



Building Communities

Creating a conversation

February 2, 2006

Eclipse Today



- 120+ members, including major Java, Linux and Embedded vendors (Actuate, HP, IBM, Sybase, BEA, Borland, JBoss, SAP, RedHat, Novell, Nokia, Monta Vista, Wind River, Mentor, ENEA, QNX)
- 50+ open source projects
- 55-65% market share of Java IDE
- Downloads: 50 million++; 1 million in 40 days for last release
- 70 project leaders, PMC members
- 618 committers
- 24,426 registered Bugzilla users

Eclipse Foundation, Inc.

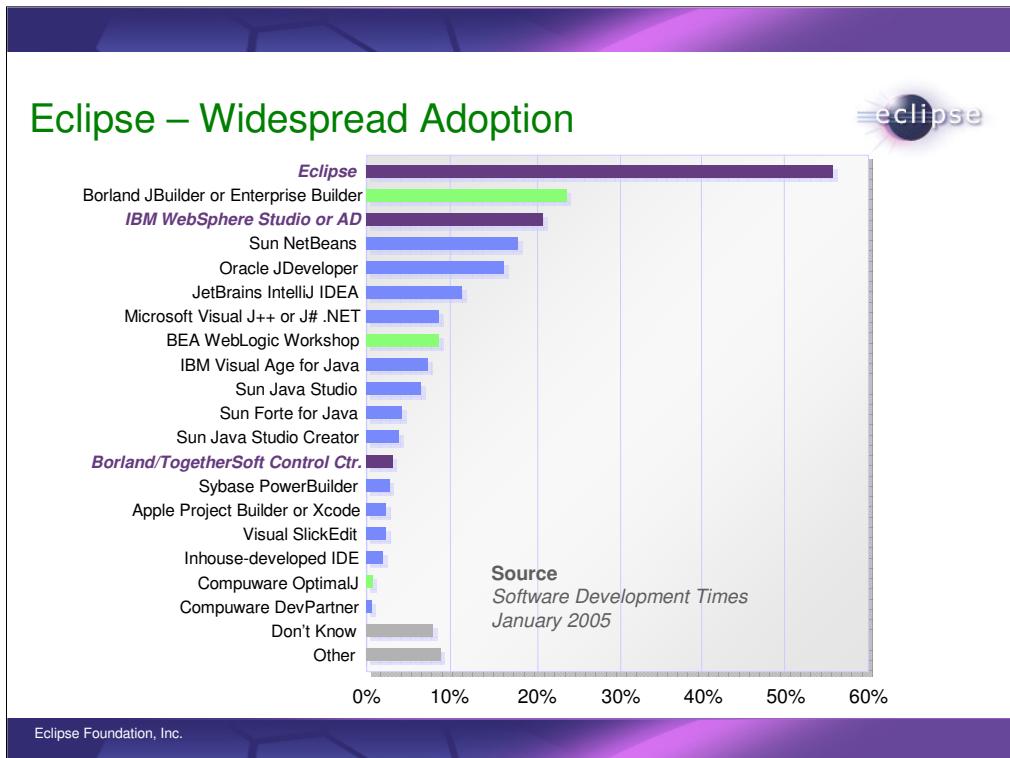
Eclipse 3.1 Metrics



- Number of newsgroup posts: 32,223
- Number of mailing list posts: 7,792
- Number of Eclipse related [blogs](#): 27
- Number of [eclipse.org](#) page views: 500,000,000
- Average Bugzilla queries per day: 600,000
- Bugzilla bug or enhancement reports entered: 21,831
- Bugzilla reports fixed: 9,871
- Most comments on a single bug: 202

Eclipse Foundation, Inc.

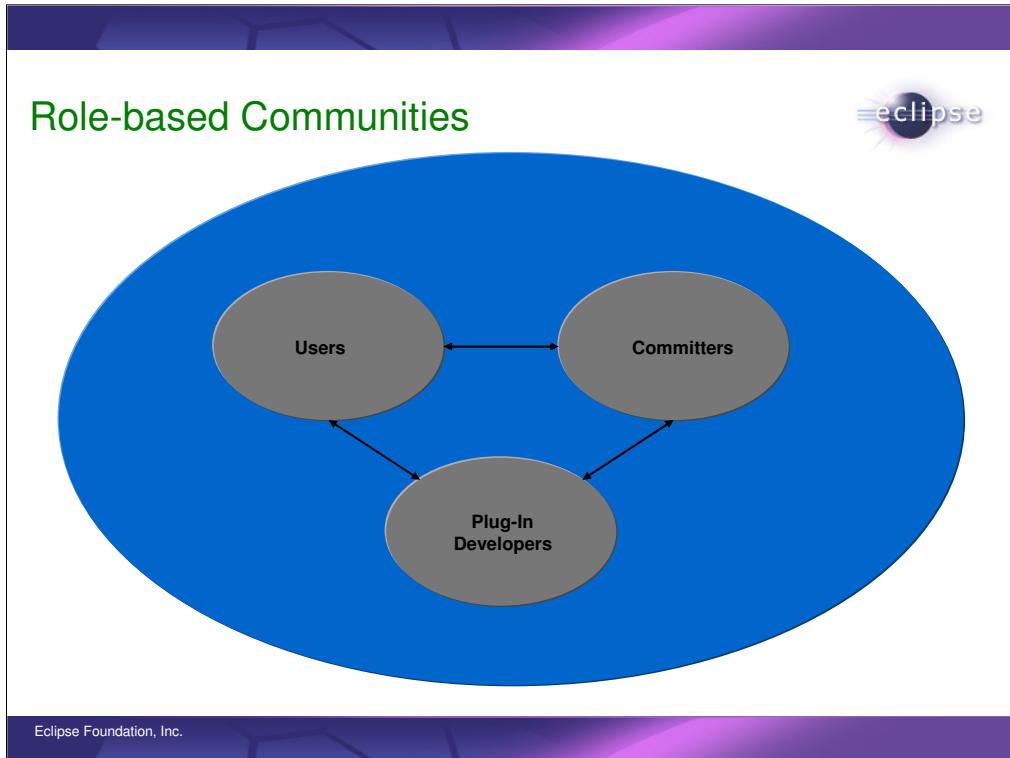
An example of the community activity is the metrics around the Eclipse 3.1 release.



