

Users' Choice: 2009 Software Development Platforms

A comprehensive user satisfaction survey of over 1200 software developers

June 2009

Evans Data Corp, 740 Front St, Santa Cruz, CA 95060
www.evansdata.com (800) 831 3080

Users' Choice: Software Development Platforms

INSIDE THIS REPORT

- 1 Overview**
- 4 Overall Rankings**
- 6 Importance of Attributes**
- 7 Attribute Relative Rankings**
- 25 Overall Ratings by Development Platform**
- 25 Delphi**
- 27 Eclipse**
- 29 Rational Tools**
- 31 Visual Studio**
- 33 NetBeans**
- 35 IntelliJ**
- 36 Sun Studio**
- 38 JDeveloper**

Overview

In the spring of 2009, Evans Data Corp conducted a worldwide survey of developers with the intent of finding user satisfaction with the features of various tools and platforms for software development. This was the fourth year that we've conducted this user satisfaction survey. In the past it was entitled an "IDE" study, but it has become clear that most vendors' offerings are not constrained to an IDE but instead are a full suite of tools that extend from architecting to development to testing and often beyond. For 2009, over 1200 developers participated with a proportional spread across regions. The survey was conducted online and was translated and posted in French, German, Portuguese, Japanese, and simplified Chinese, as well as in English.

Only users of a particular development platform were asked to rank the features of that vendor's offerings. Thus we've restricted rankings to only those developers who are actually users of the tools and environments.

The developers were asked to rank eighteen different features commonly found in development platforms. Each feature could be ranked as "excellent", "very good", "adequate", "needs improvement" or "NA". During processing, the first four rankings were assigned a numerical value according to their relative significance. The last value ("N/A") was discarded. Values were then combined to produce a score for each element and also for an overall total score. The relative values for each element ranked are displayed in graphs with the same scale, so that comparisons between graphs can

By Janel Garvin

be made. The numbers in the graphs only display relative values.

Offerings that were user-ranked and had enough users to be included in the study were:

- Eclipse
- Delphi
- Rational Development Tools Suite
- IntelliJ
- Visual Studio and Tools
- NetBeans
- JDeveloper and tools
- Sun Studio

Eighteen features and capabilities were rated by the users. These were:

Basic tools - editor/debugger/ compiler and/or interpreter

App Modeling tools

Web Design/Development Tools

Support for Parallel Programming

Ability to integrate tools

Availability of 3rd party tools

Quality of Tech Support

Size/quality of developer community

Documentation

Sample apps

Runtime memory analysis tools

Visual tools and declarative support

Ready to use out of box experience

Database development tools

Integration with databases

Support for frameworks

Test tools including Code Coverage tools

Support for Remote development

The goal was to shed light on the different characteristics of each platform and toolset and to show the strengths and weaknesses of each. The design

philosophy behind integrated toolsets can differ significantly as vendors target different market segments and provide different capabilities focused on the needs of those segments. Some are aimed at large enterprise developers, while others are designed for individuals or small teams. Some are Open Source and grow and mature organically, while others have received the benefit of millions of dollars worth of research and development, in addition to market research and user testing.

So, even though IBM's Rational tools suite came out as the highest ranked by its users, that's not to say it's the right choice for everyone. Companies with small staff and limited budget should look into NetBeans or IntelliJ for Java development, or possibly Delphi. If there's time and expertise to explore Eclipse plug-ins and configure a system on Eclipse then that will provide a platform that is free but that can be tailored to your particular environment. Anyone who already has an Oracle database in place should give some serious consideration to JDeveloper, as its database tools and integration got the best scores from users. Similarly, you might want to try Visual Studio if you're looking for a ready to use out of box experience with visual tools and wizards. Of course, if your software development is critical to your business and you haven't got database restraints, then you probably want to go with Rational. If you're anything like their current users you will be the most pleased with that selection.

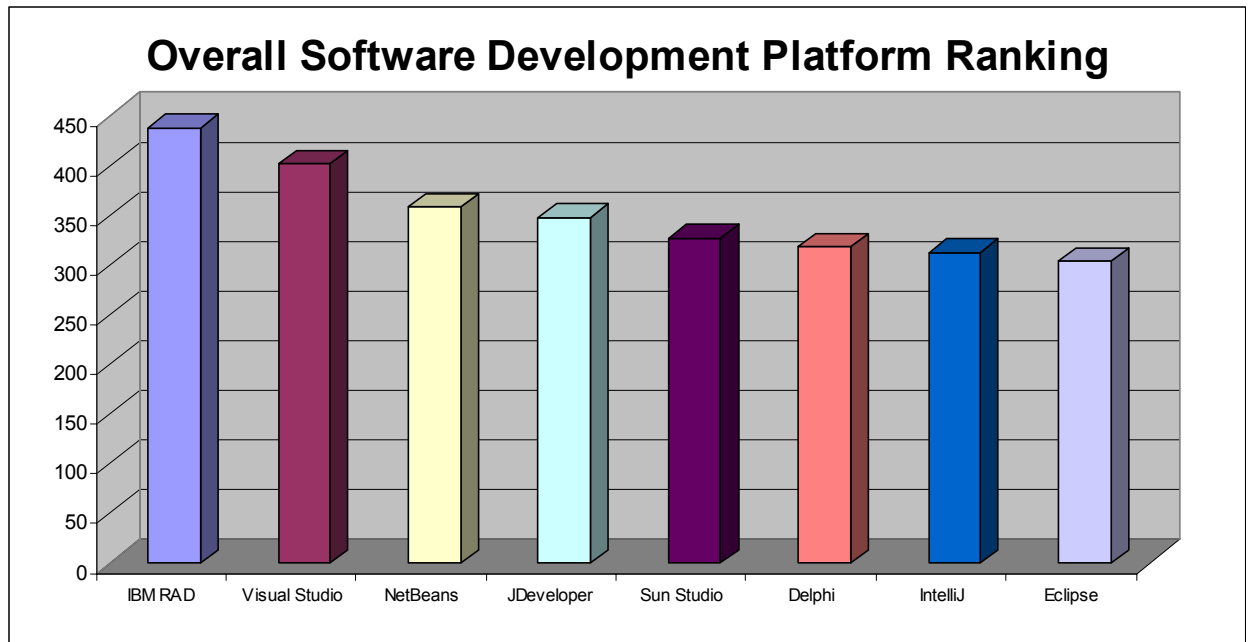
This study shows how well each has done in pleasing users.

Overall Rankings

Once again, IBM's Rational team has taken top honors in our annual survey of developers and their satisfaction with their development platform and tools. This survey is conducted worldwide and gathers over 1200 developer opinions. It is conducted in English, French, German, Chinese, Japanese and Portuguese. It directly asks the users of these toolsets to evaluate and rate the individual attributes and only the users rank the products. We've been doing the survey for four years now and in three out of four years the Rational product team has pulled in the highest satisfaction levels. It's abundantly clear that the Rational team knows what developers want, that they understand the value of top quality, and that their users absolutely love their tools.

"It's abundantly clear that the Rational team knows what developers want, that they understand the value of top quality, and that their users absolutely love their tools..."

And, they love them for good reason. IBM has spent a lot of time and effort to evolve their set of tools into a complete software delivery system, and they have succeeded. This is no set of disparate tools thrown together in an attempt to bridge the problems of developers as they move from design to coding to deployment. Rather, the Rational tools suite, integrated together in the Jazz platform, take software



development into a new realm which blends together requirements, asset management, architecture, development, QA, deployment and further to touch on operations and maintenance.

In the past we have called this survey an IDE user satisfaction survey, but this year when we polled the product managers for updates and suggestions for attributes to rank it became clear that several vendors were providing development solutions that extend

beyond the IDE. Also, the questions we were asked to include embraced more than just IDE functionality. So the survey has evolved into a user satisfaction survey on the whole integrated development system whether or not it is entirely included in a formal IDE or like Rational is a rich set of tools integrated on the Jazz platform.

Another thing to note about this year's survey is that it was conducted in Spring 2009 while Sun was an independent company. It still is as this is being written and these pages reflect that, but the landscape is about to radically change with Oracle's purchase of Sun. Not only will Oracle now have the Sun hardware, Solaris, and MySQL, but it will also gain Sun Studio and the parallel programming tools it features for high performance computing, as well as the stewardship of the Java community and NetBeans that Sun has been leading up. Thus, we can expect some significant changes in the next year along with the answer to some significant questions. How will Java developers react? Will NetBeans, supported by its own development community fare as well without Sun? What parts will Oracle take from Sun Studio to enhance its own tool suite? We will see in the next year.

“Not only will Oracle now have Solaris, and MySQL, but it will also gain Sun Studio and the parallel programming tools it features for high performance computing, as well as the stewardship of the Java community and NetBeans...”

Importance of Attributes

Just as we asked the users to rate the individual features of the tools offerings, so we also asked them to rate the importance of the features they were rating. And, in a similar fashion we assigned positive and negative values to the importance categories to come up with a relative ranking for each of them. This relative ranking was then applied to the scores for each product family to produce a weighted ranking, which is shown on the previous page. The table below shows how each feature or quality was rated in order of importance.

Clearly, the most important features for any set of tools are the basic tools themselves. These are necessary to getting the job done at all and so are indispensable. The next two in importance are also related to the core tools experience: documentation allows the user to know how to fully utilize the tools and since sometimes developers have their own preferences in tools, and the ability to integrate outside tools into the tool chain is also highly valued.

Database support is important for obvious reasons. Most applications need to interact with a database and sometimes multiple databases. Frameworks have also become a very common way of reusing components and getting applications developed quickly and uniformly.

The last two – support for parallel programming and remote development have less importance overall in this ranking because there are fewer developers who are actively engaged in either of these activities.

However, these rankings are what apply to the overall population worldwide. Within that expansive universe there are many subsegments which have different values based on the type of development being done, or type of environment. Evans Data has an additional data set developed for this tools survey, which is not included here, but which examines other features and cross relationships. In addition drill down by technology use, geography, industry, versions used, etc is also available as a service. If you would like additional information or more targeted drill down than is presented here please call us at 1-800-831-3080 or contact edcsales@evansdata.com.

Relative Importance of Attributes	Relative Ranking
Basic tools - Editor/ Debugger/ Compiler and/or Interpreter, etc	272.1
Documentation	212
Ability to integrate tools	195.5
Integration with databases	174.5
Support for frameworks	165.2
Database development tools	160.1
Size/quality of developer community	159
Web Design/Development Tools	154.7
Developer test tools including Code Coverage tools	154.7
Availability of 3rd party tools	149.9
Ready to use out of box experience	138.8
Sample apps	137.3
Visual tools and declarative support	136.9
Quality of Tech Support	136.4
Runtime memory analysis tools	135.4
App Modeling tools	108.3
Support for Parallel Programming	87
Support for Remote development	76.3

*User's Choice – Software Developer Platforms 2009
© Evans Data Corp 2009*

Attribute Relative Rankings

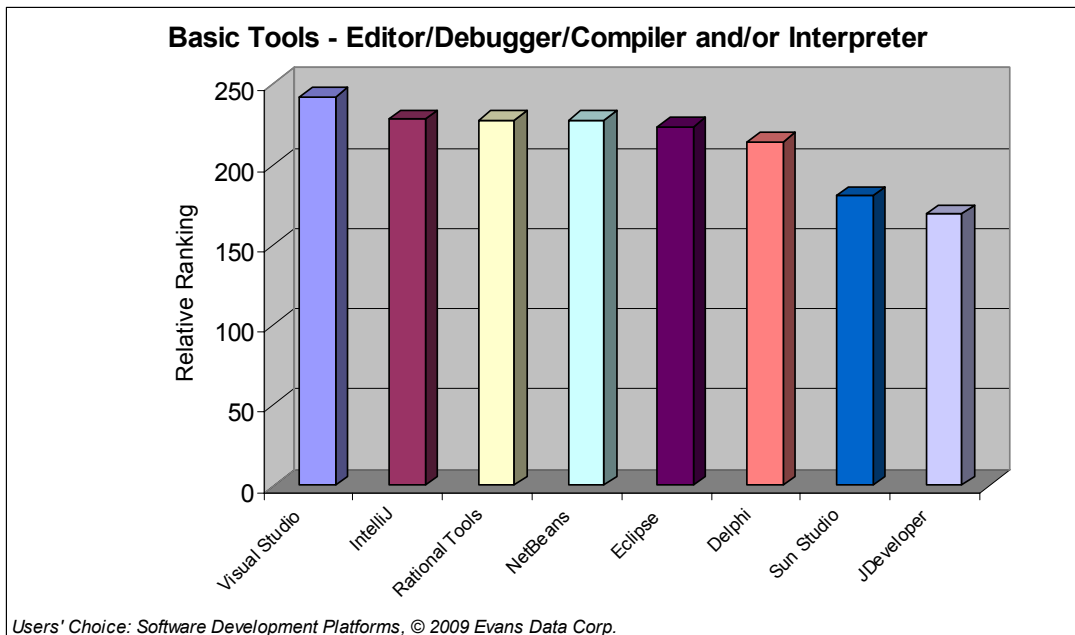
Basic Tools – Editor/ Debugger/ Compiler and/or interpreter

What’s more important in a car; the GPS system, the built-in hands-free phone, the Bose stereo, or the wheels? This is a similar situation for IDEs. Obviously we love all the bells and whistles, but the core of an IDE or toolset is the basic tools that it houses, namely the compiler, the editor, and the debugger. The developers rated these as the most important of the attributes we measured by a far margin because these are the features that are essential. Without them you can’t do anything.

“Obviously we love all the bells and whistles, but the core of an IDE or toolset is the basic tools that it houses...”

The original IDEs of the late eighties were revolutionary in bringing those essential tools together in a unified environment. Today’s IDEs are a lot more sophisticated with a wide range of tools supplied, and in some cases, such as the Rational tools suite and Oracle’s tools, they have grown “out of the box” to form whole infrastructures that encompass the software development lifecycle from initial requirements through architecting the app through development and testing to deployment and maintenance. Not all of those tools are a part of Rational Application Developer, but all can be connected and interoperate. On the other hand, Eclipse by itself really doesn’t provide tools, but just an IDE so developers who judge Eclipse on the basics may be using different barometers; i.e., they may be judging based on the plug-ins that are available and that they have used.

Visual Studio was rated highest in the basics. It is a good solid set of tools that Microsoft has been working on for some time and the quality shows. The next three were extremely close and should be considered a “tie” as there was hardly any difference in their scores.

