Building out the SharePoint parts is going to be much more straightforward and efficient with the new tooling support and templates available with Visual Studio 2010. The new integrated features should substantially speed up the development process.

It'll be easy for Fabrikam to pick up Coho's core business documents and design a suitable SharePoint solution using the new designers and tools available in Visual Studio 2010.

Creating the KPI dashboard.

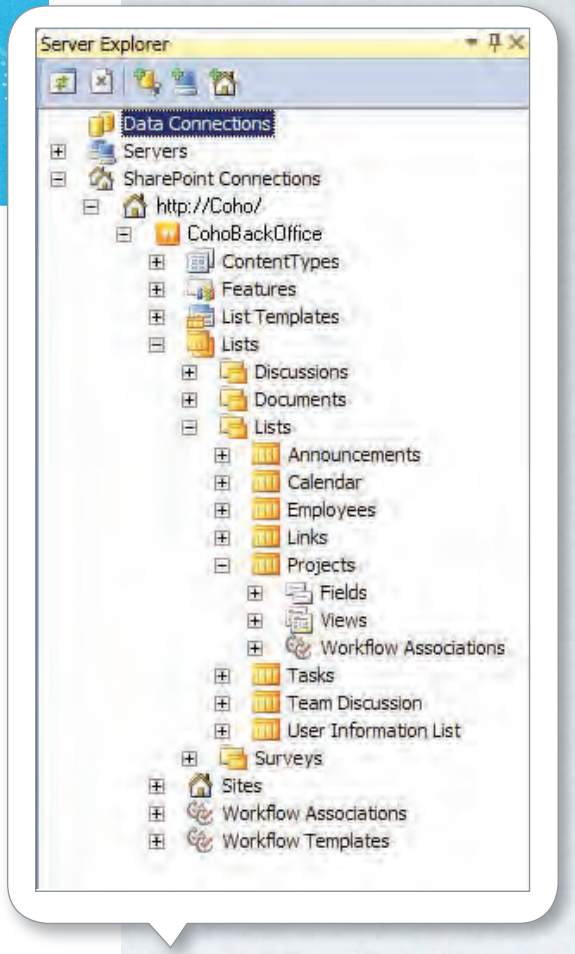
Getting in to major distribution channels is clearly a huge opportunity for Coho. But they're smart — they want to see real-time metrics that prove the investment of time and money is worth it. Detailed reporting is important, but the management team is going to be more focused on some key financial and non-financial metrics to keep their eye on the big picture.

The Fabrikam SharePoint team has built reporting tools before, but they've never been as cool as this.

The new SharePoint tooling in Visual Studio 2010 makes it easy for them to create a tricked-out solution that will blow the Coho management team away. Not only does it LOOK better than traditional online reporting, it's going to allow the team to customize their dashboards and give them a more intuitive way to dig into the data.

First, Fabrikam wants to leverage Coho's existing inventory-management solution without creating a new database. The team discovers the new enhancements found in the Business Connectivity Service (BCS), which allow SharePoint to reach out and connect to the existing system and retrieve the key metrics Coho management requires. This also means Coho is able to update their existing system with new data via the custom SharePoint Web part.

Once they've collected all the business data, Fabrikam wants to help the management team at Coho visualize trends — and trouble — by visualizing the reporting. Fabrikam had limited capabilities to do this in the past, but Visual Studio 2010 lets them use Silverlight to create SharePoint Web parts that pull data from both the line-of-business application AND from SharePoint, then combine the two (or more) to create a single interactive report.*



The team can navigate the SharePoint assets — directly from Visual Studio.

* These capabilities will be available in SharePoint 2010.

Fabrikam builds a report to help Coho determine optimal pricing to hit their revenue goals.

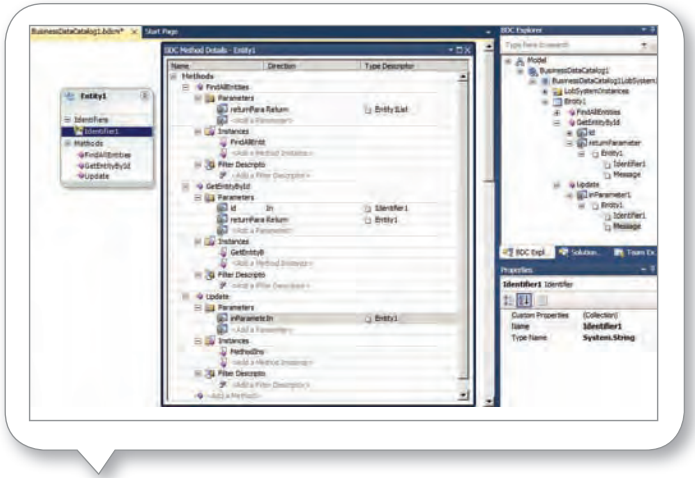
The Silverlight Web part pulls data from their line-of-business system to determine month-over-month sales for Merlot versus Cabernet. At the same time, the Web part collects financial data from SharePoint and maps the two against each another. The management team can then apply different price points in the model to see how they can reasonably adjust pricing to hit their target.