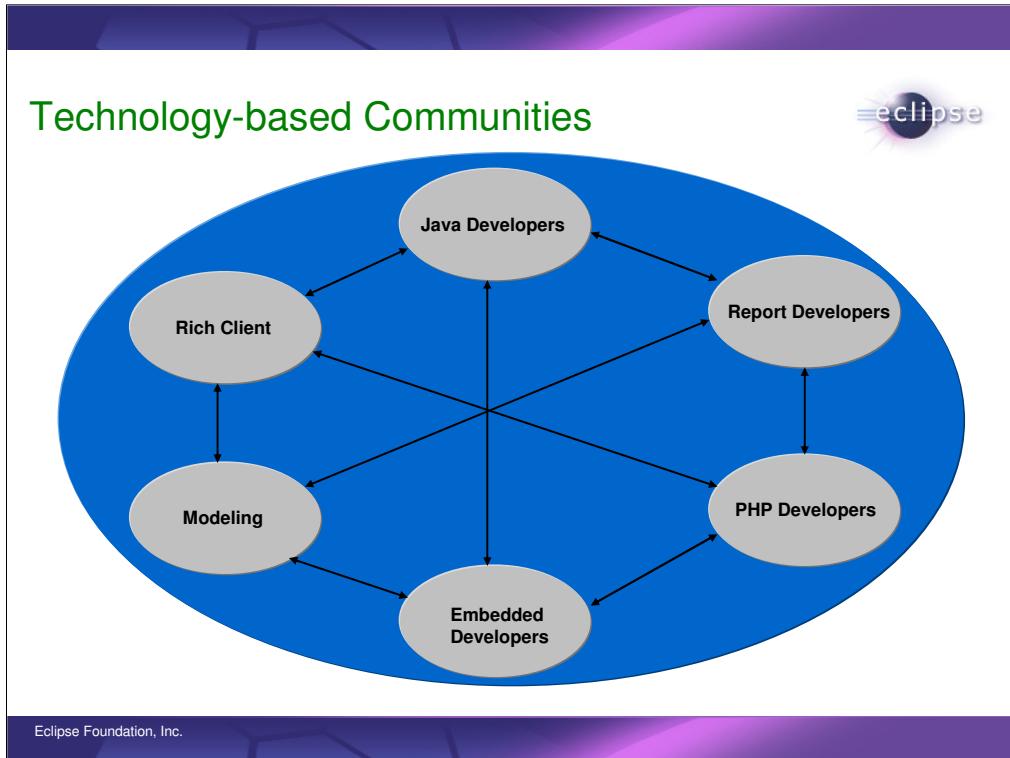
Of course a measure of success is that people are using us.



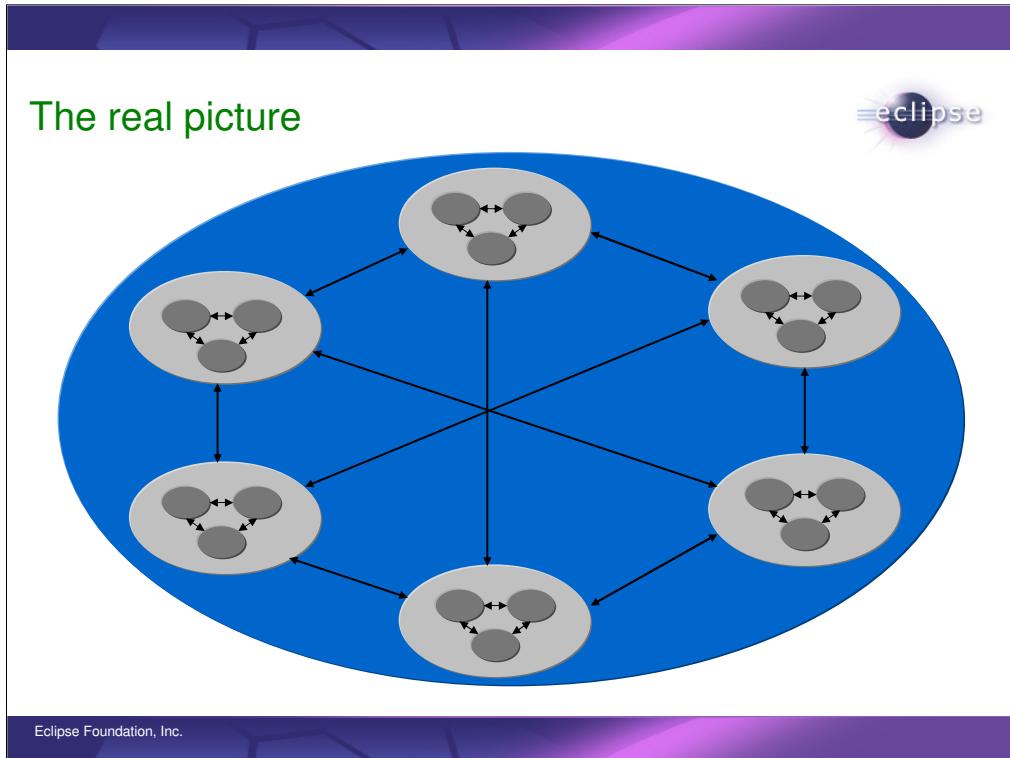
Lots of people think of Eclipse of one big community.