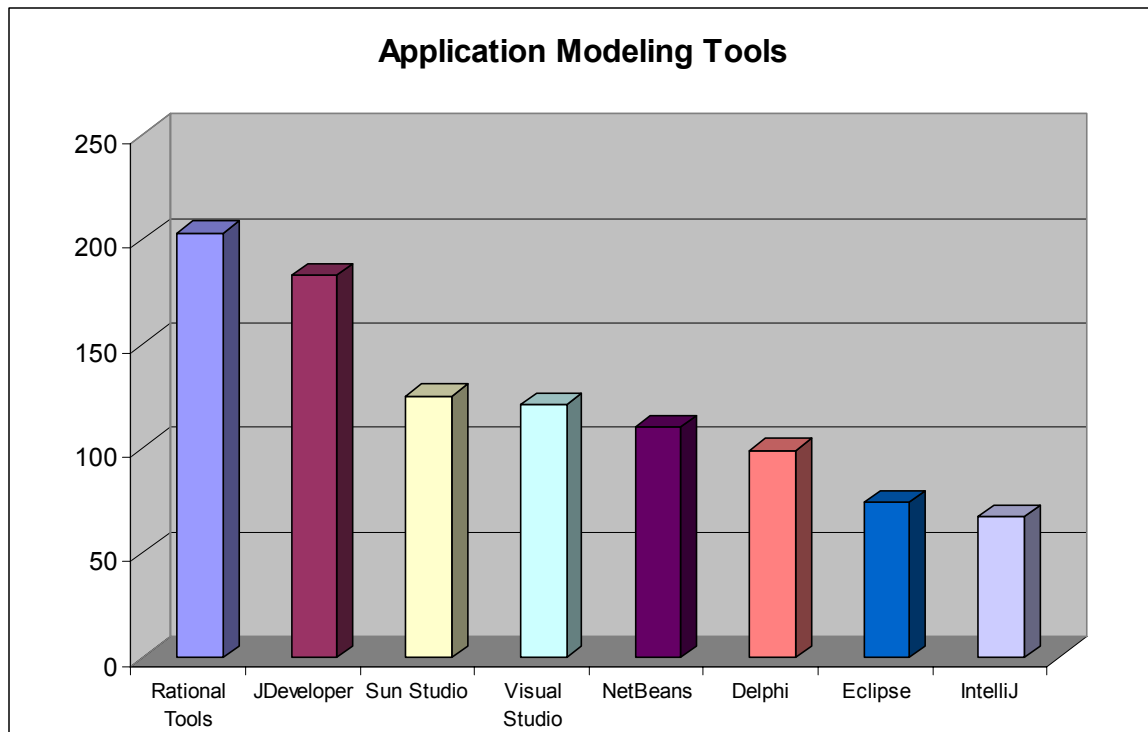


Application Modeling Tools

In 2006 when we asked developers to rate quality of application modeling tools, it was no surprise that the Rational Application Developer turned out on top with JDeveloper rated almost as high, and the next year they traded places. Last year the favorite was once again Rational and that's the same thing this year. The excellent application modeling tools that were at one time the cornerstone of the Rational product line before Rational was acquired by IBM have again been ranked more highly by their users than any other IDE and toolset and by a wide margin. Now they are in Rational Software Architect, a strong part of the Rational product line, and IBM has continued to evolve them into extremely powerful and easy to use modeling tools.

"...IBM has continued to evolve them into extremely powerful and easy to use modeling tools..."

Traditional modeling tools allow an architect to create a visual model of how a complex application will work, outlining components, calls, and other functions and thus providing means to optimize and provide stability to applications. Thus, application modeling tools are most useful in environments where large and complex applications are common. These are precisely the environments that Websphere and Oracle's flagship database product occupy, and so it is no wonder that these two companies have devoted time to perfecting their modeling tool capabilities.



Users' Choice: Software Development Platforms, © 2009 Evans Data Corp.

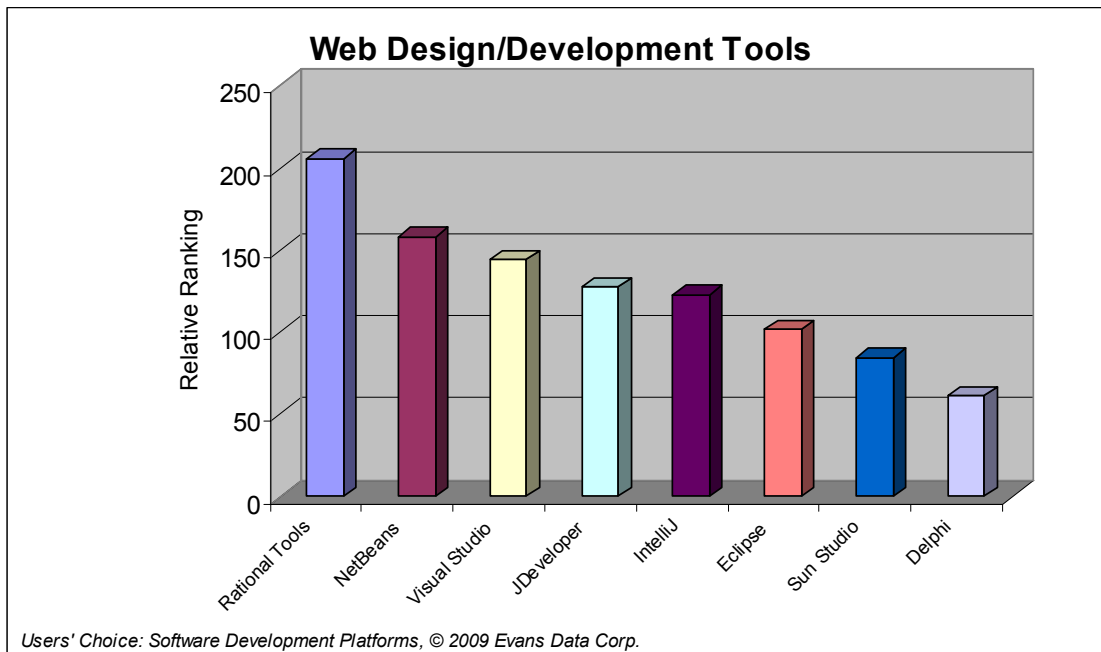
Web Design/Development Tools

Web Design tools are designed specifically for just that – designing websites, portals, and other applications that run over the Internet. They evolve as new advances make techniques possible that previously weren't and new technologies inspire new implementations. AJAX was such a technique that was highly popular a couple of years ago. Mashups on the web was similar. Now these haven't been replaced but are accompanied by other new techniques. As we begin to adopt development in the cloud as well as cloud deployment of applications, then even more specialized tools will be added to the mix to automate builds, provisioning and virtualization, and probably more.

"The tools help Java developers rapidly design, develop, assemble, test, profile and deploy high quality Java/J2EE™, Portal, Web/Web 2.0, Web services and SOA applications..."

IBM likes to keep at the forefront of technology trends, and Rational Application Developer for Websphere 7.5 contains lots of web development goodies. The tools help Java developers rapidly design, develop, assemble, test, profile and deploy high quality Java/J2EE™, Portal, Web/Web 2.0, Web services and SOA applications. IBM's software group has devoted a lot of time and effort to making web development easier, faster and more accessible to everyone. The result is obviously a well pleased group of users.

NetBeans has moved up considerably in the last year in this category and now comes in second. The NetBeans team has been working hard and we've seen NetBeans improve consistently through the last two years as reported by their users. Designed as a web development IDE and toolset, NetBeans has really come into its own in that respect.



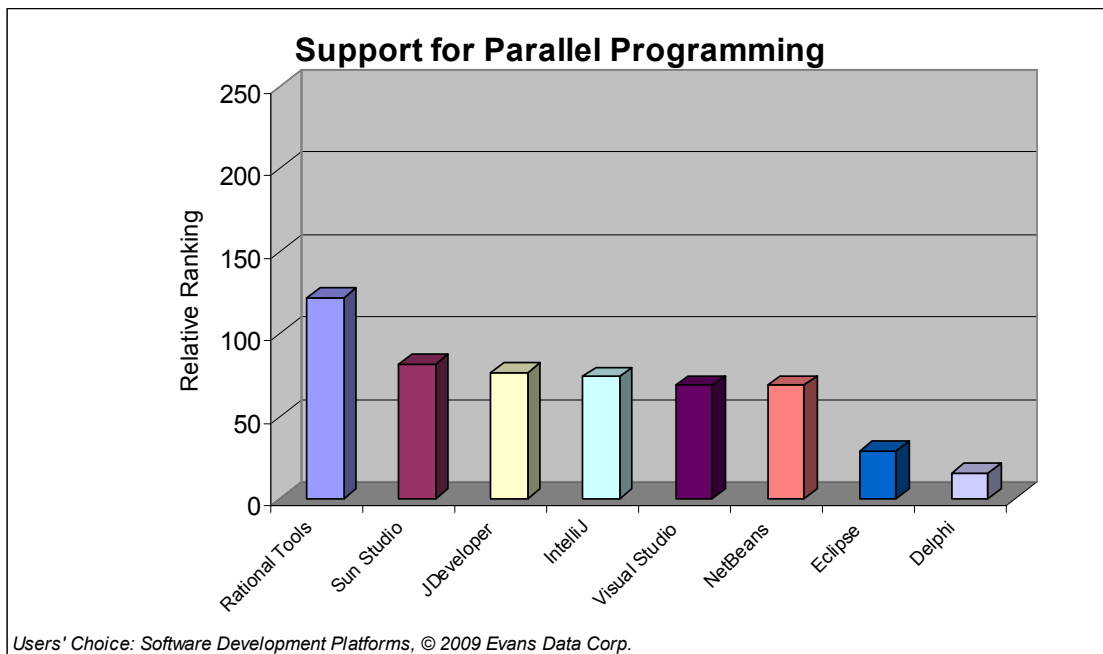
Support for Parallel Programming

“As multi-core processors become ubiquitous, the only way to truly exploit them is to explicitly design and develop multi-threaded applications that can take advantage of the performance and scaling capabilities that multi-core processors offer...”

Parallel programming is not yet one of the most important capabilities in the minds of most developers, but that’s about to change. As multi-core processors become ubiquitous, the only way to truly exploit them is to explicitly design and develop multi-threaded applications that can take advantage of the performance and scaling capabilities that multi-core processors offer. But multi-threading is complex and developers tend to want to avoid it. So how can we move to take advantage of multi-core processors? The answer is in tools and libraries specifically designed for the task.

Rational ranked highest of all the offerings in the survey, but notice that none of the products elicited marks that were close to those we saw in attributes concerning the core basics, or web development capabilities. That shows that parallel programming has still not yet come of age and that there is a need for this type of tools that hasn’t been addressed. Sun Studio is known for good parallel programming tools, which it has perfected in the Solaris environment, and this is reflected in their ranking here.

Shortly after this survey was conducted, Intel released Parallel Studio for Visual Studio. It’s an add-on to Visual Studio that lets users plug in a C/C++ compiler with pre-threaded libraries, plus a debugger with a memory and thread error checker, and a performance analyzer. While this isn’t exactly a part of Visual Studio, it may be that we’ll see some of the effects of Parallel Studio on the ranks Visual Studio gets next year.



Ability to integrate tools

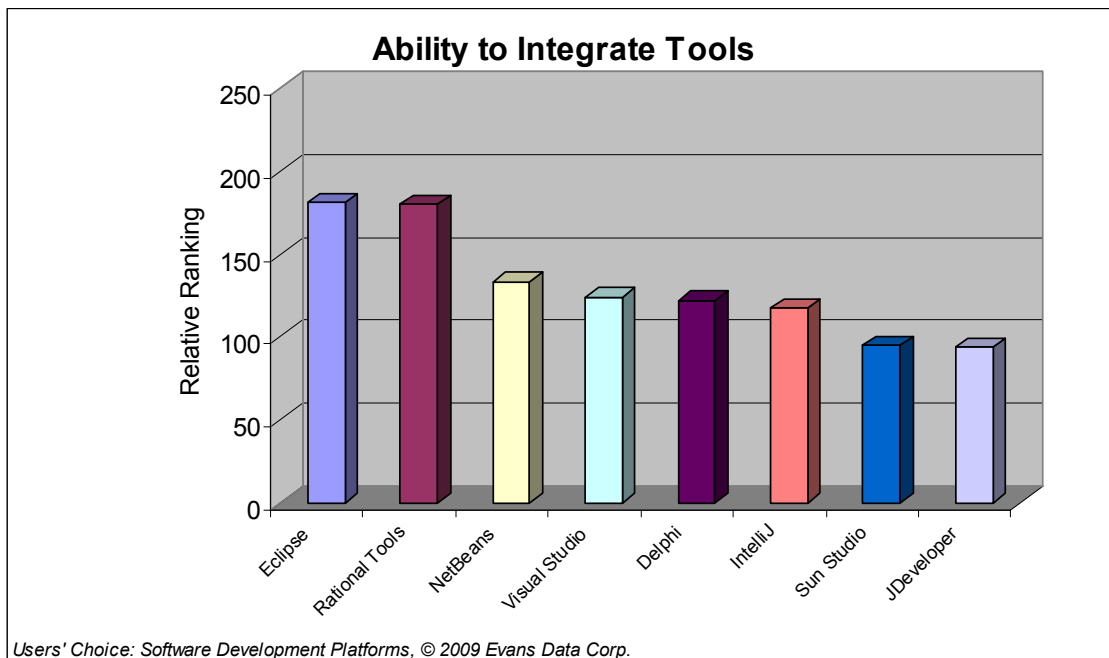
Before Eclipse, tool and IDE makers spent resources on developing partners who could and would create add-on tools, which could be integrated into various IDEs. Eclipse turned that model on its head by providing the core IDE and encouraging a community of developers to form who would create plug-ins of every conceivable variety as extensions to the core Eclipse IDE

Consequently, the importance of a developer community to create plug-ins and the ability of the IDE to integrate them has become much more important.

“There are literally hundreds of plug-ins for the Eclipse IDE and integration with external tools is Eclipse’s raison d’etre...”

It’s no surprise that Eclipse would be rated highest in its ability to integrate tools. There are literally hundreds of plug-ins for the Eclipse IDE and integration with external tools is Eclipse’s raison d’etre. But Rational’s tools are also supported by a vigorous community of plug-in vendors and have a rich assortment of partner tools that can be integrated. These can be found at its Rational Plug-in Central site where partners and other ISVs can have their products tested and certified for Rational compatibility and then listed for RAD users in one convenient location.