Turns out that increasing the unit cost of Merlot by $1.40 for consumers and $.70 for distributors puts them at their sweet spot. This mini financial model can also be used to determine sale pricing to reduce inventory, and to optimize profits for vintages in high demand because of national reviews.

Not only will this help with visualization, the end-user can INTERACT with the data. . . zooming in, shifting axises, changing date ranges. . . which ultimately gives the client greater insight with less work.

How is this finally possible? With Visual Studio 2010, the client object model lets the developers build Silverlight Web parts that can access SharePoint functionality from client-side calls. The team sets up the solution to simply call the Client Object Model. It's local. And it mimics what will ultimately happen with the SharePoint server.

The SharePoint solution is good to go once the client returns from vacation. Score!



Fabrikam can surface line-of-business data to the end user.

New

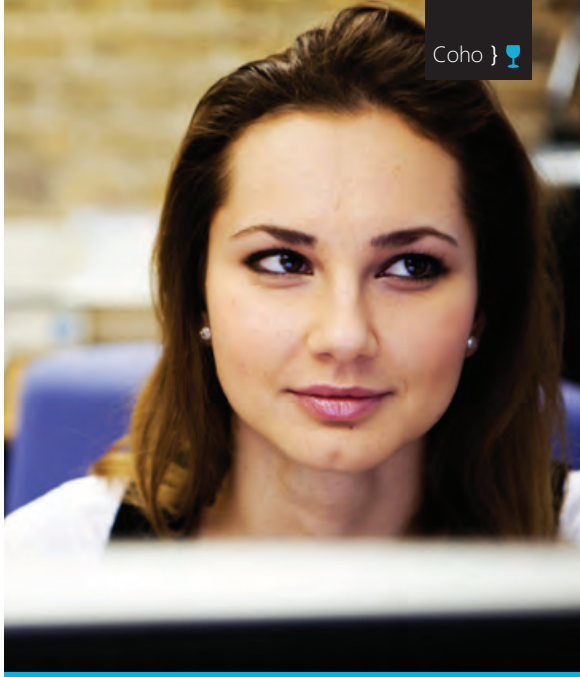
Visual Studio 2010 will change the way you customize SharePoint.

Now you can:

1. Build, test, and debug content types on your local development machine running a client Windows operating system.
2. Pull data from various sources, including SharePoint lists, Web services, On-site databases, XML files, and more.
3. Display data in Silverlight Web parts.
4. Use SharePoint Explorer to view your SharePoint assets directly within the Visual Studio IDE.
5. Automate deployment as the end-user, effectively removing the SharePoint administrator as the middleman.

Sprint 3

The last push to project completion.



It's been a busy five months, and they only have a few weeks left to hit their deadline.

The final sprint in this project should actually be the easiest. Fabrikam has certainly hit a few hurdles — especially within the team — but overall, the development is moving along smoothly and everyone is collaborating better than ever.

Now it's just down to:

+ Flowing in live copy.
+ Finalizing the graphics.
+ Creating trigger e-mails for profile changes, orders, and shipments.
+ Bashing the last bugs.

Or so they thought.

They can't afford the servers they need.

All the core features are in place and the design is locked. The client is happy and can't wait to see their new applications in action. That's right when Coho realizes they don't have the capacity to host it on the servers they already have, and don't have the cash to spend on new ones. It's four days before September. Whoops.

Fabrikam proposes deploying to Windows Azure. It's the perfect solution. It's scalable to handle shifts in traffic, and it's inexpensive to start up. True, they've never deployed to the cloud before, but they've solved plenty of tougher challenges on this job so far.

Taking the leap to deploying to Windows Azure.

Since this is their first time developing for Windows Azure, the team is curious to see how development will be different. They dig in and discover their experience translates well. They can quickly tweak the site for the cloud using their current skills.

Coho Solution 90% complete

The final product.

Coho's solution is up and running, right on schedule. They couldn't be happier — or more relieved. When they started, Coho's management team was overwhelmed and nervous about getting it done in time for the holiday launch.

Now, not only do they have a high-quality, secure solution, distributors are impressed at the level of technology a small vineyard has adopted. It's clear that Coho takes the wholesale business seriously, and the new Web site and ordering system make distributors want to do business with Coho — it's that easy to use.