However, we think of the community in two dimensions, the first begin role based. Within the Eclipse community there are users, committers and plug-in developers. Each of these roles participate in the community in different ways and for different purposes. Therefore it is important to approach each role differently.

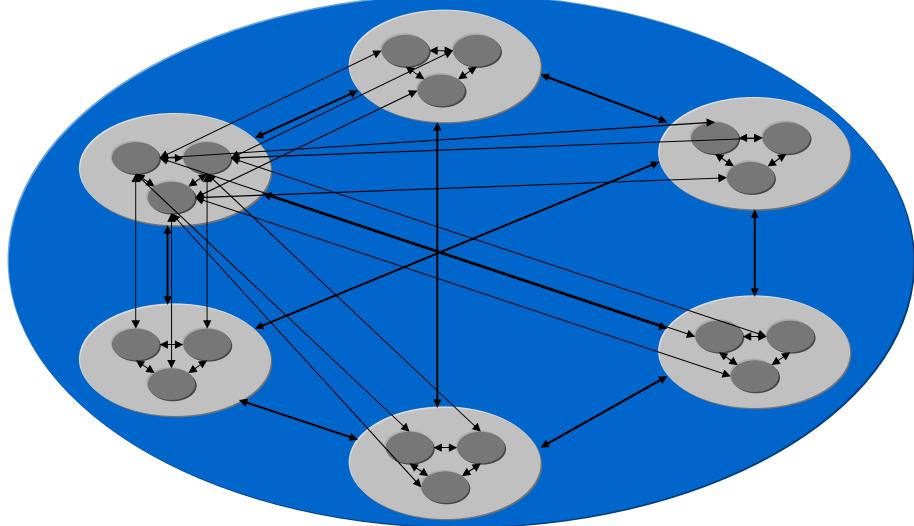


Eclipse is also a set of smaller communities that are developed around different technologies. They often leverage and work together but they do so from different perspectives.



So the real picture is that within the different technology communities, there exists different roles. There is also a lot of collaboration between the participants across roles and technology.

Metcalfe's Law Works for Communities as Well



Eclipse Foundation, Inc.

Of course the network effect starts to take off even within developer communities. The more users you have, the more plug-in developers you get and the more committers are attracted.

Communities are about conversations

eclipse

Eclipse Foundation, Inc.

How do we think about building and communicating to the Eclipse community? A lot of what we do has been documented and discussed in different books and blogs. There are a few. If you would read one book, everyone needs to read The Cluetrain Manifesto.

Cluetrain Manifesto – 95 Theses



1. Markets are **conversations**
2. Markets consist of **human beings**, not demographic sectors.
3. Conversations among human beings sound human. They are conducted in a **human voice**.
9. These networked conversations are enabling powerful new forms of social organization and knowledge exchange to emerge.
- 62 Markets do not want to talk to flacks and hucksters. They want to participate in the conversations going on behind the corporate firewall.
- 90 Even at its worst, our newfound conversation is more interesting than most trade shows, more entertaining than any TV sitcom, and certainly more true-to-life than the corporate web sites we've been seeing.

Eclipse Foundation, Inc.

The Cluetrain manifesto include 95 theses that describe their approach. For the purpose of this presentation, the key thing is that markets are conversations with human beings. These conversations are doing enabling powerful networks. For vendors is be part of the conversations you need to use a human voice and not use market-speak. The risk of not doing this, is that these conversations are where people are turning to to get their information.

Starting the conversation



Developers want to talk to
developers

Fire the Marketers!

Eclipse Foundation, Inc.

The starting point for any developer oriented community must be developers talking to developers. Developers see through market-speak and don't respect it. Marketers shouldn't be fired but they need to change their tactics if they are going to be successful.



“My development manager thinks we are being out marketed, so I would like to do a press release....”

- Developers and Marketers need to share the responsibility of creating the community.
- We are all marketers!
- btww, press releases do not create communities

Eclipse Foundation, Inc.

The quote is based on a true story of someone talking to me. The important point is that developers need to take joint responsibility on the success of building their community. It is not just a marketing role.

Starting the conversation



The Conversation Starts with Code

18 month release cycles don't cut it;

Try 6 weeks

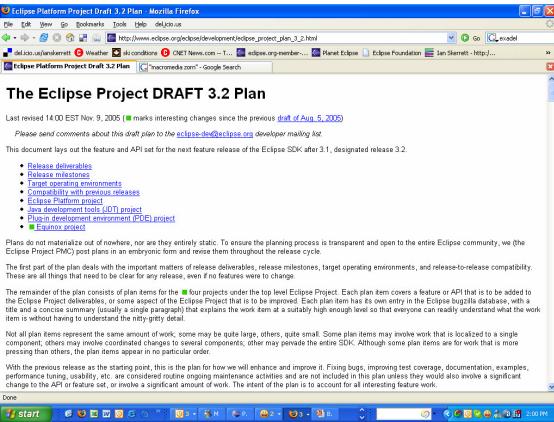
Eclipse Foundation, Inc.