Delphi has lost considerable ground this year in this category. This is most likely due to a natural erosion of their community through turbulent years in the immediate past. As products pass from one company to another there is a tendency for confidence in the product to wane, and then the community begins to diminish. Hopefully Embarcadero will provide the stewardship to reverse this.



Availability of 3rd party tools

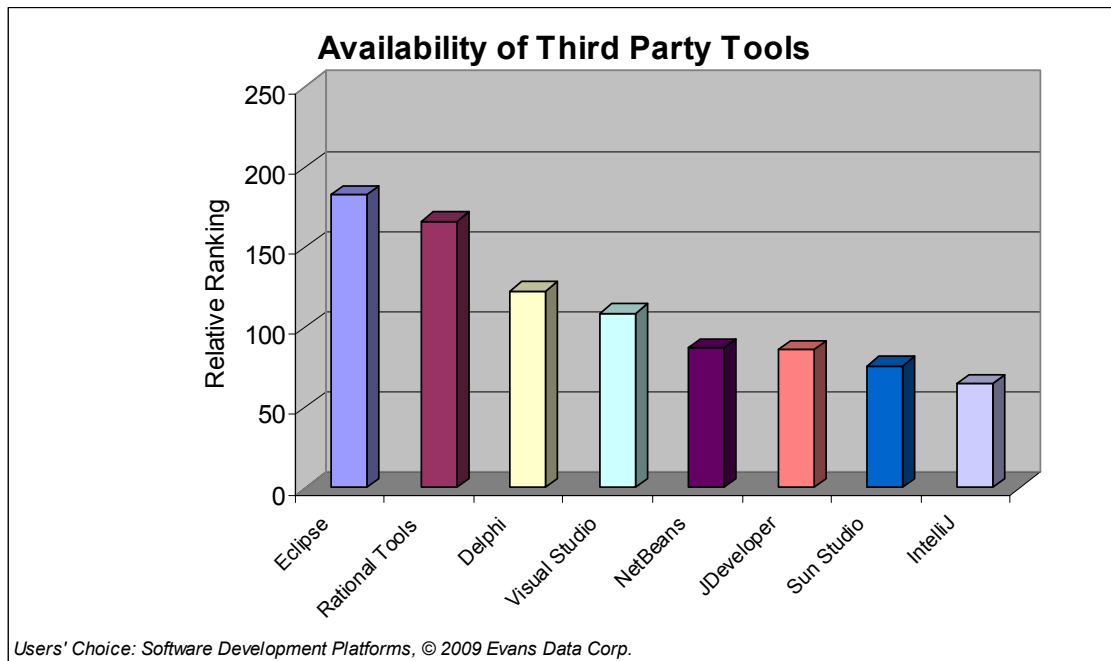
The availability of third party tools is largely related to the ability to integrate tools and the size and quality of the developer community. The larger the community and the more well fostered, the more third party tools will spring up.

Eclipse is a community as much as a technology and it's no surprise that Eclipse would take the lead in this category. As of this writing there were over 1200 Eclipse plug-ins listed at the Eclipse Plug-in central. There's most likely many more that aren't listed. In addition, there are a plethora of commercial tools available that are built on the Eclipse technology, including some of those in this report

Rational also has a Plug-in Central feature on the DeveloperWorks web site which lists a wide range of tools that have been tested and certified compatible with Rational offerings. When you consider the entire Jazz platform, and the number of partner applications that are available the variety is even more impressive.

Delphi has always had a large community with a lot of available add-on tools, and the Delphi developers still gave it good marks in this category. However, as we see more vividly in some of the other categories, Delphi needs reinvigoration.

"Eclipse is a community as much as a technology..."



Quality of Tech Support

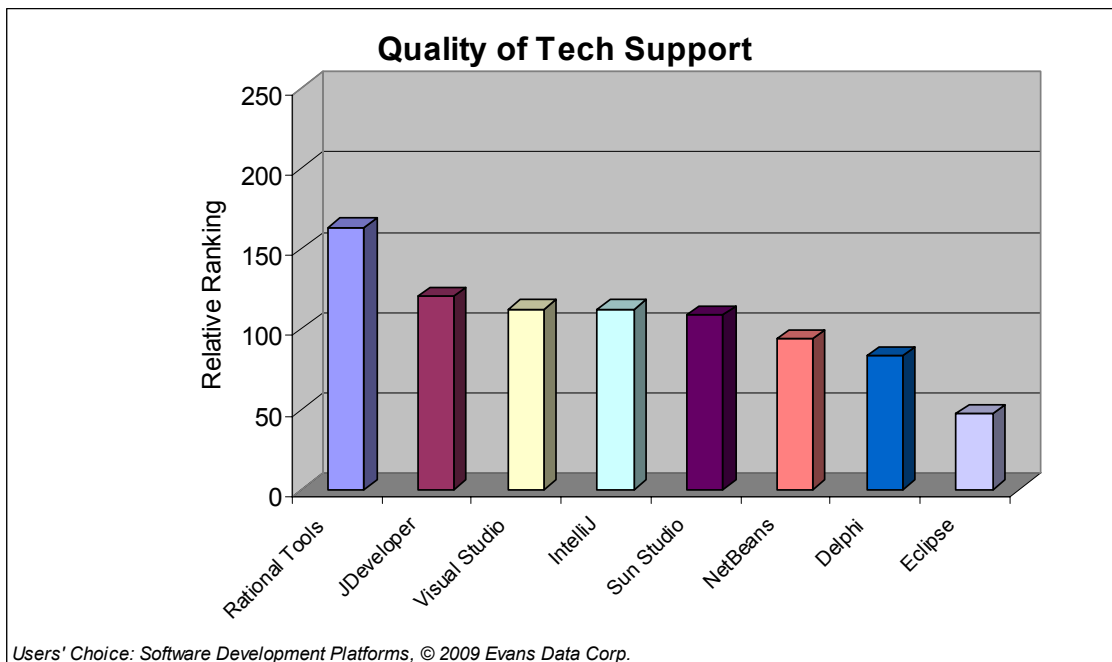
The first thing to notice about this chart is that, while there are some differences in the perception of users between the various platforms, all of them were ranked relatively poorly when compared to other attributes, with the sole exception of IBM. This is most likely due to the inherently unhappy nature of technical support. It is only needed when there's a problem and getting answers is often a long and frustrating process.

IBM's users were the happiest of the respondents by quite a long ways, followed by Oracle. IBM provides excellent 24 x 7 phone support for its tools product line through a front line support team who are experts in their fields. In addition, support teams work together across disciplines to solve problems, and IBM is known for providing support for multiple versions for years. IBM now offers a minimum of 5 years of purchasable support from the release date of a version/release with an optional purchase of an additional 3 years of support. This allows customers to maximize their software support investment and control their upgrade strategy.

Oracle was second last year and is second again this year, which is no small feat. Oracle has obviously been devoting time and effort towards putting in place programs that aid the developer and it shows. JDeveloper users rate their support highly.

On the other end of the scale, Eclipse is seen as lacking in viable tech support even though some companies have tried to make a living out of providing Eclipse support and implementation. Still, with Eclipse it's an uphill struggle to provide support even for a dedicated firm since there can be a staggering number of possible configurations a developer could use in an Eclipse model. The problem isn't one of lack of attention from the community, but the sheer difficulty of the task due to the myriad Eclipse configurations possible.

"...support teams work together across disciplines to solve problems, and IBM is known for providing support for multiple versions for years..."

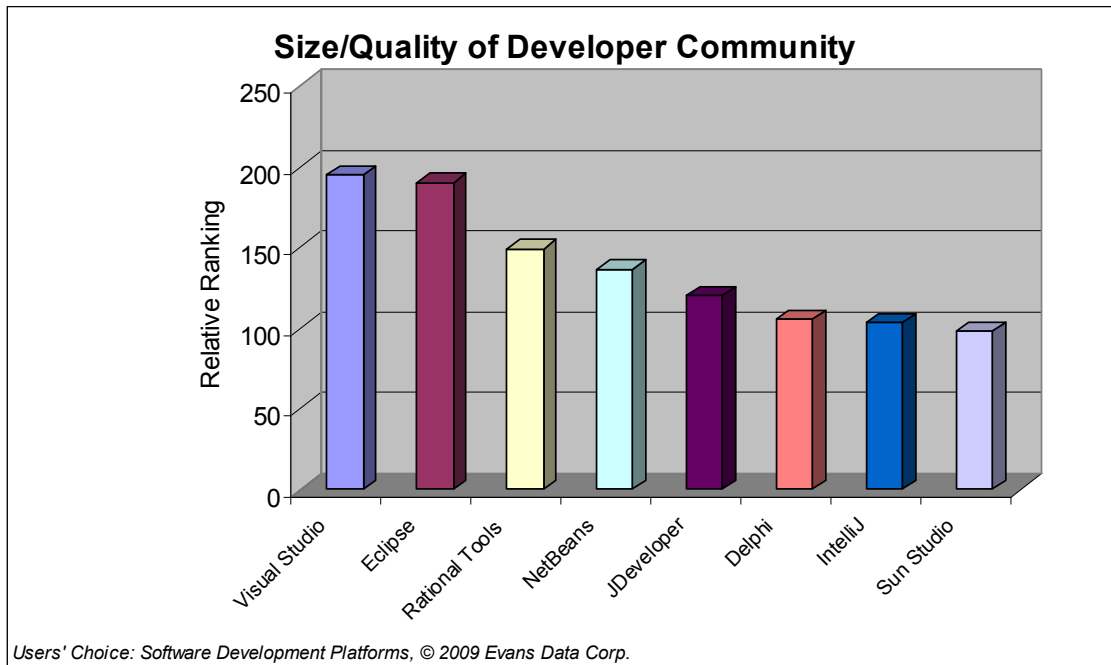


Size/quality of developer community

Visual Studio is the most widely used development platform and toolset in the world and it has been for many years. Because of the dominance of the Windows operating system and the technical optimizations for Windows and .Net in Visual Studio, this is the tool suite of choice for more than half the world's developers. From the early days of DOS, Microsoft understood the strategic value of getting developers to write to their operating system. They developed MSDN, the world's largest and most respected developer relations program, and with Windows they have dominated development for the last fifteen years. More developers write for Microsoft platforms than for any other and this means there is a huge and active developer community.

“More developers write for Microsoft platforms than for any other and this means there is a huge and active developer community...”

Eclipse, of course, is Visual Studio's closest rival when it comes to sheer number of developers using it. Our biannual Global Development Survey has shown the Eclipse community growing each year throughout the major regions until in some it is now closing in on Visual Studio. As we note elsewhere, Eclipse can be thought of as a community rather than just a set of plug-ins for an IDE stub, and that is apparent here.



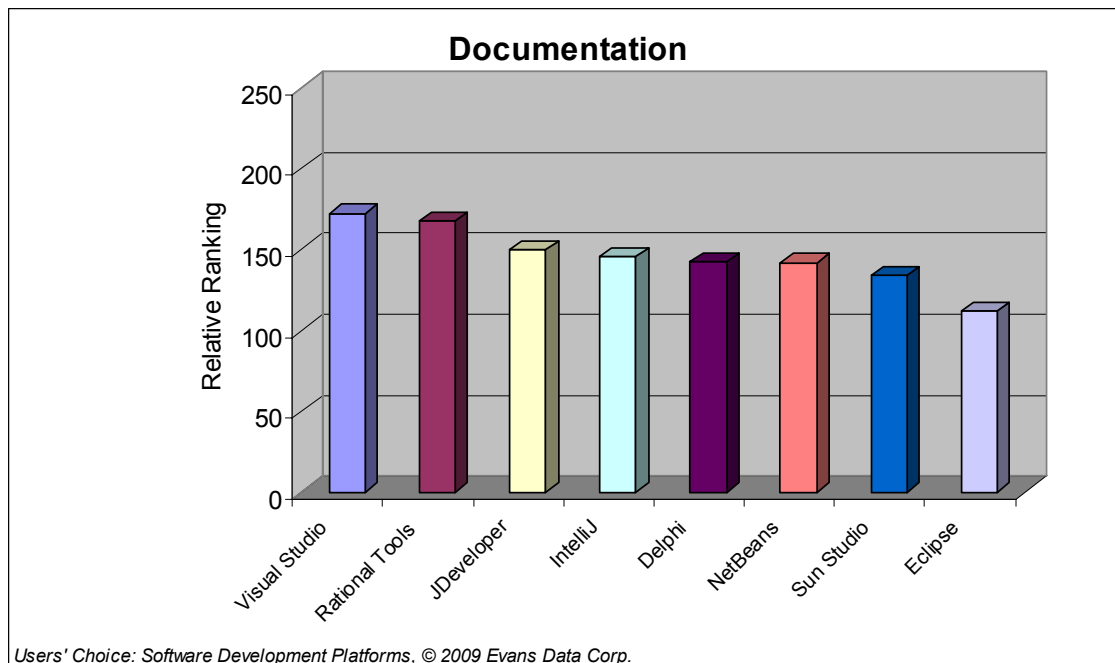
Documentation

Documentation is just one of the things that have been upended by the Open Source world. While there are plenty of developers who will spend long hours of their own time working on the code for an Open Source projects, there are far fewer who would like to spend any time creating documentation for a project or a plug-in.

Documentation, or the lack of it, has always been a favorite topic of complaint for development tools and for any other software product. Logically there is no reason why the documentation should always be such a problem, but it nonetheless usually is. The reason for this may be more human than technical. Everyone is exposed to the help system and everyone has an opinion about how things should be said and how they should be presented. With an extensive help system there is plenty of room to present information in a way that not everyone will like.

“Logically there is no reason why the documentation should always be such a problem...”

Note that none of the product offerings in this study were rated very high on documentation, although many of them were rated better than last year. All of the graphs in this study have been drawn with the same scale so that the overall difference can be seen. Note also that this was considered one of the most important categories by the surveyed users. This is an area where it might be profitable to spend some time.