And customers love the new site, too. It looks and works great, and does a much better job of supporting the Coho brand. The event-planning tool is getting hit with dozens of requests, the nearly 80% of visitors to the site checking out the video on the home page. The best part? Coho can see the difference Fabrikam's solution makes. It's right there on the KPI dashboard, with sales sharply increasing and revenue going through the roof.

The real end result? A high-quality solution and a well-oiled team.

Despite the internal drama during this project, Fabrikam still found the whole process more efficient with Visual Studio 2010. Prototyping, modeling, and design tools got their vision in front of the client quickly. Coding was quick, since they could use past code and update it to the new solution. SharePoint was easier to code for — and the solution was dynamic. Thanks to the new tracking and debugging tools, and the diagramming tools, the Fabrikam team could identify problems, respond quickly, and bring on new team members seamlessly.

What's new in Visual Studio 2010.

Visual Studio 2010 is packed with new features, updates, and enhancements that will change the way developers and development teams work. We can't hit on every little bit we've changed, but you'll find the updates we think you'll be most excited about on the following pages.

Microsoft® Visual Studio®

> Visual Studio IDE improvements.

Simplified appearance and new behaviors.

+ Visual enhancements

The IDE — the entire shell and editor experience — has been redesigned (based on Windows Presentation Foundation, no less) for improved readability. Unnecessary lines and gradients have been removed to reduce clutter, making it easier for you to stay focused on the work at hand.

+ Multiple monitor support

You can now bring document windows like the Code Editor and design view outside the IDE window. This means you're able to view the Code Editor and the design view window side by side. You will be able to edit multiple code files without starting up multiple copies of Visual Studio.

Explore code more quickly.

+ Updated Code Editor

The new Code Editor makes code easier to read and scan. Zoom in on text by simply pressing CTRL while scrolling your mouse wheel. You can now quickly highlight all instances of any symbol in Visual C#® or Visual Basic® just by clicking it.

+ Navigate To

Finally, search-as-you-type is supported for everything in a solution, including file names, text, symbols, and comments. You no longer have to decide among Quick Find, Find in File, Look in Current Document, and so on. Access the Navigate To function using the keyboard shortcut CTRL+COMMA.

+ Call hierarchy

Easily navigate from a member to the members that call it and to the members that it calls. This is especially useful for exploring object-oriented code.

Improved debugging.

A redesigned Threads window provides filtering, grouping, and call-stack searching and expansion. You can also organize and search breakpoints, then share them with other developers.

Built-in Extension Manager.

The built-in Extension Manager makes it easier than ever to download, install, share, and manage templates, packages, and components. You can search the Visual Studio Gallery for extensions and wizards directly in the IDE.

IDE features correspond to project capabilities.

Features in the IDE, such as IntelliSense®, the Code Editor, and the Toolbox, now correspond to the version of the .NET Framework that your project targets. As a result, unsupported types, members, and controls either don't appear all together or they produce background compilation errors. So for example, if your project targets .NET Framework version 2.0, the "var" keyword will not appear in IntelliSense lists.

Support for test-first development.

The improved Code Editor generates new types and members in the background when you write code that references them. That means you get to write your tests first and generate the code later.

Bonus! IntelliSense now uses a consume-first mode when classes and members are used before they are defined.

> C# and Visual Basic languages: Your choice for the full power of .NET.

Dynamic support in both languages now makes it straightforward to interoperate with the new dynamic languages IronPython and IronRuby, and to interoperate with the Document Object Model in Silverlight Web pages.

Microsoft Office programming has been made dramatically easier. Both languages now support late-binding and named and optional parameters, making it easier to call COM APIs. Plus, the new feature of type embedding simplifies your deployment by allowing you to no longer ship Primary Interop Assemblies — the relevant types can be embedded directly in your own assembly.

Everyday code has been made easier to write, with features now in both languages, such as auto-implemented properties and collection initializers and array literals. The type system of both languages has been made more flexible with co- and contra-variance. Both languages now support statement lambdas, which are particularly useful for the new Task Parallel Library and Parallel LINQ.

Each language also has its own characteristic enhancements. C# has a dynamic keyword to restrict the scope of late-binding. Visual Basic has implicit line continuations that let you split statements or LINQ queries over multiple lines.

Now it's easier to stay compatible with different versions of the .NET Framework: multi-targeting lets you develop a project in Visual Studio 2010, using the new features of both languages, while still targeting an earlier version of .NET. And side-by-side lets a single .NET executable link with assemblies built against different versions of the framework.

> Visual C++® development.

User experience.