Developers like to talk about code and technology. Before any community can be started you need something to talk about and that is typically the code. To keep the conversation going you need to provide updates often.

Case Study



- Release due June 2006 but in January 2006 already 64,100 Google hit on “Eclipse 3.2”



The Eclipse Project DRAFT 3.2 Plan

Last revised 14:00 EST Nov. 9, 2005 (■ marks interesting changes since the previous draft of Aug. 5, 2005)

Please send comments about this draft plan to the eclipse-dev@eclipse.org developer mailing list.

This document lays out the feature and API set for the next feature release of the Eclipse SDK after 3.1, designated release 3.2.

- [Release deliverables](#)
- [Release milestones](#)
- [Target operating environments](#)
- [Cross-project dependency releases](#)
- [Eclipse Platform project](#)
- [Java development environment \(JDT\) project](#)
- [Plug-in development environment \(PDE\) project](#)
- [Eclipse project](#)

Plans do not materialize out of nowhere; nor are they entirely static. To ensure the planning process is transparent and open to the entire Eclipse community, we the Eclipse Project Plan team publish an interim plan and review it before the final release cycle.

The first part of the plan deals with the important matters: release deliverables, release milestones, target operating environments, and release-to-release compatibility. These are all things that need to be clear for any release, even if no features were to change.

The remainder of the plan consists of plan items for the ■ four projects under the top level Eclipse Project. Each plan item covers a feature or API that is to be added to the Eclipse Project deliverables, or some aspect of the Eclipse Project that is to be improved. Each plan item has its own entry in the Eclipse bugzilla database, with a title and a concise summary (usually a single paragraph) that explains what the item is at a suitably high enough level so that everyone can readily understand what the work item is about and its importance.

Not all plan items represent the same amount of work; some may be quite large, others, quite small. Some plan items may involve work that is localized to a single component, others may involve coordinated changes to several components; others may pervade the entire SDK. Although some plan items are for work that is more pressing than others, the plan items appear in no particular order.

With the previous release as the starting point, this is the plan for how we will enhance and improve it. Fixing bugs, improving test coverage, documentation, examples, performance, stability, etc. are considered regular maintenance activities and are not included in this plan unless they would also involve a significant change to the API or feature set or involve a significant amount of work. The intent of the plan is to account for all interesting feature work.

Done

Eclipse Foundation, Inc.

The Eclipse project is scheduled to release their next major release in June 2006. It is called Eclipse 3.2. There is already 64,000 Google hits on “Eclipse 3.2”.

6 weeks after the 3.1 release, the development team had the first 3.2 M1 release and the project plan for 3.2 up on the web site. They are creating an environment for people to join the conversation.

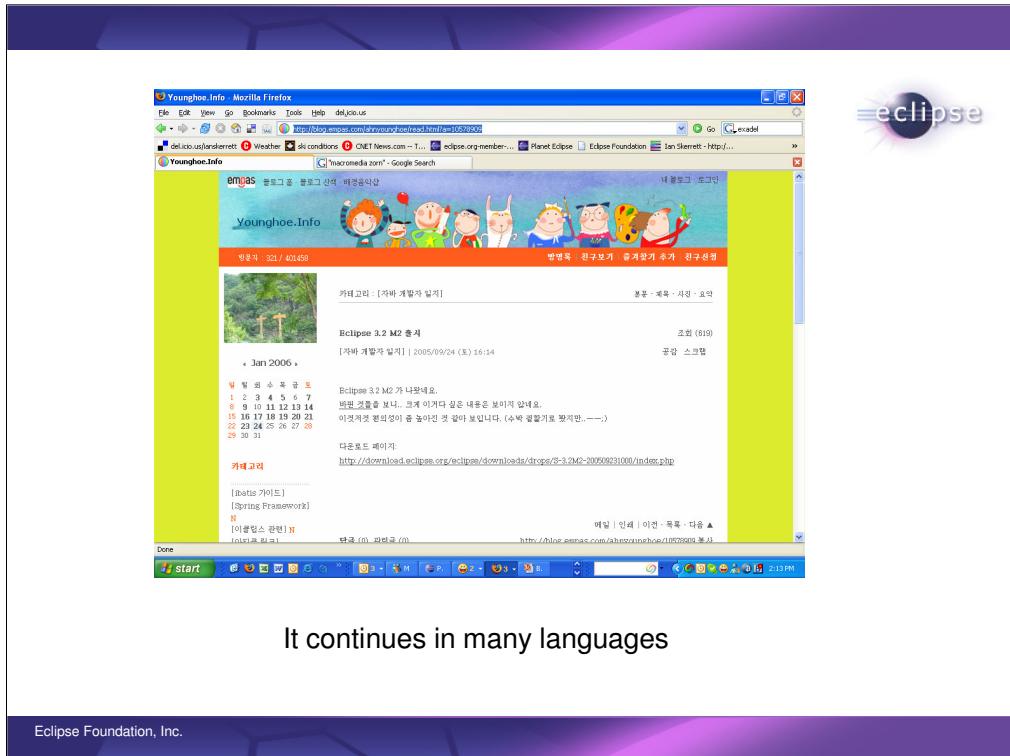
The screenshot shows a Mozilla Firefox browser window with the title "Malcolm Davis's Blog: Eclipse 3.2M1 - Mozilla Firefox". The page content is from java.net, specifically a blog post by Malcolm Davis titled "Eclipse 3.2M1". The post discusses the release and includes comments from users. The Eclipse logo is visible in the top right corner of the slide.