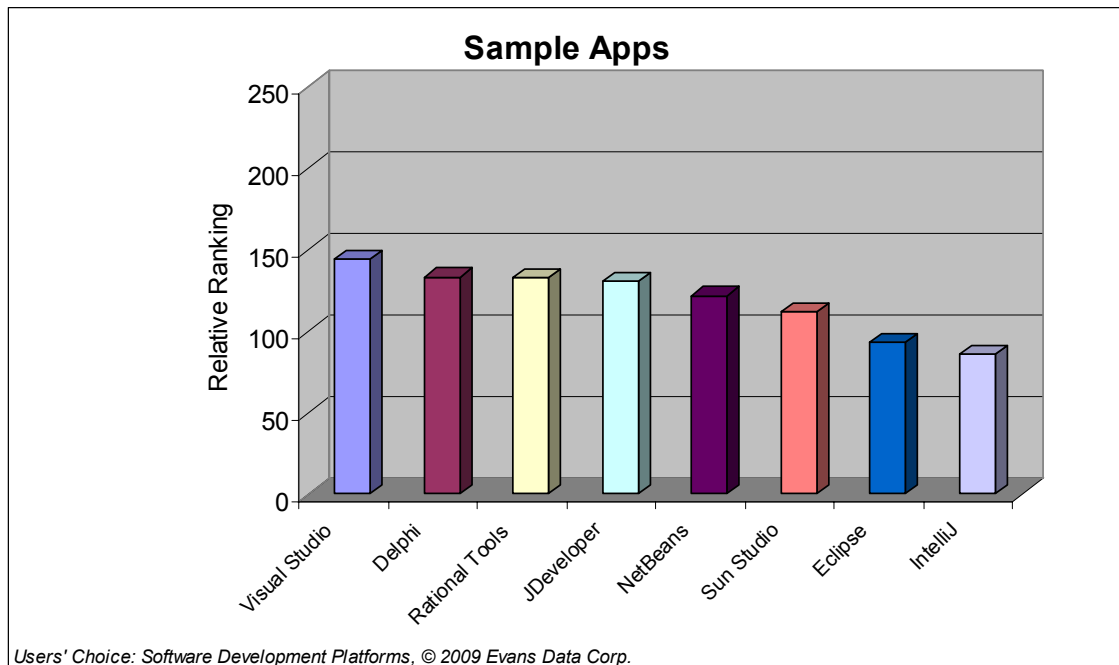
Sample apps

Some might say that sample applications are not a tool and not necessary for the operation of any development platform and therefore, they are not a valid feature for comparison. However, sample applications not only illustrate to the developer how a tool works, but they're also mostly reusable. Developers can take snippets or whole programs of sample application code and build them right into their own applications. Thus sample applications provide both education and some valuable shortcuts.

Sample applications do not ship only with commercial products. In the Open Source community there are plenty of applications with which source code can be found. The value of any particular sample applications will vary, of course, on how applicable it is the application a developer is writing.

These developers liked the sample applications for Visual Studio. They are many and varied and apparently Visual Studio developers find them to be quite useful.

"...sample applications provide both education and some valuable shortcuts..."



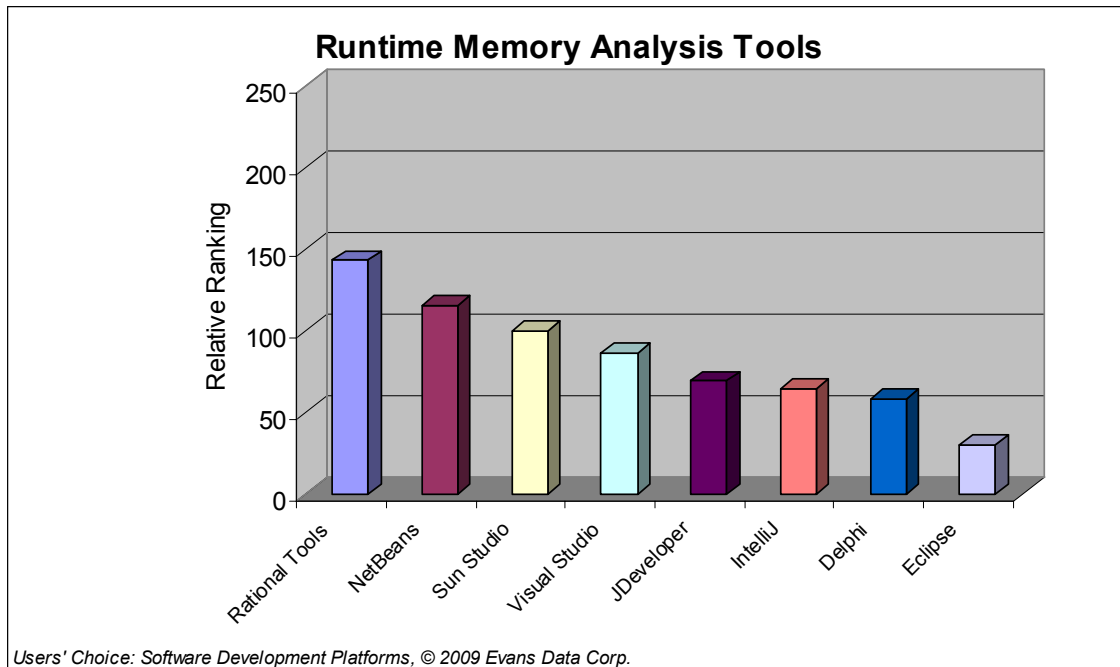
Runtime memory analysis tools

Rational's Purify Plus set the standard for runtime memory checking back in the mid-nineties when C/C++ applications were more common and memory checkers typically looked mostly for memory leaks. Today Purify Plus has evolved and can be found in its current incarnation in the Rational Quality Manager product. But IBM has not stopped with improving and evolving Purify Plus, it has also introduced through its Alphaworks program a multi-threaded Java memory analyzer. Multi-threaded apps are inherently harder to debug due to the fact that race or deadlock conditions may only exist under certain circumstances, and not all circumstances can adequately be tested in the lab. Thus a tool specially designed for analyzing threads and their interactions is extremely valuable.

"While most people may assume that the Java VM automatically releases memory, this is not true in all cases..."

NetBeans ships with a profiler that allows for memory analysis and leak detection. A memory leak is when a program fails to release memory after a specific function is accomplished. While most people may assume that the Java VM automatically releases memory, this is not true in all cases, hence the need for a runtime memory analysis tool such as the NetBeans profiler.

Note however that none of the offerings did particularly well on this capability, though it is one that will become more and more important. There's room for improvement everywhere.



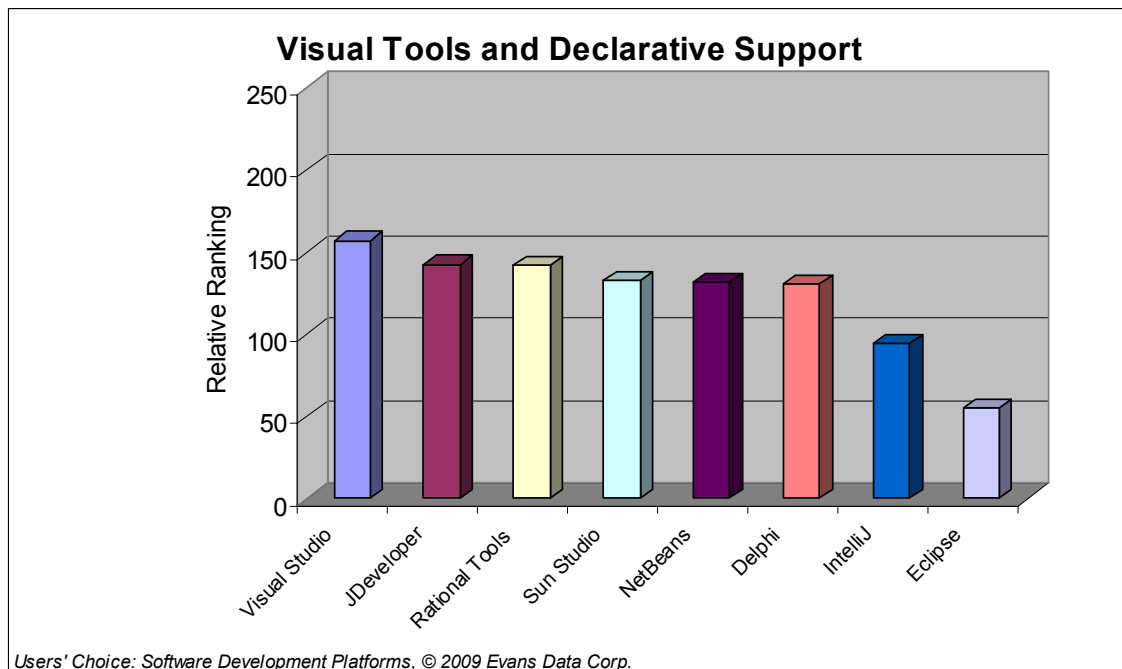
Visual tools and declarative support

Visual tools make the developer's life easier by relieving him of having to string together concepts and execution mentally, of having to write all the code and juggle all the pieces in his own memory, and they allow him to "see" relationships and flow in a meaningful way. Microsoft brought visual programming into the mainstream with Visual Basic, which promised to make programming so accessible that even untrained people would be able to write applications. It didn't turn out to be exactly that easy, but it was an evolution in programming that most others have followed since. Visual tools are much expected in any development platform and are very important in the initial architecture stage of the development.

"Microsoft has had strong visual tools since the early nineties. Also, they have recently announced that they are working on a new declarative language to make programming accessible."

Declarative support has been more recently debated. It essentially allows the developer to define what the result should be without specifying the exact "how" to arrive at that result.

Microsoft has had strong visual tools since the early nineties. Also, they have recently announced that they are working on a new declarative language to make programming accessible to non-programmers, specifically those who want to make web apps. Oracle JDeveloper's visual editors and page flows along with the declarative wizards for creating business services make the IDE and toolset much more declarative in nature compared with traditional Java tools. Eclipse, of course, is at a disadvantage since it has no way to control what the community creates and so can't react when a particular programming paradigm becomes popular.



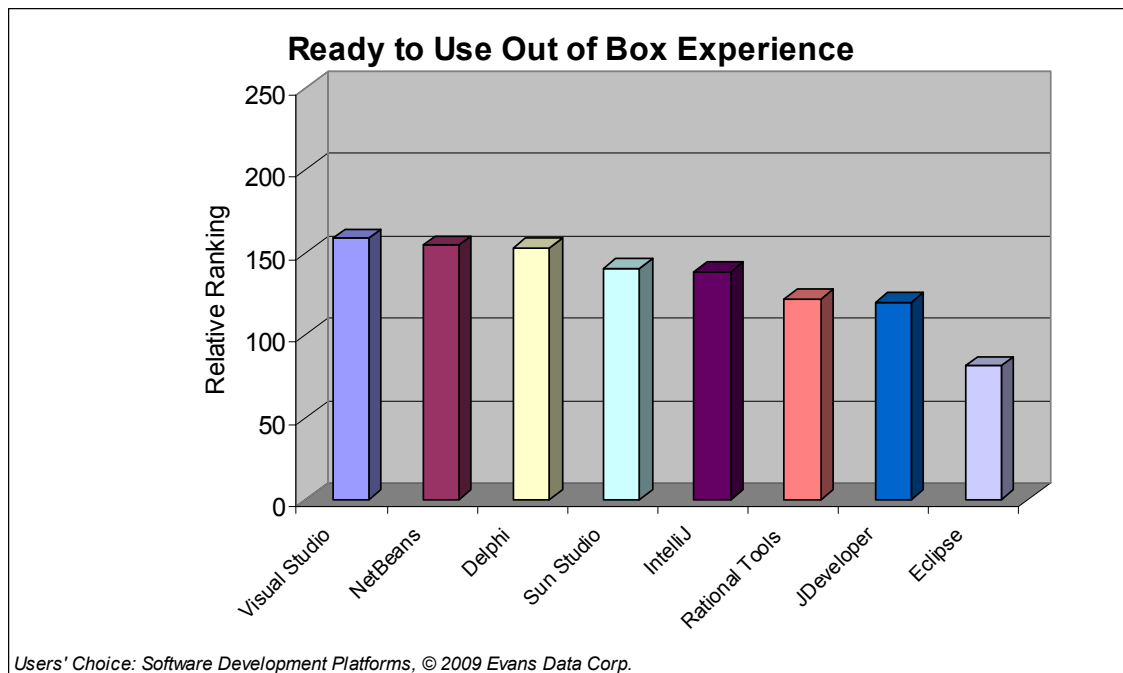
Ready to use out of box experience

This category is kind of unusual in that many of the offerings in this study were rated as having a high score in being ready to use right out of the box relative to other attributes they were rated on. But then when the scores were compared to the scores achieved by the other IDEs that perception no longer stands. The graph here, of course, shows the rankings on this subject relative to the competition.

Visual Studio had the best ranking for out of box experience, but the scores of the top three have very little difference. NetBeans, Delphi and Visual Studio all had scores close enough to be considered a virtual tie. Rational tools and JDeveloper have both done a lot to make their user experience as ready to go out of the box as possible, but both are large and complex tool suites optimized for large environments and sporting a lot of functionality. Such tool suites are very hard to make into a ready to go out of the box experience.

Eclipse, on the other hand suffers from the opposite problem. As an IDE "stub" and a world full of plug-ins, Eclipse configurations and installations can be quite a chore both in selecting the desired plug-ins, integrating them together, configuring and installing.

"Eclipse configurations and installations can be quite a chore both in selecting the desired plug-ins, integrating them together, configuring and installing."