A number of improvements have been made to make the Visual C++ IDE more responsive, especially when you work with large applications. Many IDE components, like the IntelliSense engine, parse and process files in the background. That means you don't have to wait for Visual Studio to re-process the entire project when you just modify a header file, switch between project elements, or reconfigure your project from debug to release.

You also get more accuracy from the updated IntelliSense features — it handles advanced C++ code constructs and conditionally defined macros that the old version couldn't process.

Targeting specific compilers and libraries.

In Visual Studio 2010, Visual C++ projects can target either the toolset in Visual Studio 2010 or in Visual Studio 2008. You get the IDE enhancements in Visual Studio 2010 while using an older version of the Visual C++ libraries and compiler. To switch between toolsets, you only have to change one property in your project file. Easy.

Build improvements.

Visual C++ projects use a new file format (.vcxproj) that replaces the old format (.vcproj). To build projects, you now use MSBuild.exe instead of VCBuild.exe. MSBuild is the build platform for Microsoft that includes better diagnostics, extensibility, and integration.

Windows 7 support.

Visual Studio 2010 adds support to a number of Windows® 7 and Windows Vista® features in Microsoft Foundation Classes (MFC) applications. Multi-touch support makes it easy to take advantage of multi-touch input and manipulation. A new Windows 7 look-and-feel for the MFC Ribbon and a visual designer for the Ribbon interface make modernizing your application's UI a breeze. Applications can also light up on the Windows 7 shell through MFC support for Jump Lists, Aero thumbnails in the task bar, customized common file dialogs, and rich icon preview. Finally, restart manager support lets your application restart itself and restore automatically saved user data in cases of accidental restarts.

New C++0x language features.

The Visual C++ compiler introduces five new features to support the C++0x standard:

+ Lambda expressions
+ rvalue references
+ Compile-time assertions
+ Expression type discovery
+ Automatic type deduction

Code generation enhancements.

The Visual C++ compiler in Visual Studio 2010 made some noteworthy improvements in compilation speed for some key scenarios:

+ Link-Time Code Generation (LTCG) compilation speed has improved significantly.

+ Profile-Guided Optimization (PGO) instrumentation runs are now about 1.7x faster.

> Visual F# development.

With Visual Studio 2010, you can use the F# language for application and component development. F# is a new .NET programming language based on a combination of functional programming and object-oriented programming. The F# language is ideally suited for parallel, algorithmic, explorative, and technical programming tasks. Important features include:

+ Immutability, making F# code more easily parallelizable and de-coupled.
+ Type-inference, giving F# a succinct and expressive syntax, combined with the type-safety of static typing.
+ F# Interactive, providing an explorative environment for interactive development with F# inside Visual Studio.
+ Asynchronous workflows, substantially easing the development of asynchronous code on .NET and Silverlight.
+ Units of Measure, enabling F# code to deeply track and verify units like meters and seconds.
+ Deep .NET integration, ensuring that F# components can comfortably plug into any part of a .NET application.

> Web development.

Faster code authoring.

+ Code snippets

Visual Studio 2010 includes code snippets for HTML, JScript, and ASP.NET controls to help you write and share code faster. You can insert snippets from the code snippets manager or directly from IntelliSense.

+ Dramatically improved IntelliSense for JScript

- + IntelliSense now parses two to five times faster, so there is virtually no processing delay, even with large script libraries.
- + IntelliSense accepts diverse coding styles, so you get full IntelliSense support for almost every library.
- + IntelliSense displays XML documentation comments as you type.
- + IntelliSense support for jQuery library.

Simplified Web deployment.

With Visual Studio 2010, you can package and publish your Web application in one click.

+ Creating Web packages

The Web Deployment Tool, also known as MSDeploy, enables you to package your Web application for deployment to an Internet Information Services (IIS) Web server. A Web package is a .zip file or a folder structure that includes everything a Web server needs to host your application, including content, IIS settings, database scripts, components, registry settings, and certificates. And it's fully integrated, so that's one fewer step you have to manage.

+ One-click publishing

You can now publish to a server by using the Web Deployment Tool, FTP, folder copying, or FrontPage Server Extensions in one click. Visual Studio stores all the setting information, such as publish method, server information, and user credentials.

+ Web configuration transformations

You can now configure your project to transform the web.config file during deployment. When you deploy the project, the settings in web.config automatically match the settings on your debug, staging, and production servers.

> Lab Management capabilities.

Visual Studio Lab Management 2010 blends virtualization with Application Lifecycle Management, making it really easy to create virtual environments and improve developer/tester collaboration.

Virtualize your Application lifecycle with Visual Studio 2010.

With Lab Management, you can quickly create multi-machine virtual environments, deploy your application, run tests and file rich actionable bugs.