Comments:

- This has got to be the best and most succinct argument against using Eclipse. Well done!
- There are quite a few visual quirks that were not in 3.1 release (such as Outline view icons randomly showing in the middle of a file pane, or closing the names after searching in Type hierarchy). These will, of course, not fix, but I think that 3.2M1 is not more stable. The different startin...

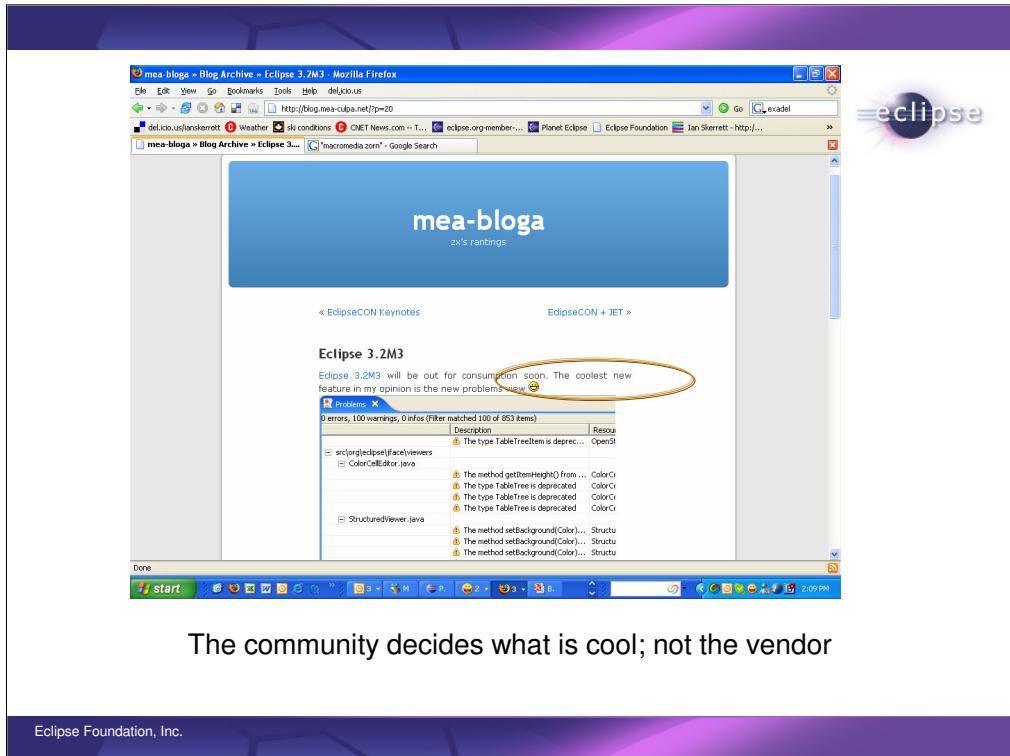
- The Eclipse 3.2 conversation started last September, 2005
- 9 months before the release!

But this conversation started last Fall. The bloggers were already talking about 3.2 in September. Every 6 weeks a new milestone is released and a conversation starts around that milestone.



It continues in many languages

The conversation also occurs in many languages. Communities are multi-lingual



The community decides what is cool; not the vendor

The community also starts to define what is cool, before the press release is written.
This is a great resource for marketing types.

Joining the conversation

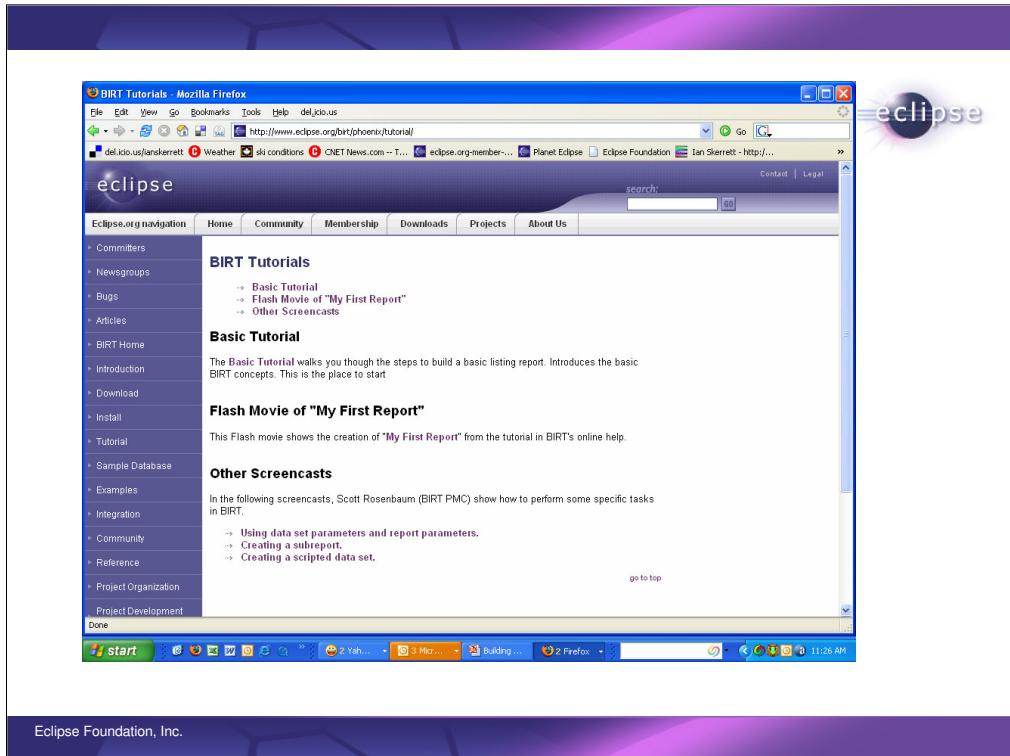


Make it easy to join the conversation

- Need to have low barriers to entry
- Tutorials, white papers, books
- Have experts monitor newsgroups and bug database

Eclipse Foundation, Inc.