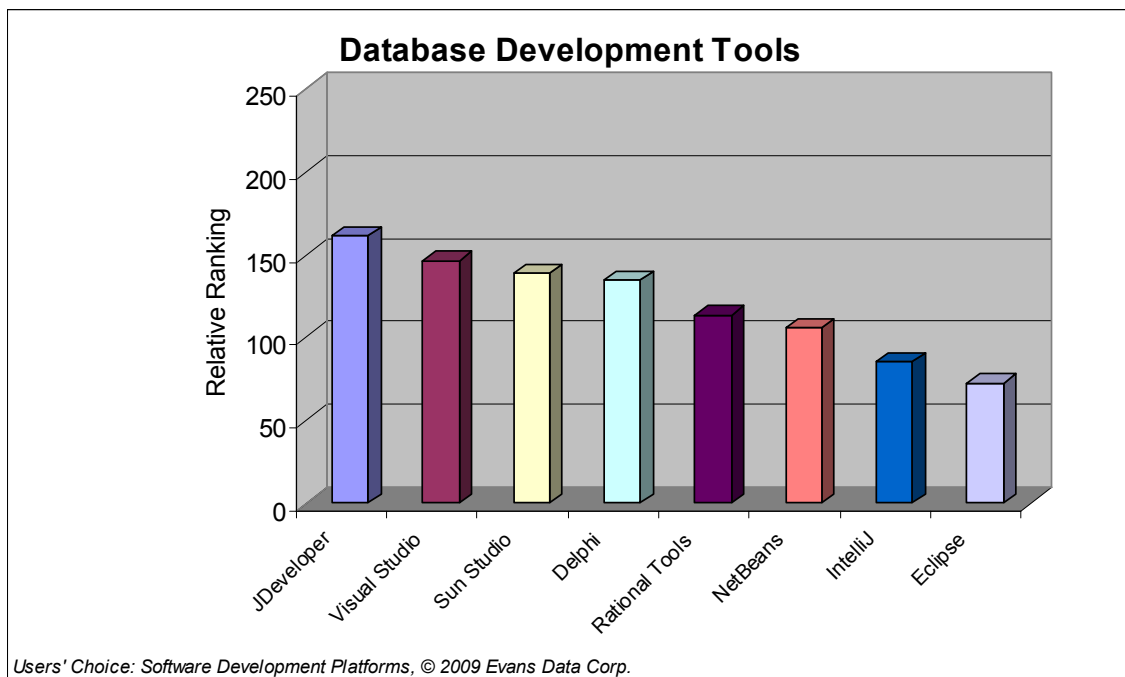
Database development tools

Databases are an integral part of almost every application and so the ease with which the developer can access and interact with the database becomes a very important consideration in the commitment to a development platform. Tools that enable the developer to create, customize and interact with the database are thus very important.

“Oracle offers a complete and integrated set of tools for application development, database development, and business intelligence and the developers who use Oracle’s JDeveloper obviously think very highly of them.”

It’s hardly surprising that Oracle comes out on top in this challenge. Oracle offers a complete and integrated set of tools for application development, database development, and business intelligence and the developers who use Oracle’s JDeveloper obviously think very highly of them. Of course they give full support to the Oracle database, but also have tools and connectivity for other databases.

Visual Studio also has good tools for database development to support its SQL Server database and its users gave it fairly good marks in this respect. However, note that most of the scores in this category are relatively low in comparison to several other categories.



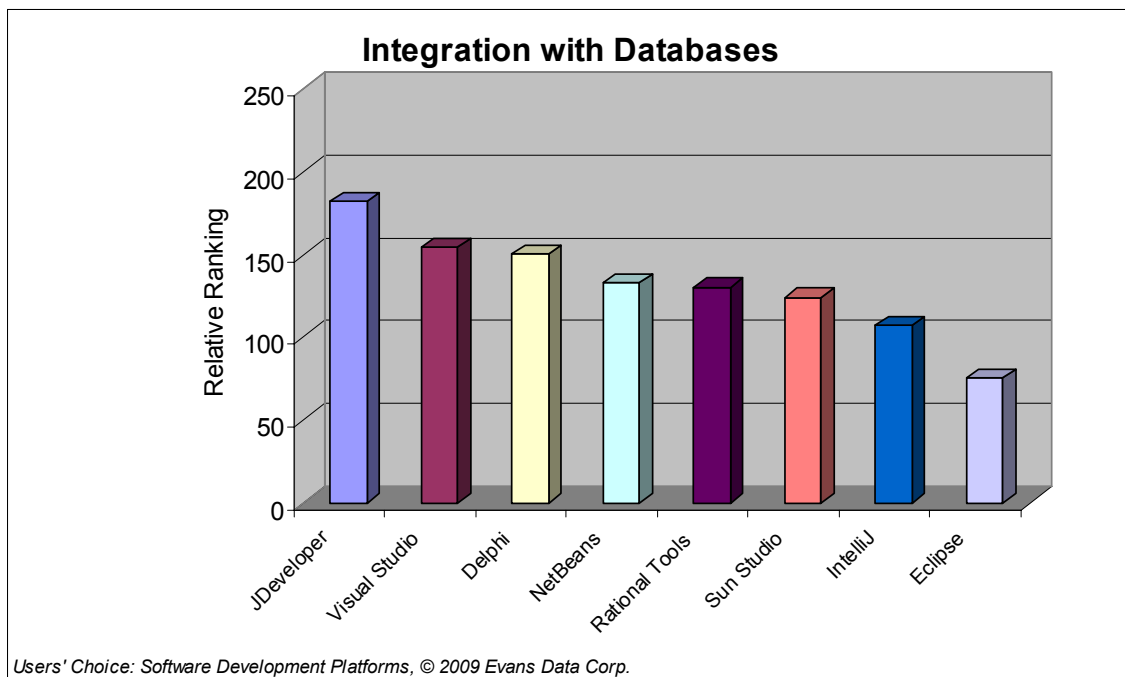
Integration with Databases

Database integration implies the ability to access relational and non-relational data sources in back end systems such as Oracle, DB2, and SQLServer, from a common framework using one tool or set of tools. Access to the data is usually in the form of a query, but can also include update capabilities, and can happen in bulk, real-time, or near real-time. As we noted in the subject of database development tools, this is a very important consideration as most programs do have a database and often more than one.

Oracle comes out on top again in this category, which is hardly surprising considering their success as a database vendor. Oracle JDeveloper enables full integration with the Oracle database but also with others including SQL Server. Now that Oracle has acquired Sun and MySQL we can expect JDeveloper to take an even more commanding lead in this category as it seeks to completely dominate the database landscape from the low end open source MySQL, all the way up to the high end traditional Oracle database.

The users of both Visual Studio and Delphi were less enthusiastic than the JDeveloper users, but they were still strong in their appreciation of the database integration of these two offerings.

“Oracle JDeveloper enables full integration with the Oracle database but also with others including SQL Server...”



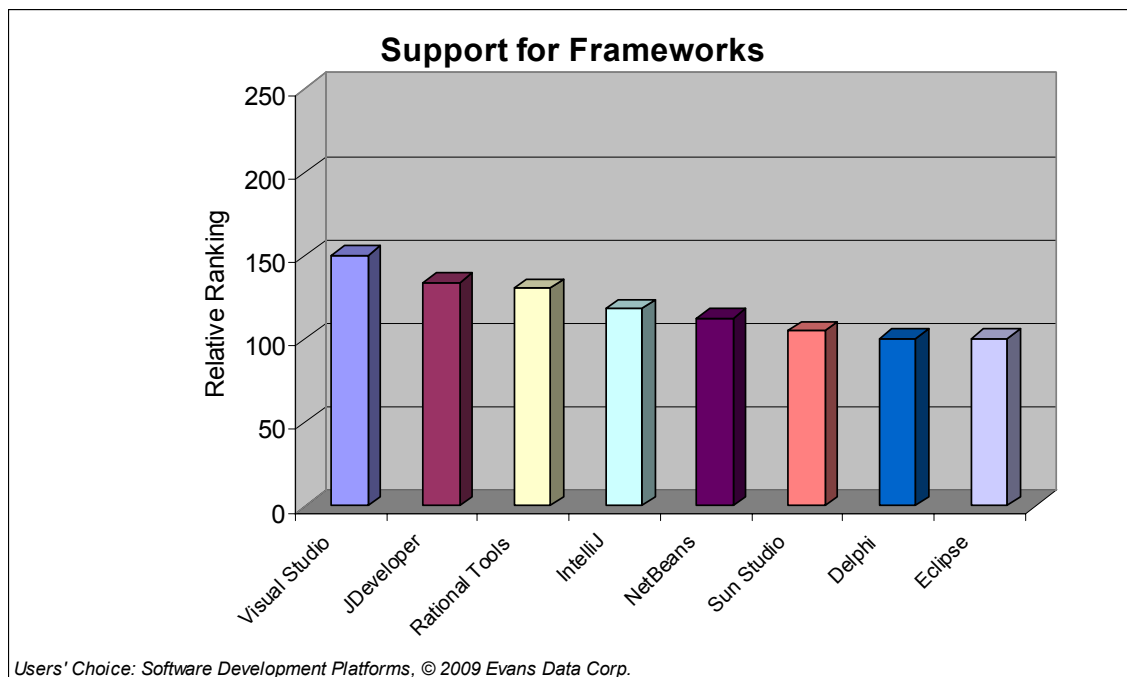
Support for frameworks

Frameworks make life a lot easier and less tedious for the developer by supplying standards components that can be used instead of recreated, such as windows, menus, and many web development frameworks provide libraries for database access, templating frameworks and session management. The grandfather of frameworks was the Microsoft Foundation Classes introduced in the early nineties as a way for Windows developers to skip recreating many parts of the GUI. Today they are more commonly thought of in the world of web development and have become more focused in that direction. Still Microsoft users clearly approve of the frameworks that Microsoft supply, as this survey shows.

“Microsoft users clearly approve of the frameworks that Microsoft supply, as this survey shows...”

JDeveloper and the Rational product suite also were well thought of by their users in the realm of frameworks. The two were so close together as to be virtually tied. Oracle Application Development Framework is a complete end to end enterprise framework that provides infrastructure and declarative support. IBM supports a number of different frameworks including Java frameworks and the Cognos framework.

Note that the remainder of the offerings were relatively weak and scores overall were down in this category.

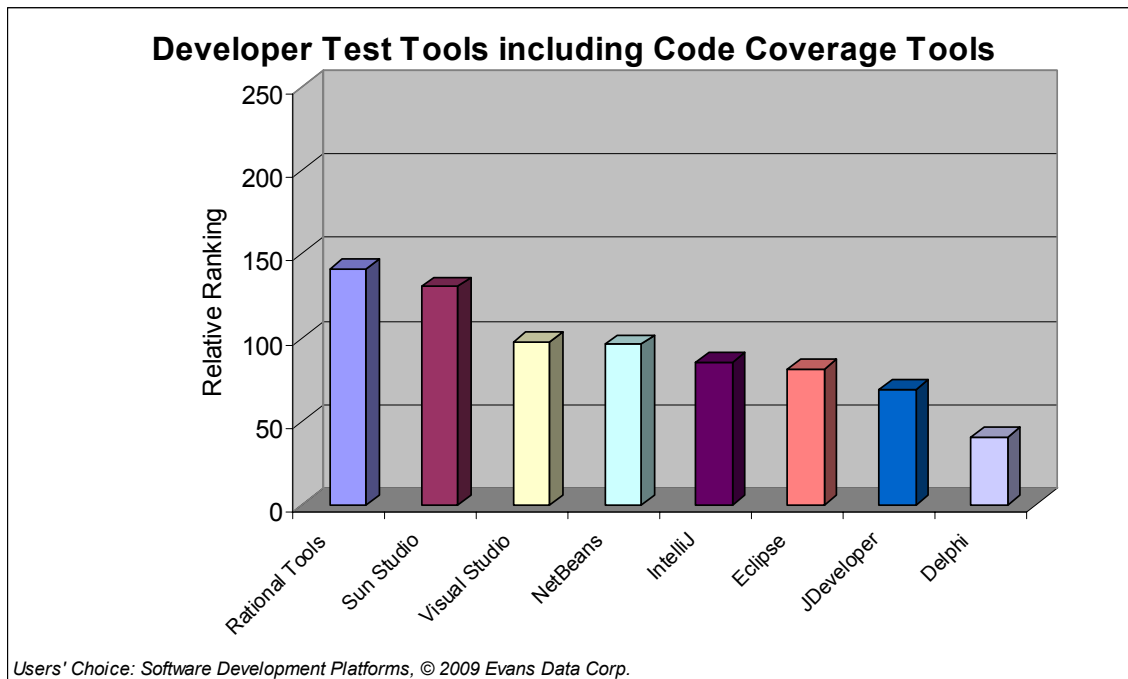


Developer test tools including Code Coverage tools

“...when we think of the development process as an end to end platform extending from requirements through development to deployment and beyond, test tools are definitely a big part of the process.”

Testing tools are sometimes neglected by product people putting together toolsets for developers. That’s because most large companies or ISVs have two distinctly different teams that do development or QA. Developers typically cost more and have more advanced skills than QA engineers, so it doesn’t usually make much sense to have developers test their own code. In addition, it’s always harder to find errors in one’s own work than in someone else’s since humans tend to unconsciously compensate – thus a separate QA team usually means more reliable testing. However, when we think of the development process as an end to end platform extending from requirements through development to deployment and beyond, test tools are definitely a big part of the process.

IBM’s Rational Quality Manager provides the first line of defense against bugs in the Rational suite of tools, but they also offer performance analyzers and are integrating the Quality Manager with their DOORS requirements management product. Sun Studio 12 offers a complement of QA tools including tools for finding and identifying threading problems.

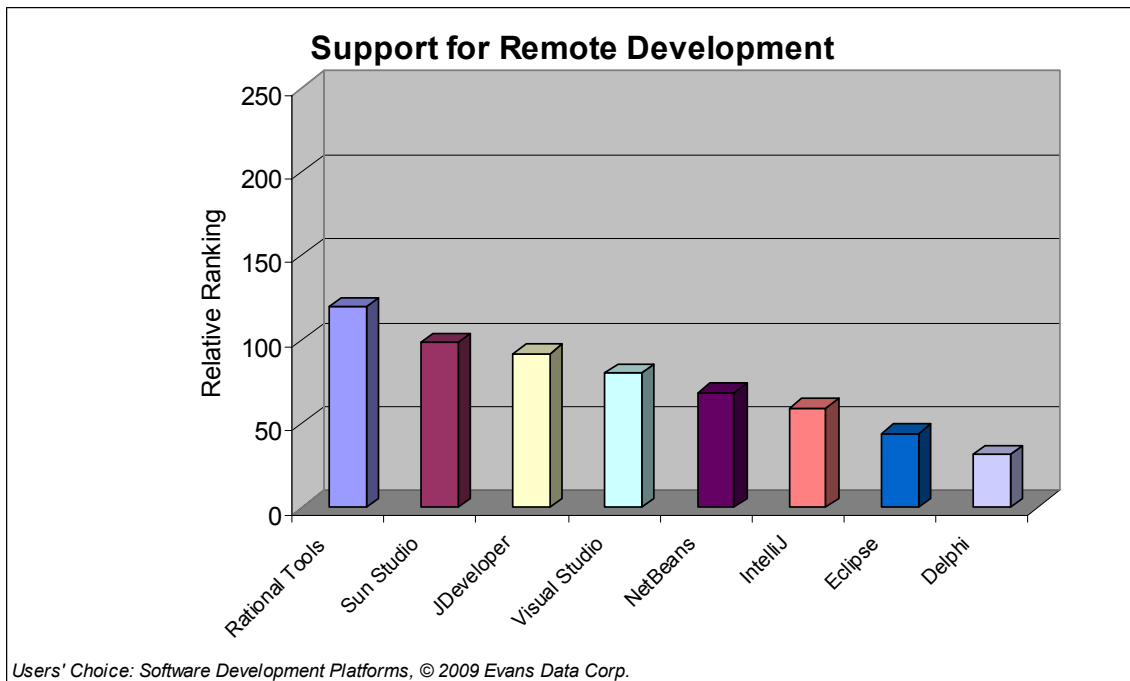


Support for Remote development

“Issues that come up often have to do with communications between team members, scheduling and division of tasks along with integrating finished components as they are completed, and managing version control.”