+ Setup virtual environments quickly

Visual Studio Lab Management is built on top of System Center Virtual Machine Manager and it lets you create virtual environments with multiple virtual machine in minutes. It also makes it really easy for you to connect to the environment and take snapshots. You can also create network isolated copies of the same environment so as to clone the virtual machines without IP or machine name conflicts.

+ Build, deploy, and test your application on a virtual environment

You can set up a workflow to use a virtual environment and configure how you want to run your build and then deploy your application on the environment. You can then run automated tests on this environment and review the results of these tests. This makes it really easy to automate application deployment and testing, and hence increases developer and tester productivity.

+ Create reproducible bugs that include the environment used for testing

When you are running your manual tests, you can create a bug that includes a snapshot of your virtual environment when the bug occurred. The developer can open the bug and connect to the snapshot for this environment to see the actual state of the environment at that point in time. This can greatly reduce the time for the developer to reproduce and fix a bug.

> WPF and Silverlight Designer.

Creating Windows Presentation Foundation (WPF) and Silverlight applications is swifter and easier, thanks to various designer improvements added to Visual Studio 2010.

Improved support for Silverlight.

Visual Studio 2010 comes with an intergrated Silverlight designer that allows you to go beyond just read-only Preview. For example, in Silverlight projects you can now select and position items with the mouse on the designer surface.

Drag-and-drop data binding for WPF.

After you add a data source to your project, you can generate data-bound WPF controls by dragging items from the Data Sources window to the WPF Designer.

Visualizing WPF objects.

The debugger in Visual Studio 2010 includes the WPF Tree visualizer, which displays WPF objects in a more useful way. When you invoke the visualizer on a WPF object from the Watch window, you see the object displayed as a tree and a list of properties.

> SharePoint tooling.

Visual Studio 2010 introduces new SharePoint® tools in an effort to make SharePoint development easier and more intuitive for .NET developers. These tools include new project and item templates you can use to create SharePoint sites, lists, document libraries, workflows, and other types of SharePoint content.

Visual Studio 2010 also provides new visual designers for Web parts, Business Data Connectivity (BDC) models, and packaging. And you can now use Server Explorer to connect to your SharePoint sites and navigate the hierarchy and contents of these sites from within Visual Studio.

And when you need more customized solutions, you can develop custom Web parts within Visual Studio 2010, or use the extensibility API to add on to the tools and share them with the community.

> Azure tools for Visual Studio 2010.

The cloud is new — and it's certainly what's next. Visual Studio 2010 gives you all the tools you need to deploy to Windows® Azure™ with ease. And your team can simply extend their current skill sets. Testing, debugging, and deploying are incredibly similar to the .NET framework, so if they've done it for the client, they can build it for the cloud.

Windows Azure Development Fabric.

The development fabric simulates the Windows Azure fabric on your local computer so that you can run and test your service locally before deploying it. The development fabric allows you to debug and fine-tune the behavior of your service before it is deployed. The development fabric utility provides a user interface for observing and managing local service deployments.

Native debugging support.

Debugging for the cloud is built in to Visual Studio 2010. No need to change your QA process when you test on your local machines.

> Parallel programming.

You can now write programs that distribute work across multiple processors without having to work directly with threads or the thread pool. Visual Studio 2010 includes parallel computing libraries for both the C runtime library (CRT) and the .NET Framework. You can also debug and profile native and managed applications by using the new multithreaded tool support in the Visual Studio Profiler and the Visual Studio debugger.

Parallel programming in the .NET Framework.

.NET Framework 4 includes new libraries that support task and data parallelism; a parallel implementation of LINQ to Objects, called Parallel Language Integrated Query (PLINQ); and various new data types for the synchronization and coordination of concurrent execution. These libraries rely on a new task scheduler integrated with the .NET thread pool.

Parallel programming in Visual C++.

The C runtime library now includes the Concurrency Runtime, which is a concurrent programming infrastructure for C++. Visual C++ also includes the new parallel patterns library and asynchronous agents library, both of which run on the Concurrency Runtime and expose productive programming models for concurrency to C++ developers.

Parallel programming tools.

The following are some tool enhancements that support parallel computing.

+ **Debugging**

  The debugger includes new Parallel Tasks and Parallel Stacks windows that enable you to do the following:

  + View a list of System. Threading. Tasks. Task instances that are running, scheduled, and waiting in an application.
  + View the call stacks for Task instances.
  + Navigate to code from the debugger.

+ **Profiling**

  The new Thread Execution Data Views enable you to determine details about an application's performance, such as the following:

  + How each processor on the computer is being used by your code as compared to other code at any time.
  + How threads in your program are interacting with one another and with the computer.
  + How threads are migrating across cores.

> Visual Studio 2010 Shell.