Standard marketing states you want to keep low barriers to entry. Building communities is no different. This allows people to join the conversation.



The Eclipse BIRT projects is a good example

The screenshot shows a Mozilla Firefox browser window with the title bar "Rich Client Platform - Mozilla Firefox". The address bar shows the URL "http://www.eclipse.org/articles/Article-RCP-2/tutorial2.html". The page content is titled "Eclipse Article" and features a sub-section titled "Rich Client Tutorial Part 2". The text discusses the Rich Client Platform (RCP) and its evolution from early versions of Eclipse. It mentions the introduction of the API and the PDE. A note at the bottom of the page states: "In this part we'll examine the code and configuration files created in Part 1 of the tutorial. To keep the parts separate I've recreated the examples for each part." The Firefox toolbar is visible at the bottom, showing various icons and the time "11:27 AM".

Eclipse Article

Rich Client Tutorial Part 2

The Rich Client Platform (RCP) allows you to build Java applications that can compete with native applications on any platform. Part 1 of the tutorial introduced you to the platform and the steps used to build the smallest possible RCP program. In part 2 we'll look at what we did in more detail and introduce some of the configuration classes that let you control much of the layout and functionality of an RCP application. This part has been updated for Eclipse 3.1.

By Ed Burnette, SAS
August 9, 2004 (Updated November 21, 2005 for Eclipse 3.1)

Introduction

In early versions of Eclipse, many functions of the IDE were hard-wired into the code. These included the name and location of the File menu, the title of the Workbench Window, and the existence of the status bar. This was fine for the IDE but when people started to use Eclipse as a basis for non-IDE programs, sometimes these things didn't make sense. Although all the source code was provided, it was inconvenient to find the right places that had to be changed.

So, beginning in Eclipse 3.0, the designers refactored the API to make these and other hard-wired aspects of the user interface controllable through public API. Subsequent releases have fine tuned that API and provided tooling in the Plug-in Development Environment (PDE) to lower the barriers to getting started using it. However, you'll eventually need to look behind the PDE wizards and editors to really understand what is going on.

In this part we'll examine the code and configuration files created in Part 1 of the tutorial. To keep the parts separate I've recreated the examples for each part.

Done

Start Home Help Mail Find 11:27 AM

- Eclipse Rich Client Platform started as a community initiative
- Not planned by a vendor

Eclipse Foundation, Inc.

It is not always the vendor that creates communities or lowers the barrier to entry. The Eclipse RCP initiative was lead by the community. A number of individuals saw the potential of Eclipse to be a rich client platform. They started hacking, writing tutorials, writing white papers. In 3.0 the Eclipse project team decided to do the work to formally support Eclipse as an rich client platform.

Joining the conversation



Create an Architecture of Participation

- Enable users and other parties to extend and modify your technology
- Eclipse, Apache, Firefox

Eclipse Foundation, Inc.

Defining the Term



*"what really distinguishes open source is not just source, but an "**architecture of participation**" that includes low barriers to entry by newcomers, and some mechanism for **isolating the cathedral from the bazaar**. This architecture of participation allows for a real free market of ideas, in which anyone can put forward a proposed solution to a problem; it becomes adopted, if at all, by acclamation and the organic spread of its usefulness."*

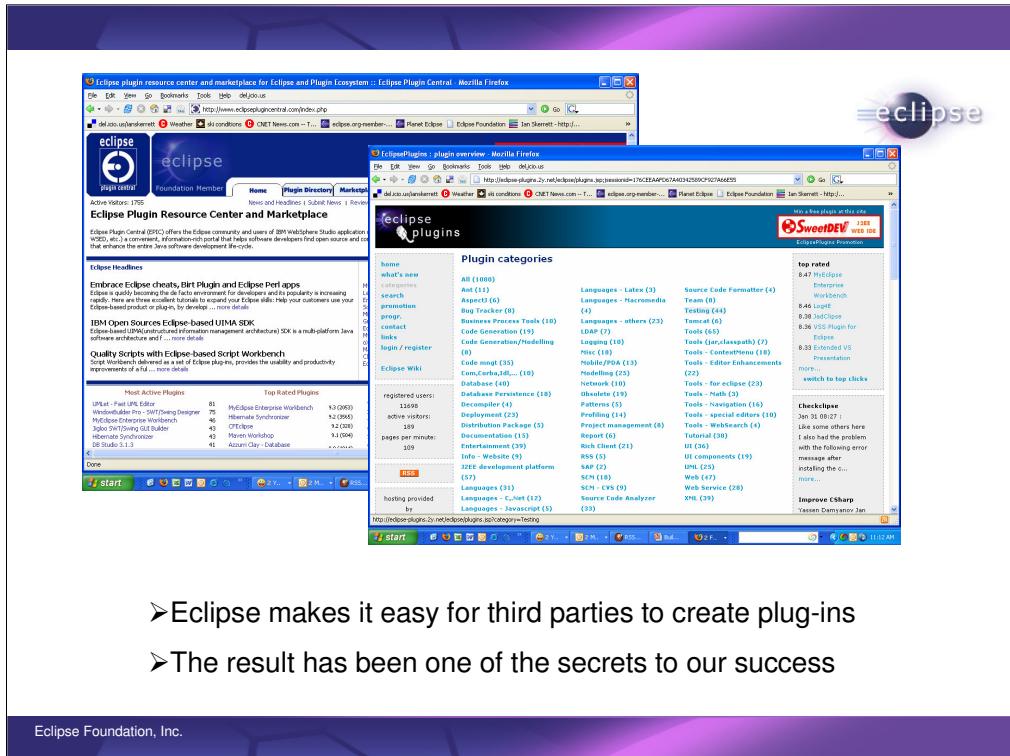
Tim O'Reilly
<http://www.oreillynet.com/pub/wlg/3017?wlg=yes>