Remote development has become increasingly important as teams become geographically separated, often by several time zones. It’s not usual for a large program to be worked on by several teams and even if they are in the same city they still need tools that will help them coordinate their activities, check code in and out, and manage repositories. Issues that come up often have to do with communications between team members, scheduling and division of tasks along with integrating finished components as they are completed, and managing version control. In some cases team members of geographically distributed teams may need to actually work closely together on a particular design or component.

Rational Team Concert is the tool that enables developers working on the Rational Jazz platform to interact with each other and coordinate their efforts. It’s a very visual tool in the suite and simplifies communications, collaboration and distributed development. Sun Studio Express provides remote development capabilities for the Sun Studio products.



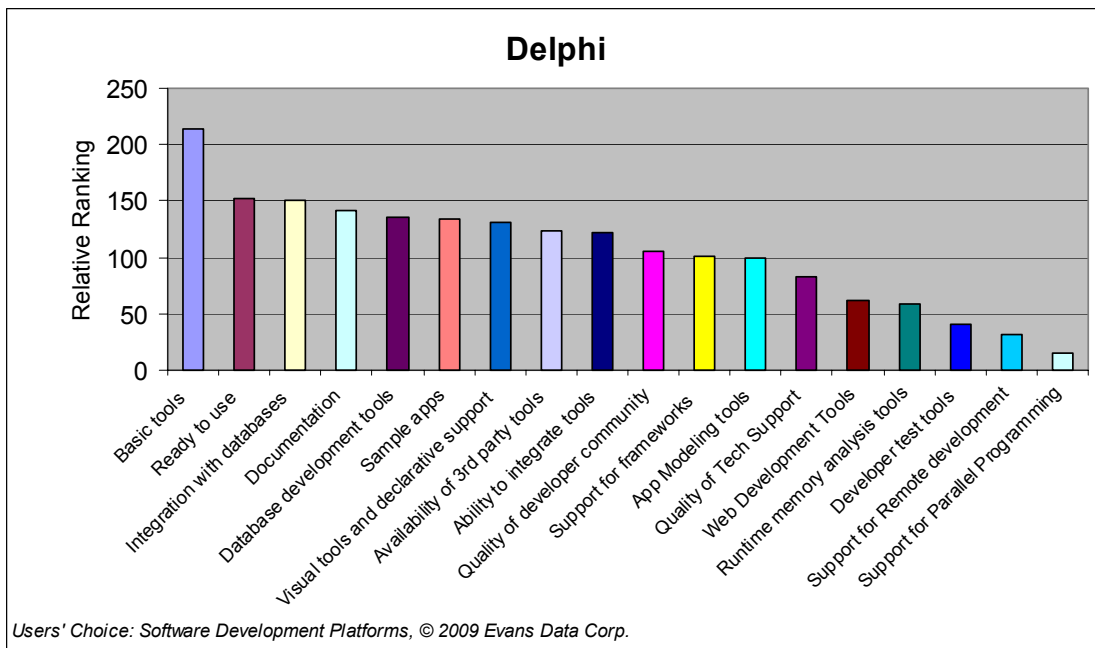
Overall Ratings by Development Platform

Delphi

Delphi has a long and illustrious past, starting in the mid-1990's when Anders Heilsberg and his team at Borland created the product as an evolution of the Turbo Pascal compiler that was Borland's first product. This was during the time when Visual Basic and PowerBuilder were first introducing visual tools and easing development and the barriers to programming. Since then it has seen the highs and lows of its creator company and has passed from belonging to Borland to belonging to Codegear and now to being an Embarcadero tool suite. With all that moving it's hard to keep the confidence of users, and while Delphi did well as late as last year, the reduced user confidence has started to show. User surveys are measurements of perceptions and perceptions are often effected by current events. It may be that as Embarcadero takes over full stewardship of Delphi and continues to make the product evolve that users will feel more enthusiastic again. We will see next year.

"Users still rank Delphi's core tools quite highly..."

Users still rank Delphi's core tools quite highly, as we can see from the table below which shows the rankings Delphi users gave to it without respect to the other toolsets n this survey. Most of the other values however are relatively low



compared to the basic tools.

Delphi was created to marry the high performance and object-oriented Turbo Pascal compiler with visual tools that let developers drag and drop components, "draw" windows, and so on. It was one of the first RAD (Rapid Application development) tools and ships with the Visual Component Library (VCL), which

is a collection of re-usable component objects that can be incorporated into applications.

Delphi sports an enterprise framework, a reusable visual component library, and a fully integrated suite of modeling tools. Its multi-language developer environment for Microsoft Windows supports Delphi (both Windows and .NET), as well as C and C++ applications within a single environment.

The current version, Delphi 2009, includes new and enhanced VCL controls, new Unicode support throughout the IDE and VCL, the addition of generics and anonymous methods to the Delphi language, a DataSnap feature that separates data and business logic, as well as the Delphi Architect to help with application modeling and visual representations of data structures.

The Delphi community is over 15 years old and is extensive with over 500 tech partners. Delphi supports its community through forums, newsgroups and peer to peer support.

Pricing is set at \$899 per seat for the new user and \$399 for an upgrade to the latest Pro version from a prior version. The Enterprise version is priced at \$1,999 per seat with an upgrade price of \$1,299. An ongoing subscription is also available and includes both minor and major upgrades and three tech support incidents with support available via phone and online.

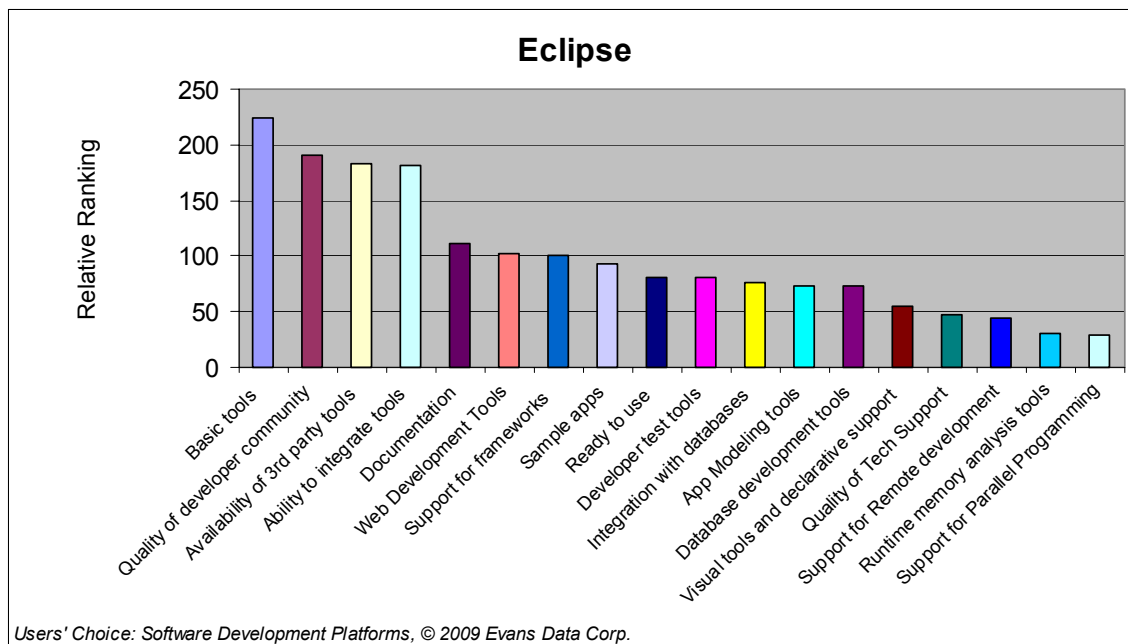
Eclipse

“Eclipse is one of the most important technologies of our time, not just technically, but because of how it has reshaped the development tools landscape.”

Eclipse is one of the most important technologies of our time, not just technically, but because of how it has reshaped the development tools landscape. Yet it is not the same type of product that the others in this study are, but to keep it out of the survey as we did last year would be to cut out one of the most used development IDEs and platforms, even though it may not be exactly comparable to the others. Indeed, we got several complaints from people who thought it was unfair to leave it out – and so here it is again.

The Eclipse Foundation is a not for profit organization with two missions – the first is to build a platform that vendors can use in their commercial products, and the second is to create an environment in which the community can build a strong tools ecosystem around that platform. In both of these missions, Eclipse has done extremely well.

Eclipse began in 2001 as a research project by IBM to create an Open Source platform for building IDEs and tools. The goal was to make a platform framework and exemplary applications. The first tool was the Java IDE that has taken the developer world by storm.



In 2004, the Eclipse Foundation was formed as a not-for-profit organization and the realm of Eclipse solidified around that organization. Today there are over 180 members in the Eclipse Foundation and there is a catalog of over 1000 plug-ins for the Eclipse platform, though many insiders in the Eclipse organization think that number represents only a small portion

of the plug-ins that are actually available. Eclipse is a phenomenon. It is wildly popular among Java developers and Open Source developers alike, and now that it supports languages other than Java, including C++ and now open scripting languages like PHP, we expect its popularity to swell even faster. Eclipse has been called the “killer app” of OSS development tools and its market share is increasing constantly. No doubt one of the reasons for its huge success is its business model as a core for plug-ins. These plug-ins are what has taken the industry by storm as more and more ISVs and OSS developers create and extend Eclipse extensions and plug-ins.

Eclipse’s strengths are related to the quality and size of the development community as well as to a great availability of third-party tools and an innate ability to integrate them. These are the facets of the original design as well as functions of the supporting community.

The target market is extremely broad – as broad as the whole Open Source movement since tools are requisite to any software project. The Eclipse Foundation is shepherding over 90 open source projects, and there are over 1000 plug-ins available from Plug-in Central, the main distribution site for Eclipse plug-ins.

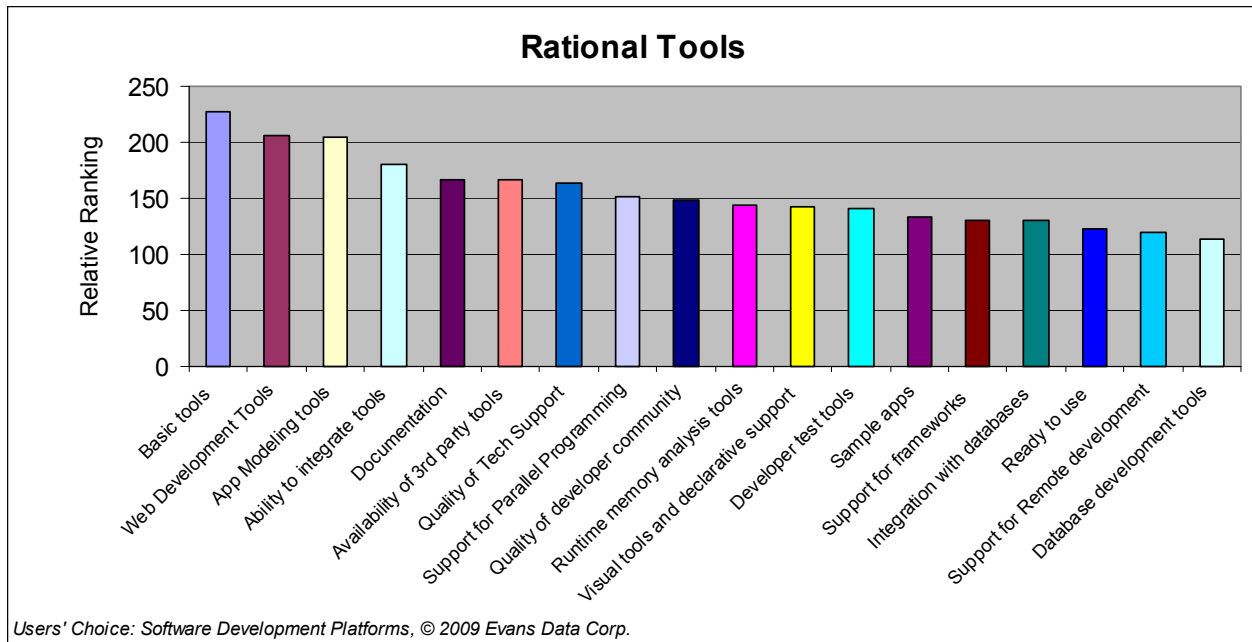
Eclipse is open source and so is available free of charge. Tech support is provided by the community and commercial vendors.

Rational Tools

“IBM thinks of software development as a platform that should seamlessly integrate across all activities from conception through testing and beyond.”

IBM’s Rational Tools Suite is an extensive and powerful line of products that integrate into the underlying Jazz framework and enhances development from the initial requirements phase through architecting and development, testing and deployment, all the way through to requirements change and maintenance. Its use stretches from complex high end Enterprise applications all the way to embedded systems implementations. It is impressive for its quality and the comprehensive nature of the component offerings, and users clearly appreciate it.

Rational scored better than any of the other development platforms in user satisfaction survey for the third year in the four that we have been conducting this study, and



the reason is a strategic targeted market focus into the needs of the development community and a devotion to excellence in execution that has produced a superior platform as well as an entire family of integrated development products that runs from requirements management to project conception to development and implementation and beyond. While in the past we have constrained the survey to only features that might be found in an IDE, this year we expanded the questions out to be more inclusive of the entire software delivery system – a survey strategy that better fits comprehensive offerings from most of the vendors, most notably IBM, Oracle and Microsoft. Still, even so the Rational tools users clearly were most satisfied.

Rational Application Developer (the IDE) is an integral part of the overall Rational product portfolio, which provides

customers with an integrated solution for their end-to-end development lifecycle. This means being able to have traceability relationships of requirements to analysis and design elements, model Service Oriented Architectures with Rational Software Architect, share those assets with team members with Rational Asset Manager, and implement and test those services collaboratively with globally dispersed teams using Rational Team Concert. All of this while storing and managing work with ClearCase and ClearQuest.