The Visual Studio 2010 Shell enables developers to rapidly create and distribute their own custom tools by building them on top of the core Visual Studio IDE. There are two shell "modes" and both are available via a royalty-free license.

The Visual Studio Shell (Integrated Mode) enables developers to distribute a copy of Visual Studio containing only their tools. If Visual Studio is then installed, these tools will be integrated with the rest of the installed Visual Studio components.

The Visual Studio Shell (Isolated Mode) allows developers to distribute custom stand-alone tools and applications. This can be ideal for applications targeting non-developers as it allows you to control and simplify the User Interface — and the users don't need Visual Studio.

Package load key requests and registry requirements are no longer needed to deploy your IDE.

Additionally, all enhancements to core Visual Studio 2010 components are included with the Visual Studio Shell. Developers using the Visual Studio 2010 Shell will have access to the new WPF Shell, the new Code Editor with updated Intellisense, all new debugger and Web features, and a new Help infrastructure optimized for search and deployment.

> Visual Studio Help.

Visual Studio 2010 will come with a completely re-engineered Help system that introduces a new flexible, standards-based Help framework. The standards-based approach delivers not only a much better local experience, but also a seamless transition to an online Web browser, and with infrastructure and tooling much more consistent than other Visual Studio and Internet technologies.

Visual Studio 2008 and prior versions use a proprietary Help system in which content was normally packaged into .hxs or .chm file formats. Now, Help content will be stored primarily as XHTML and will be packaged into .zip-format containers that can hold any type of content, from HTML to XML, from art to videos.

5

Five reasons you'll love Visual Studio 2010.

Microsoft® Visual Studio®

Top 5 features for developers:

1. Build innovative customizations for SharePoint
2. Develop applications for Windows 7
3. Understand existing code and architecture
4. Identify test impact from code changes
5. Customize Visual Studio to fit your style

Top 5 features for testers:

1. Enjoy deep collaboration with the development team
2. Fast forward through manual tests
3. Reproduce bugs on a common virtualized environment
4. Automatically attach context to bugs
5. Enjoy full visibility to the test progress

Top 5 features for project managers:

1. New dashboards keep the team in sync
2. Agile planning templates help the estimation process
3. Requirements traceability keeps stakeholders informed
4. Visual Studio Team Web Access eases reporting pains
5. New reports help enable proactive project management

Top 5 features for user-experience and graphic designers:

1. SketchFlow for rapid prototyping and sharing
2. Integration with the development team via Team Foundation Server
3. Improved HTML and CSS editing with support for CSS 2.1 standard
4. Improved sharing of XAML assets
5. New Silverlight design support

Product lineup and system requirements.

Visual Studio 2010 product family.

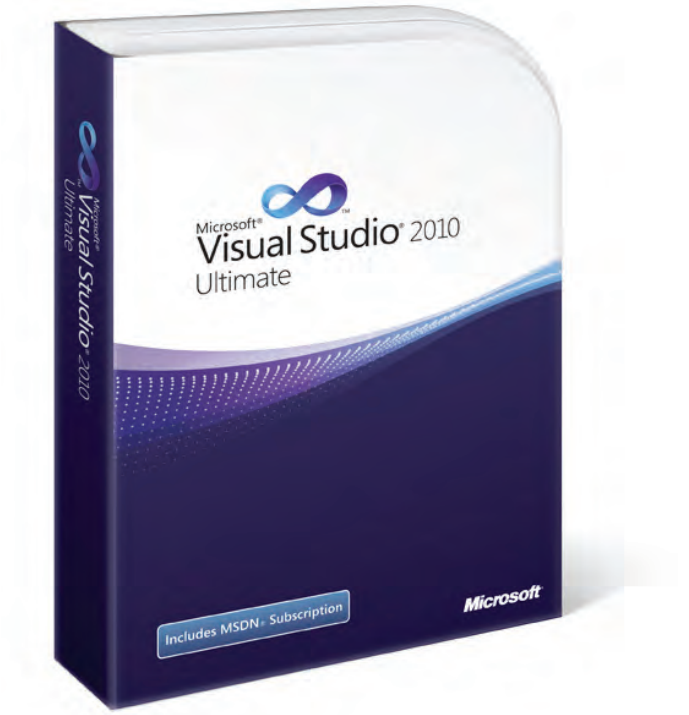
Microsoft® Visual Studio®

Microsoft Visual Studio 2010 Ultimate
is just what it sounds like. Ultimate.

Visual Studio 2010 Ultimate provides an integrated environment of tools and server infrastructure that simplifies the entire application development process.

Your team will be able to better deliver business results using productive, predictable, customizable processes. Detailed analytics and reports give everyone increased transparency and traceability throughout the lifecycle. Advanced collaboration features make the whole team more productive. And integrated testing and debugging tools help ensure that your final product is high-quality while driving down the overall cost.