Eclipse Foundation, Inc.



Key Points

- You need a cathedral to enable the bazaar
 - This is not about joining the project itself *per se*, but about making it easy for individuals to add their own ideas to the ecosystem
- Low barriers to entry
- New add ons are first class citizens, not second class
- Empowering individuals and small groups is key
 - Innovation comes from committed fanatics in small groups, not corporate committees
- *Laissez faire* --- never ever ever pretend you can pick a winner
- Provide the seeds for a broad ecosystem



➤ Eclipse makes it easy for third parties to create plug-ins

- The result has been one of the secrets to our success

Eclipse Foundation, Inc.

Feeding the conversation



Create and publish technical content

- Technical content feeds the conversation
- People love to generate links

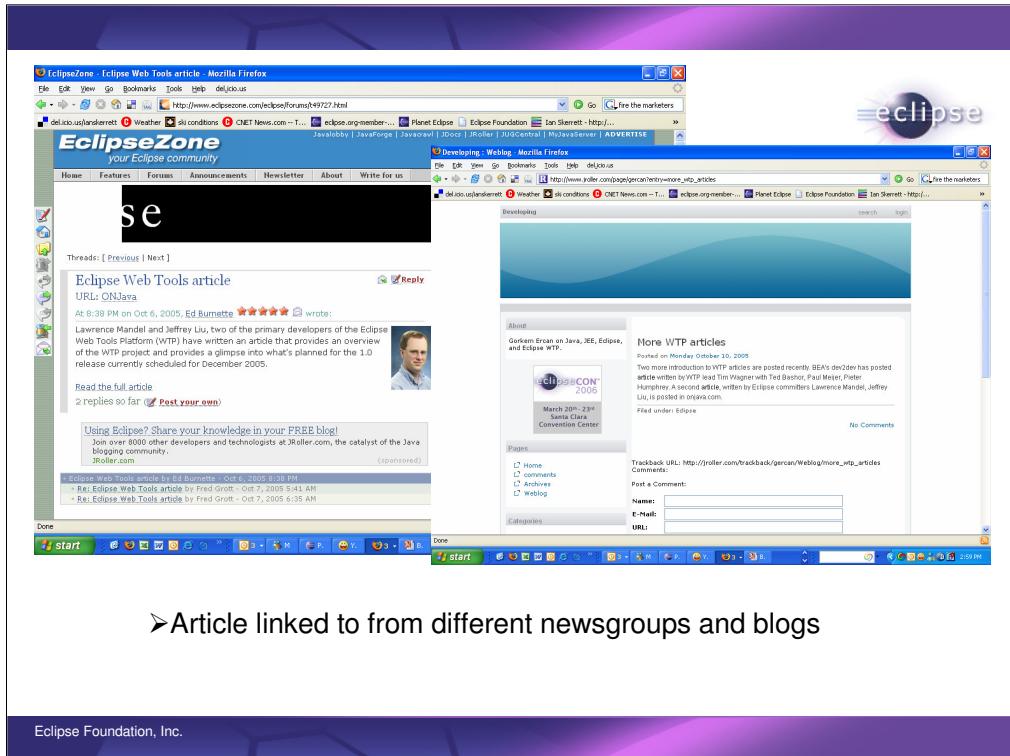
Eclipse Foundation, Inc.

I suggest you think about feeding the conversation for a community. One way of doing this is writing and publishing technical articles.