IBM thinks of software development as a platform that should seamlessly integrate across all activities from conception through testing and beyond, and Rational Application Developer represents the core construction part of IBM's software delivery platform. When combined with robust modeling capabilities such as those found in Rational Software Architect, the right solution can be offered for the right role spanning across software analysts, architects, and developers.

Having an integrated solution delivery capability is part of a larger and even more valuable solution for customers. When expanded to include business requirements capture, integrated business process analysis and modeling, and Enterprise Architecture, the Rational tool suite becomes an important element of an overall Architecture Management discipline.

Rational's users were impressed this time, just like every time before, by the application modeling tools and capabilities as we can see by the competitive scores. This draws from the venerable history of the Rational Rose UML modeling tools, now found in the rational Software Architect product. But users were also captivated by the web development capabilities that the Rational family provides. Rational's web tools include a Web Diagram Editor with visual application flows that supports Struts and JSF applications, a page designer to visually layout pages, a site designer, and AJAX support with a function rich JSF library. . RAD's WYSIWYG Page Designer incorporates a palette of web components (JSF, HTML tags, Form tags, and JSP tags). Portlet and portal pages can be quickly built using the portal page designer. A developer can quickly build the web UI visually and then associate it to a back end java bean, EJB, database table or Web service. Rational supports the creation of asynchronous, reliable, secure web services from Java beans or WSDL/WSIL files, and the ability to define Policy sets for Quality of Service. Rational helps developers build applications or fully develop the application right down to generating client code for the application and deploying it to the target runtime.

The Web development section of the developerWorks web site illustrates how far IBM has gone to provide easy high quality web development tools based on Java, with support for other languages as well - such as PHP. The developerWorks Rational zone same site shows off the availability of third party tools and their easy integration through their Plug-in Central

site offshoot that features a wide variety of third party tools that are certified compatible with the Rational suite and the underlying Jazz platform. This was another strong point for Rational.

The Rational team also got the best score for quality of technical support – not exactly a toolset feature, but certainly an important consideration in the purchase of any piece of software, and especially important for development tools, which require a higher level of expertise than ordinary software.

IBM provides excellent 24 x 7 phone support for RAD through a front line support team who are experts in their fields. In addition, support teams work together across disciplines to solve problems, and IBM is known for providing support for multiple versions for years. Additionally, IBM now offers a minimum of 5 years of purchasable support from the release date of a version/release with an optional purchase of an additional 3 years of support. This allows customers to maximize their software support investment and control their upgrade strategy.

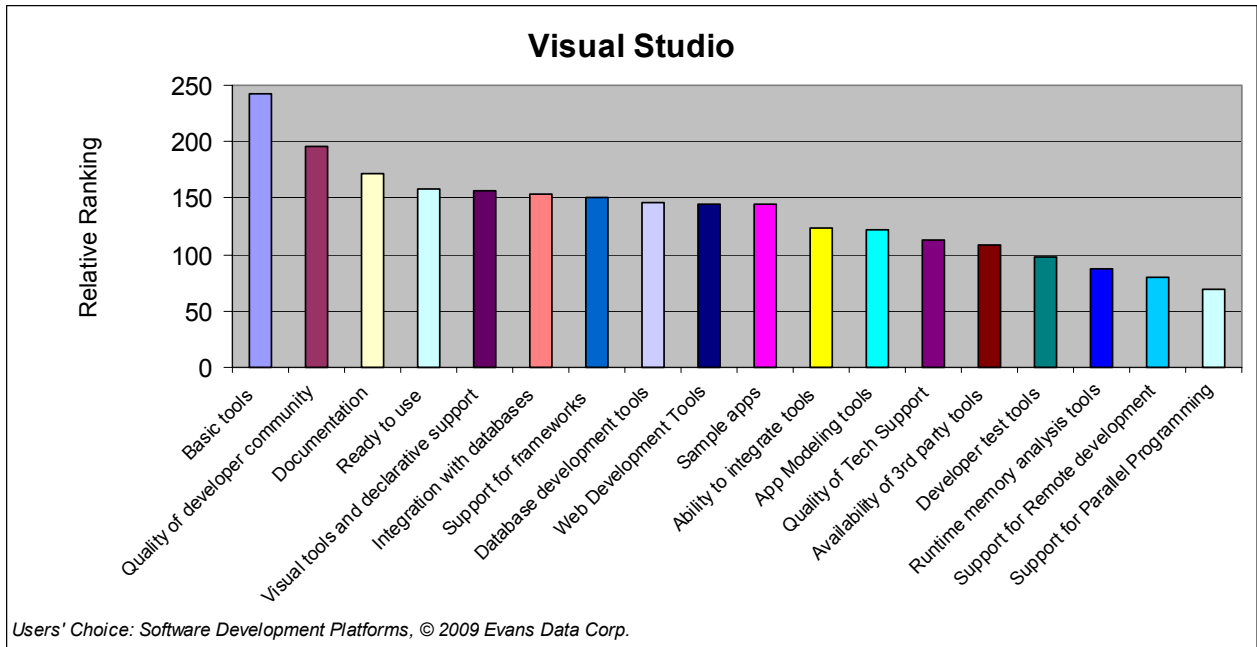
Visual Studio

Visual Studio is the combination of Visual C++, Visual Basic, Visual C# and Visual J# – all in one package. Ten years ago these were all sold separately but the idea of an IDE that could be used across languages was a good one, as most developers use multiple languages. Visual Studio is a complete development platform with a complete tool suite that can be used to create programs, websites, web applications, and Web Services that run on Microsoft Windows, PocketPC, Smartphones, and the World Wide Web.

“Microsoft Visual Studio is by far the most used IDE anywhere in the world with about twice as many users as any of its competitors.”

Microsoft Visual Studio is by far the most used IDE anywhere in the world with about twice as many users as any of its competitors. It is supported by the world’s largest and best implemented developer relations program and it serves the world’s largest installed base of platform users.

The latest version of Visual Studio (Visual Studio 2008) was released in November of 2007 along with the new .Net 3.5. An early beta of Visual Studio 2010 is now available, though it’s



probably safe to assume that most of the people taking this survey are reporting on 2008 or earlier versions. Visual Studio 2008 features; the ability to target multiple versions of .Net, built-in ASP.NET and Ajax support, a significantly improved HTML web designer (the same one that ships with Expression Web), with support for split-view editing, nested master pages, and improved CSS integration. It also contains LINQ (language integrated query) a feature that makes querying and working with data easier.

While the developers in this survey liked the basic tools that ship with Visual Studio –solid technologies with a long

history at Microsoft, one of the most highly rated aspects of Visual Studio was the size and quality of its developer community. These days Open Source development is famous for its communities and it's easy to forget that Microsoft commands the largest and best established developer community anywhere. That developer community means it's simple for Visual Studio developers to find helpful tips and tricks, online support groups, and even jobs. Another perhaps surprising thing about the results was that Visual Studio actually received very good marks for being ready to use right out of the box. Of course the entire "Visual" line of development tools, which stretches back to the advent of Visual Studio were based on the concept of wizards and making development tasks visual instead of text based, so thus easier. It impressive that such a powerful toolset with so many features and functions has managed to keep the essential ease of use it originally conceived.

Parallel programming was not considered a strength for Visual Studio, but that may change during the coming year as developers become more familiar with Intel's Parallel Studio for Visual Studio. It provides pre-threaded libraries along with a compiler, debugger and analyzer to help developers create multi-threaded applications.

Aside from the fact that Visual Studio produces code that's optimized for Windows and, .Net and Windows is the largest target market anywhere, its main strengths are the quality of the toolset, and the extent and quality of its developer relations program. Microsoft has long understood the need to cultivate developers to write for their Windows platform and has gone to great lengths to make sure those developers are happy with their tools and their programs.

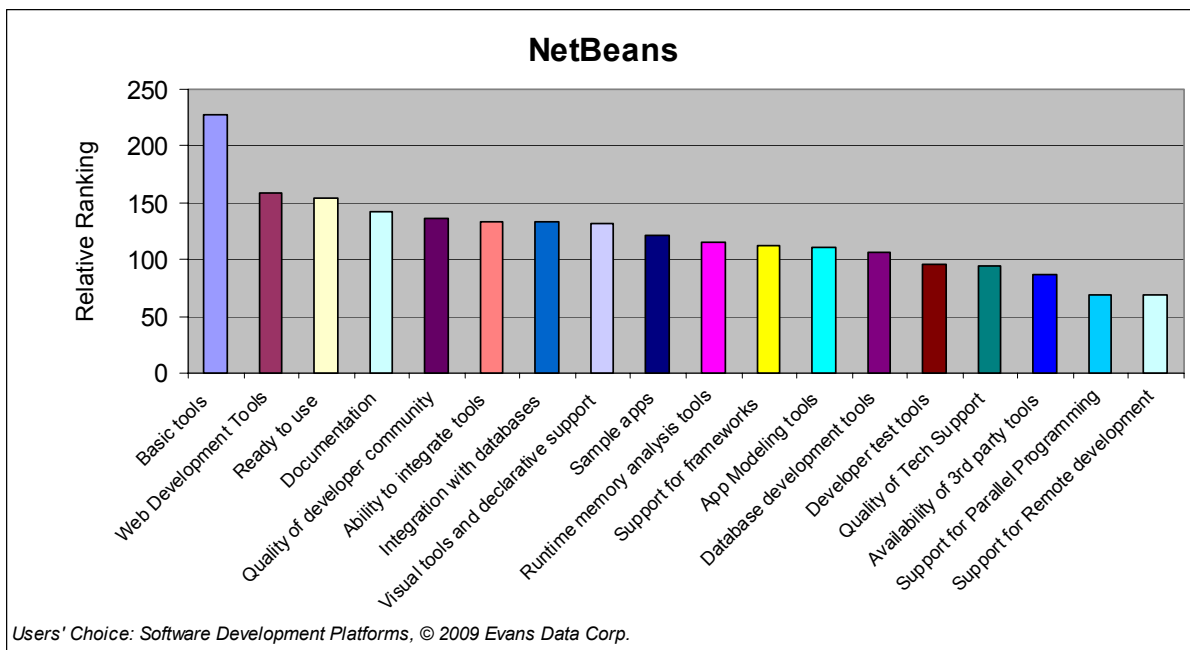
NetBeans

“ The profile that NetBeans shows us today is that of an easy to use web development platform with good supporting documentation and a top quality development community.”

NetBeans began in 1997 as an IDE and platform for Java and it was bought by Sun Microsystems in 1999. Sun Open-Sourced the NetBeans IDE the following year. Since then, the NetBeans community has continued to grow. Applications are built on modules, and can be extended by adding new modules. Thus applications based on NetBeans can be easily extended by third-party developers. This is similar to the Eclipse model.

However, NetBeans has had a life of its own and continues to thrive. After supporting NetBeans along with Sun Java Studio for the last few years, Sun has now decided to turn its attention squarely onto NetBeans as its primary Java tools focus and has discontinued Sun Java Studio, though Sun Studio (for C) still exists.

Today NetBeans is a complete open source development platform written in Java using the NetBeans Platform. NetBeans supports development of all Java application types (J2SE, web,



EJB and mobile applications). Among other features are an Ant-based make build system, version control, and refactoring.

NetBeans is currently at version 6.5 though a release candidate for version 6.7 is also available. However, the chances are that most developers answering this survey are referring to earlier versions. Version numbers were asked for the various tools suites and are available as a service drill down for those who are interested. Please contact our Sales department.

As with last year, the NetBeans users were most impressed with the basic tools – the wheels of the car, rather

than the GPS or stereo. However, they are also impressed with the ready to use out of the box experience, and of course the web development tools. NetBeans has moved up steadily in this survey from three years ago when its users rated it almost last until this year when its users brought it into third place overall. The profile that NetBeans shows us today is that of an easy to use web development platform with good supporting documentation and a top quality development community.

NetBeans' other strengths are in its support for multiple language technologies, the powerful nature of the supporting toolset, and its single "straight out of the box" installation that needs very little configuration. In addition, NetBeans includes a profiler and a popular Swing GUI builder – Matisse, which simplifies GUI development and automates a lot of the tasks that developers must complete.

NetBeans' most obvious direct competitor is Eclipse. Eclipse has by far the larger user audience, which helps it to provide a greater range of support and toolset plug-ins. NetBeans, on the other hand is pre-configured for an easier installation and configuration experience, and Sun has put in a lot of work providing robust support for new dynamic languages.

NetBeans is mainly distributed from the NetBeans website, although Sun does provide NetBeans on media by request at its Sun Tech Days training events, as well as at various conferences conducted worldwide. As an open source IDE, NetBeans is free, and is supported mainly by the NetBeans community.

IntelliJ

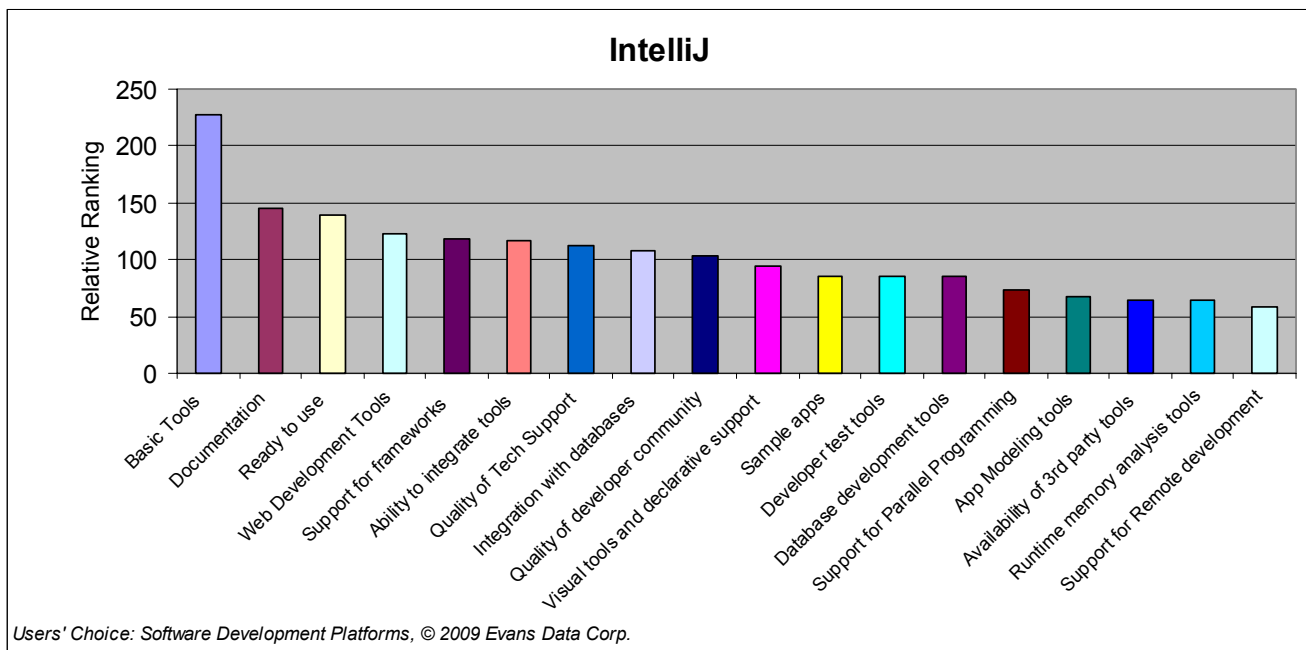
IntelliJ IDEA is a Java IDE from JetBrains, a Czech company with offices in the U.S. and Russia. It's popular with Java developers and has a devoted following. It is promoted as an "intelligent IDE" which can aid in developer productivity, but now it also supports other languages including JavaScript, Flex, XML, XSL, Ruby, JRuby, and Groovy. The latest version of Idea (8.1) includes UML roundtrip engineering, visual Hibernate modeling, and a Swing based GUI designer.