So whether you're creating new solutions or enhancing existing applications — targeting a wide number of platforms — everyone can focus on bringing the idea to life with powerful prototyping, architecture, and development tools.



Visual Studio 2010 Ultimate system requirements

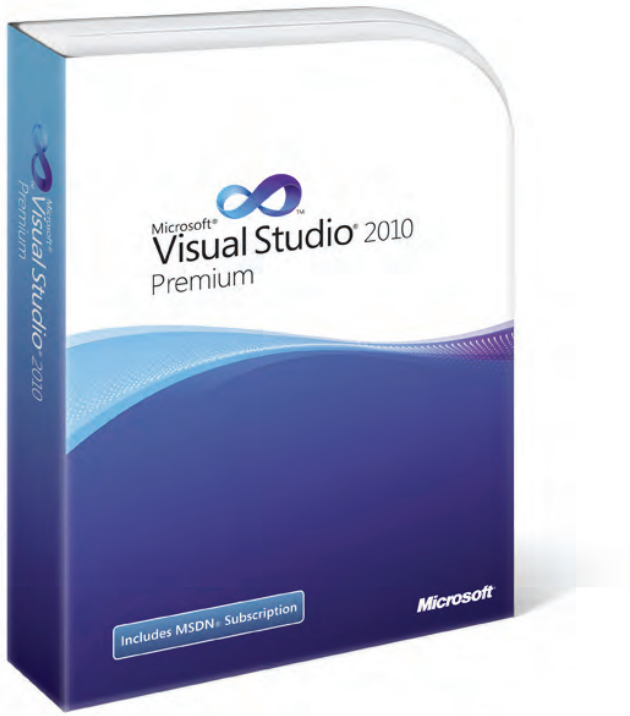
OS	Processor	RAM	Hard Disk Space	.NET Framework	Monitor Resolution	Internet	Other
Windows XP® with Service Pack 2, Windows Vista or Windows 7 operating system	PC with 1.6 GHz or faster processor	1 GB of RAM or more	3 GB of available hard-disk space	.NET Framework 4 (if it is not already installed, it will be installed automatically during setup)	1280 x 1024 or higher-resolution monitor with 24-bit color, DirectX 9 Capable video card	Internet Explorer 8	Microsoft® Office 2007 or greater, DVD-ROM drive

Advanced tools uncomplicate application development in Microsoft Visual Studio 2010 Premium.

Microsoft Visual Studio 2010 Premium provides an integrated environment that simplifies application development, giving you advanced tools that help you tackle your toughest problems.

Whether you're writing code, building databases, testing or debugging, team — and team members' — productivity increases with tools designed to work the way YOU work.

Take control of your code and keep your application and database in sync. Spend less time bug bashing by avoiding them in the first place with advanced code analysis, testing, and debugging tools that let you find and fix bugs as you go. At the end of the day, you're building more scalable and higher-quality solutions.



Visual Studio 2010 Premium system requirements

OS	Processor	RAM	Hard Disk Space	.NET Framework	Monitor Resolution	Internet	Other
Windows XP® with Service Pack 2, Windows Vista or Windows 7 operating system	PC with 1.6 GHz or faster processor	1 GB of RAM or more	3 GB of available hard-disk space	.NET Framework 4 (if it is not already installed, it will be installed automatically during setup)	1280 x 1024 or higher-resolution monitor with 24-bit color, DirectX 9 Capable video card	Internet Explorer 8	Microsoft Office 2007 or greater, DVD-ROM drive

Ready to rock? **Microsoft Visual Studio 2010 Professional** gives small teams a big advantage.

Microsoft Visual Studio 2010 Professional is an integrated environment that simplifies creating, debugging, and deploying applications.

If you're a solo developer or part of a small team with a big idea, put powerful design surfaces to work and bring your vision to life.

Not only can you target a growing number of platforms — including SharePoint and Cloud applications — you can do it all using your existing skills. Integrated support for test-first development and new debugging tools let you find and fix bugs quickly and easily to ensure high-quality solutions. Now you're really a pro.

Visual Studio 2010 Professional system requirements

OS	Processor	RAM	Hard Disk Space	.NET Framework	Monitor Resolution	Internet	Other
Windows XP® with Service Pack 2, Windows Vista or Windows 7 operating system	PC with 1.6 GHz or faster processor	1 GB of RAM or more	3 GB of available hard-disk space	.NET Framework 4 (if it is not already installed, it will be installed automatically during setup)	1280 x 1024 or higher-resolution monitor with 24-bit color, DirectX 9 Capable video card	Internet Explorer 8	Microsoft® Office 2007 or greater, DVD-ROM drive.