As with most of the rest of the IDEs, the IntelliJ users found the basic toolset to be the most compelling, and as we have mentioned previously, there is an obvious reason why – the basic tools are the necessities. Another reason is that the developers spend more of their time using basic tools than using or being concerned by some of the other attributes that were measured in the survey.

IntelliJ users also liked the documentation and the ready to use experience. The feeling is that users think of IntelliJ as an easy to use high productivity web development platform. The languages it supports all lend themselves to web development, and with an emphasis on scripting languages, this toolset is most likely to be used for those implementations. Another credit that IntelliJ has is its ability to integrate tools – mostly open source tools that might also integrate with Eclipse.

A personal license for IntelliJ costs \$249, and a commercial license is \$599.

"...users think of IntelliJ as an easy to use high productivity web development platform."



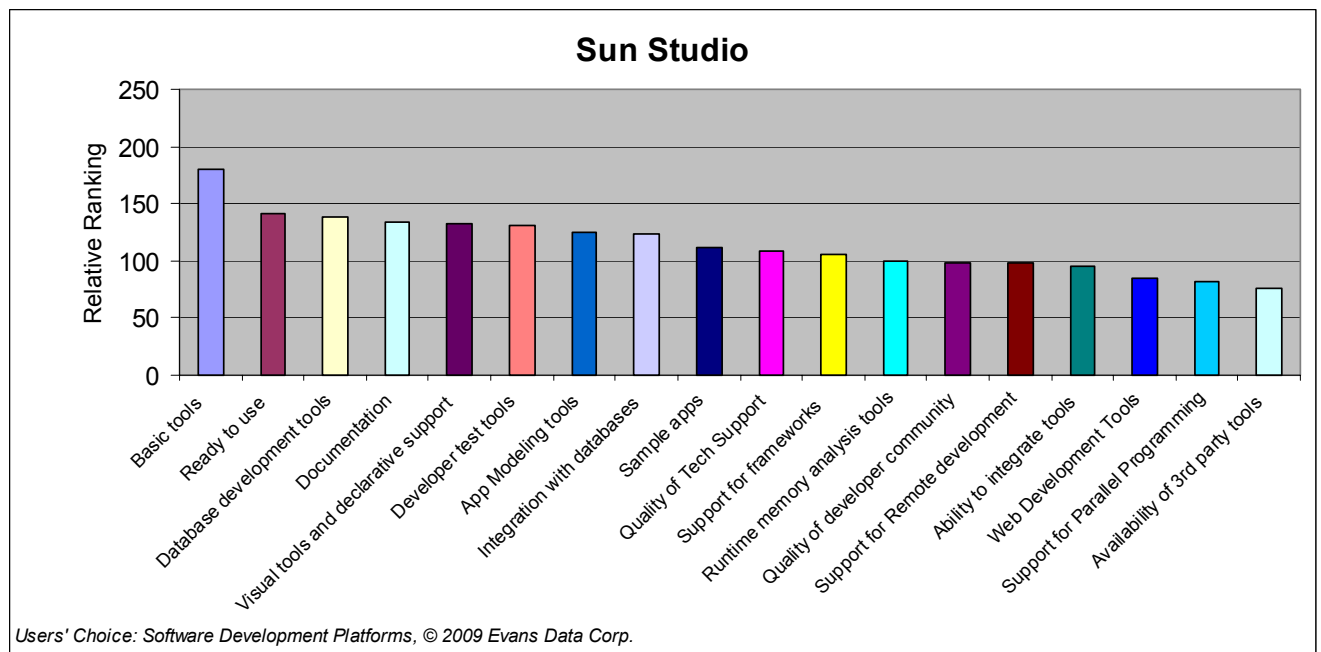
Sun Studio

Sun Studio is Sun's high-end development system for C, C++ and Fortran. It supports Solaris, OpenSolaris, and the Linux platforms. Stressing performance, it is designed to optimize implementations on Sun's SPARC multi-core chips, as well as on dual and quad core processors from Intel and AMD. According to users, Sun Studio does an excellent job of producing the ultimate in applications optimized for performance.

Sun's traditional market for high-end servers includes the finance, telecom, corporate Fortune 500 enterprises, along with high performance computing, and Sun Studio, the toolchain of choice for Sun server installations, is equally focused on these areas.

The product provides a complete and polished user experience with a fully integrated toolchain: IDE, optimizing compilers, code/memory/thread debuggers, high-performance

"But the capabilities that really set the product apart are those designed to increase performance and to enable parallel programming (auto-parallelizing compilers, thread debugging, thread profiling, thread analysis, OpenMP, etc.)."



libraries, and more. In addition, it provides for a degree of cross-platform development across the three that it supports. But the capabilities that really set the product apart are those designed to increase performance and to enable parallel programming (auto-parallelizing compilers, thread debugging, thread profiling, thread analysis, OpenMP, etc.). These capabilities were enhanced further with additional performance improvements for Intel and AMD architectures. Sun Studio now claims to be competitive even with compilers from Intel and AMD. That's undoubtedly why they were second only to IBM in this survey relative to the competition on parallel programming. The fact that users rated that capability low in relation to other

Sun Studio capabilities is an indication that this capability is not well used.

Sun sees the Sun Studio competitors as mainly those open source tools, such as GCC and GDB, as well as the commercial Intel Compiler Collection for High Performance Computing. While Sun Studio is a free license, it is not completely open source, and does not have the same type of community of developers that other open source platforms in this survey do.

As multi-core processing becomes more mainstream developers are going to have to adapt their new and existing applications for multi-threading to take advantage of the multiple cores. Sun Studio's auto-parallelizing compilers can take existing single thread apps and recompile them into multi-threaded applications- multi-thread code written as a single thread. It also includes thread debugging; the ability to follow execution and debug state across multiple threads; thread profiling, performance analysis of threads; thread analysis tools to identify race and deadlock conditions, OpenMP, visual, clock-profiling, hardware counters, and more.

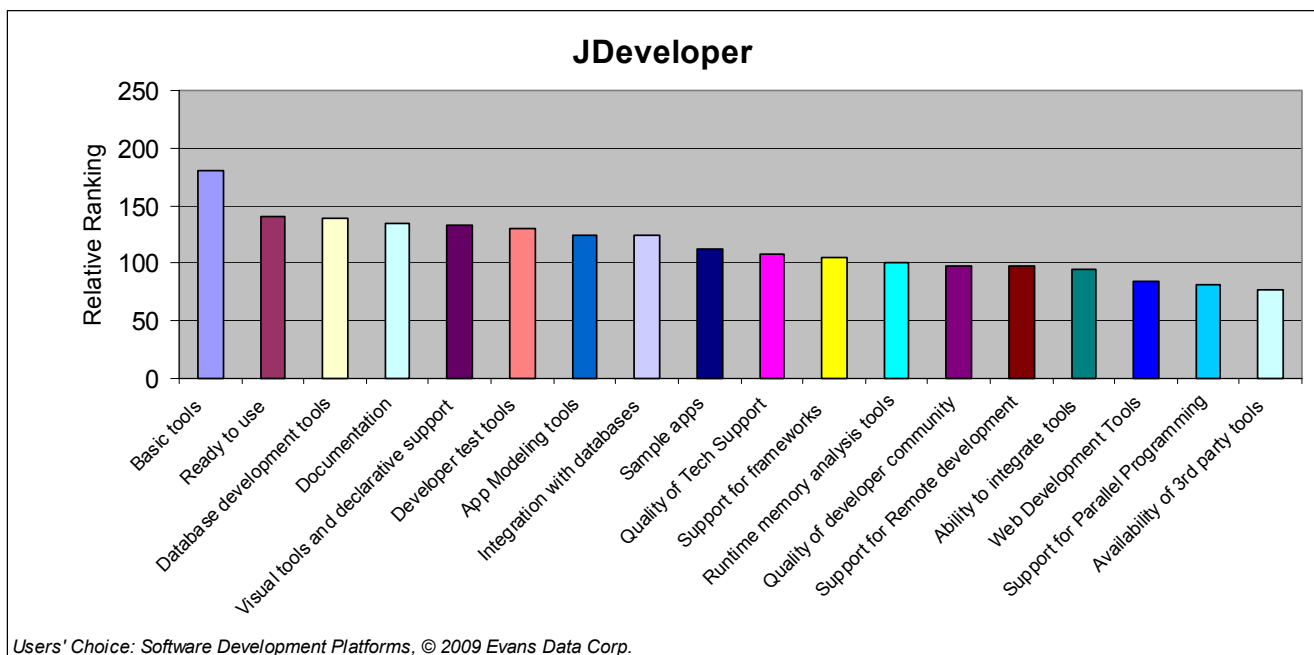
JDeveloper

“Today it is a full featured software development and delivery platform and includes database admin functions as well as SOA enabling tools, tools for BPEL, XML capabilities and a full spectrum of reusable components.”

JDeveloper began life as a licensed version of Borland's JBuilder ten years ago. However, in 2001 Oracle completely re-wrote the product from the ground up and today there is no JBuilder code left in it. In fact, Oracle spent lots of talent and effort on creating JDeveloper, not just as an IDE, but as an entire platform for the full Oracle family of applications; tooling, CRM, portal development, database, and application servers, and to very good result. Today it is a full featured software development and delivery platform and includes database admin functions as well as SOA enabling tools, tools for BPEL, XML capabilities and a full spectrum of reusable components. It sports greatly improved visual tools and powerful declarative support, along with Object Relational Mapping and Web 2.0 support - all in an easy ready to use - straight out of the box experience.

JDeveloper's strengths also include strong support for frameworks. JDeveloper ships with the Oracle Application Framework, which is designed to make development easier and more powerful through the use of visual objects and a very complete framework

Of course, JDeveloper has no rival when it comes to database development tools and database integration. It is strongly integrated with existing Oracle databases, but Evans



believes with the pending acquisition of Sun and the MySQL database we can expect JDeveloper to extend and enhance support for the entire world of data management from small shops using the low end open source MySQL all the way up to

the Fortune 1000 corporations running Oracle's high end namesake database.

JDeveloper is targeted at enterprise developers building enterprise applications, including both the traditional Java/Java EE developer as well as SOA developers and particularly those from a 4GL background (using tools such as Oracle Forms, PowerBuilder and Visual Basic), as well as developers who are doing custom development to front the set of Oracle enterprise applications (E-Business Suite, Siebel, Peoplesoft).

Included tools cover the full development life cycle from UML modeling through coding, testing, tuning, analyzing and deployment. In addition, they offer a highly visual approach to development with such features as WYSIWYG design of UIs (web and desktop), visual page flows diagrams for JSF and Struts, Modeling of Java Classes and EJBs, Visual editors for XML Schemas and WSDL. And their visual editors, property inspectors and structure window are synchronized with the code editor offering a choice of development approach. JDeveloper's visual editors and page flows along with the declarative wizards for creating business services make the IDE and toolset much more declarative in nature compared with traditional Java tools - and as such offers an easier transition and better productivity to enterprise developers.

Oracle's rankings confirm these strengths. As with all the IDEs JDeveloper was rated best on the basic tools, but all IDEs showed that profile and it reflects the general attitude of developers towards those basic tools. The ready to use nature of JDeveloper was appreciated by the JDeveloper developers, as were the database tools and documentation. Ready to use was a comparative strength even though it scored lower than some of the less complete platforms in the competitive scores. Although Application Modeling tools were a bit further down the list of JDeveloper's strengths when compared to its other features, it was second to IBM's Rational tool suite in this area and one of only two to receive good marks in this respect.

It was also very strong competitively in its support for visual tools and declarative programming. It beat all others hands down in database development tools and integration with databases, but was also quite strong competitively in its support of frameworks.

An area for improvement is in regards to support for third party tools, both in availability and thus growing their community, and in support for parallel programming. In Evans opinion, this latter will likely improve as Oracle integrates technologies from Sun including some of its relatively strong parallel programming tools

JDeveloper is a free development tool suite and environment, which ships free with no licensing fee.

Evans Data Corp
740 Front St., Suite 240
Santa Cruz, CA 95060

Phone:
800-831-3080

Fax:
831-425-7913

Website:
www.evansdata.com

Evans Data Corp. (EDC) has made every effort to produce the highest quality research product in this series. The customer understands that EDC uses those statistical and data-gathering techniques which, in its opinion, are the most accurate possible. However, inherent in any statistical inquiry is a possibility of error, which must be taken into account in evaluating the results. Evaluations and interpretations of statistical research findings and decisions based on them are solely the responsibility of the customer, not EDC. The conclusions, summaries and interpretations provided by EDC are based strictly on the analysis of the data gathered and are not to be construed as recommendations; therefore, EDC neither warrants their viability or accuracy nor assumes responsibility for the success or failure of any customer actions subsequently taken.

This report is the exclusive and proprietary property of Evans Data Corp. and is subject to limited distribution and restricted disclosure only. Any unauthorized use, reproduction or transfer of this material is strictly prohibited. Clients of this study may freely distribute this report to full-time employees within their corporate entity, providing that all recipients adhere to the above requirements. However, none of the information in this report may be distributed outside any client's company without expressed written permission of EDC.