Dial up the quality. Dial down the headaches.
Microsoft Visual Studio Test Professional 2010.

Kick up the quality of your development projects with Microsoft Visual Studio Test Professional 2010. It's an integrated testing toolset that gives you a complete plan-test-track workflow.

File high-quality bugs with rich diagnostics for your developers. Focus your time and energy on high-value tasks, thanks to a task-driven user interface with features like Fast Forward for Manual Testing.

Test Professional 2010 is tightly integrated with Team Foundation Server, so you'll have in-context collaboration between all team roles. This gives you better visibility to the overall project, while providing full traceability to user stories and requirements, progress reports, and real-time quality metrics. Test Professional 2010 helps you make informed, timely decisions to drive down the risks associated with your software releases.

Visual Studio Test Professional 2010 system requirements

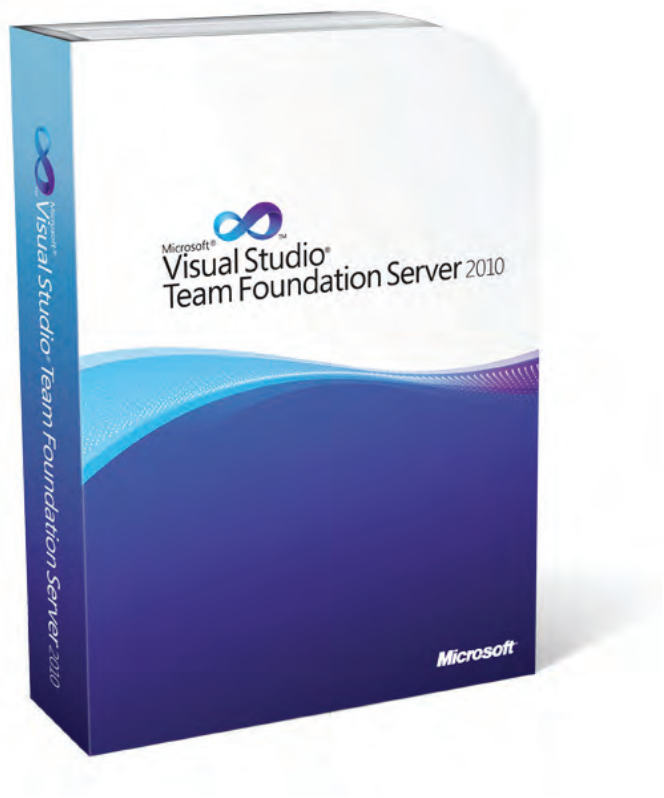
OS	Processor	RAM	Hard Disk Space	Hard Drive	Monitor Resolution	Internet	Other
Windows 7, Windows Vista®, Windows® XP with Service Pack 2 (excluding starter editions). Windows Server® 2003 with Service Pack 1, Windows Server 2003 R2 (all editions).	1.6 GHz or faster processor	1 GB (32 Bit) or 2 GB (64 Bit) of RAM (Add 512MB to host if running in a virtual machine)	3 GB of available hard-disk space	5400 RPM hard drive	DirectX 9 capable video card running at 1280 x 1024 or higher-resolution display	Internet Explorer 8	DVD-ROM Drive

Way. To. Work. Microsoft Visual Studio Team Foundation Server 2010.

When you really need to collaborate, what your team needs is Microsoft Visual Studio Team Foundation Server 2010. It's the collaboration platform at the core of Microsoft's application lifecycle management solution that automates the software delivery process and lets entire organizations manage software development projects from start to finish.

Not only will everyone on the team be able to collaborate more effectively, you'll be more agile and deliver better quality software while building and sharing institutional knowledge.

No one's in the dark. Everything is stored in a data warehouse, from project artifacts and data from work item tracking, source control, builds, and testing tools. Powerful reporting and dashboards provide historical trending, full traceability and real-time visibility into quality and progress against business intent. It's a great way to work together, better than ever.



Visual Studio Team Foundation Server 2010 system requirements

OS	Processor	RAM	Hard Disk Space	.NET Framework	Monitor Resolution	Internet	Other
Microsoft Windows Server 2008, 2008 R2, 2003, 2003 R2 Microsoft Windows 7, Windows Vista	2.2 GHz or faster processor	1 GB of RAM or more	8 GB of available hard-disk space	.NET Framework 4 (if it is not already installed, it will be installed automatically during setup)	1280 x 1024 or higher-resolution monitor with 24-bit color, DirectX 9 Capable video card	Internet Explorer 8	Microsoft® SQL Server 2008, Windows SharePoint Server 3.0 SP1, DVD-ROM drive.