An article creates the seed for the conversation

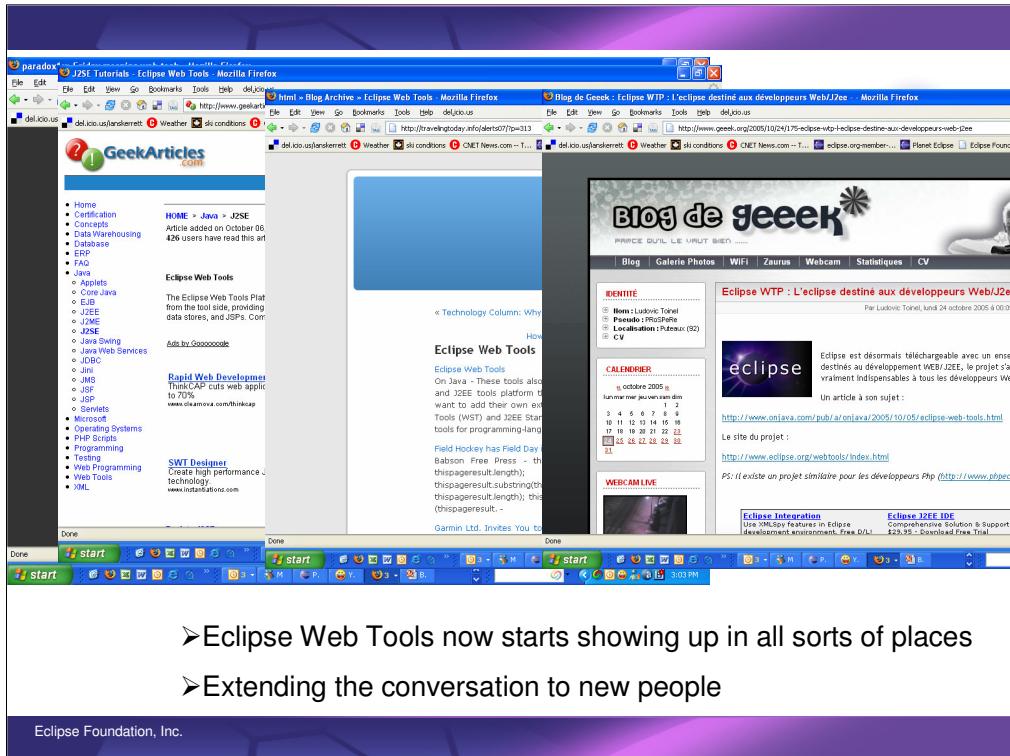
Eclipse Foundation, Inc.

For example, an article written about the Eclipse Web Tools (WTP) project was publishing on onJava. The article by itself is great to get exposure for WTP. However, look what happens next....



➤ Article linked to from different newsgroups and blogs

This results in some of the other Eclipse portals and blogs referencing the article; extending the reach of the article into other publications.



- Eclipse Web Tools now starts showing up in all sorts of places
- Extending the conversation to new people

It is also picked up by other blogs that aren't usually focused on Eclipse, thus extending the awareness and education of Eclipse WTP into the communities not usually reached.

This url has been saved by 35 people. --- show related items

2005-01-23 ONJava.com: Eclipse Web Tools
to eclipse wtp by lanskerrett ... edit | delete

2005-01-20 ONJava.com: Eclipse Web Tool Platform for JEE
to jee java tools by sprangly ... copy

2005-12-14 Eclipse Web Tools
by acapiggi ... copy

2005-12-05 ONJava.com: Eclipse Web Tools
to java by bobmoe ... copy

2005-12-02 ONJava.com: Eclipse Web Tools
to eclipse wtp by muthukrishna ... copy

2005-11-23 ONJava.com: Eclipse Web Tools
to eclipse java by pringle ... copy

2005-11-14 ONJava.com: Eclipse Web Tools
to eclipse webdev/java webservices todo by dirts ... copy

2005-11-12 Eclipse Web Tools
by mokaz ... copy

2005-11-07 Eclipse Web Tools
facilidades para escrever aplicações web com o eclipse
to eclipse jee web jsp ejb by abilio ... copy

➤ People also save it as a resource

Eclipse Foundation, Inc.

It is also saved by individuals for future reference. Del.cios.us show 35 individuals have tagged this article.

Feeding the conversation



Nurture and promote your tribal leaders

- Persona of the community
- Speak at conferences; write books
- Often define the culture of the community

Eclipse Foundation, Inc.

Tribal leaders is a term from the book 'Brand Hijack'. These are the people that define the culture and pace of the community. They can be some of the core developers but also users.



Be as transparent as possible

- Public project plans
- Open bug tracking databases

Eclipse Foundation, Inc.

Users are developers and they want to be involved in the development process of their community. To often they have been in a situation where they report a bug to a vendor and never hear back. A successful community is a transparent community. Develop in the open, exposure your bugs, everyone knows they exist.

Bug 95034 [Graphics] Eclipse 3.1 Branding Graphics - Mozilla Firefox

Bug List: (This bug is not in your last search results) Show last search results Search page Enter new bug

Bug#:	95034	Hardware:	PC	OS:	Windows XP	Reporter:	Michael Van Meekeren mvm@ca.ibm.com	
Product:	Platform	Version:	3.1	Priority:	P1	Add CC:	<input type="text"/>	
Component:	UI	Severity:	normal	Target:	-	Cc:	genologue@yahoo.com eclipse-diseradio@neverbox.org ed.burnett@sas.com gunnar@vegenknecht.org <input type="button" value="Remove selected Ccs"/>	
Status:	RESOLVED	Milestone:						
Resolution:	FIXED	QA Contact:						
Assigned To:	Linda Watson lndawat@us.ibm.com	URL:						
Summary:	[Graphics] Eclipse 3.1 Branding Graphics							
Status Whiteboard:								
Keywords:								

Attachment Type Created Size Actions

Splash-Screen-(Blue)-with-Swoosh	image/png	2005-05-15 09:48	67.67 KB	[Edit](#)			
Splash Screen (Blue) with Swoosh and Blend effect	image/png	2005-05-15 09:57	84.00 KB	[Edit](#)			
eclipse.org.3.1	image/png	2005-05-16 07:14	47.68 KB	[Edit](#)			
img of splash iteration factoring feedback	image/png	2005-05-19 12:23	112.29 KB	[Edit](#)			
Done							

Eclipse Foundation, Inc.

17 people watching!

➤ 81 comments

For example, the splash screen in 3.1 was being changed. Typically this change would have been done by the marketing folks with no discussion. For Eclipse it was done as a bug and it had 81 comments and 17 people watching the bug. Marketing was not involved. A good example of getting community buy-in.

Expanding the community

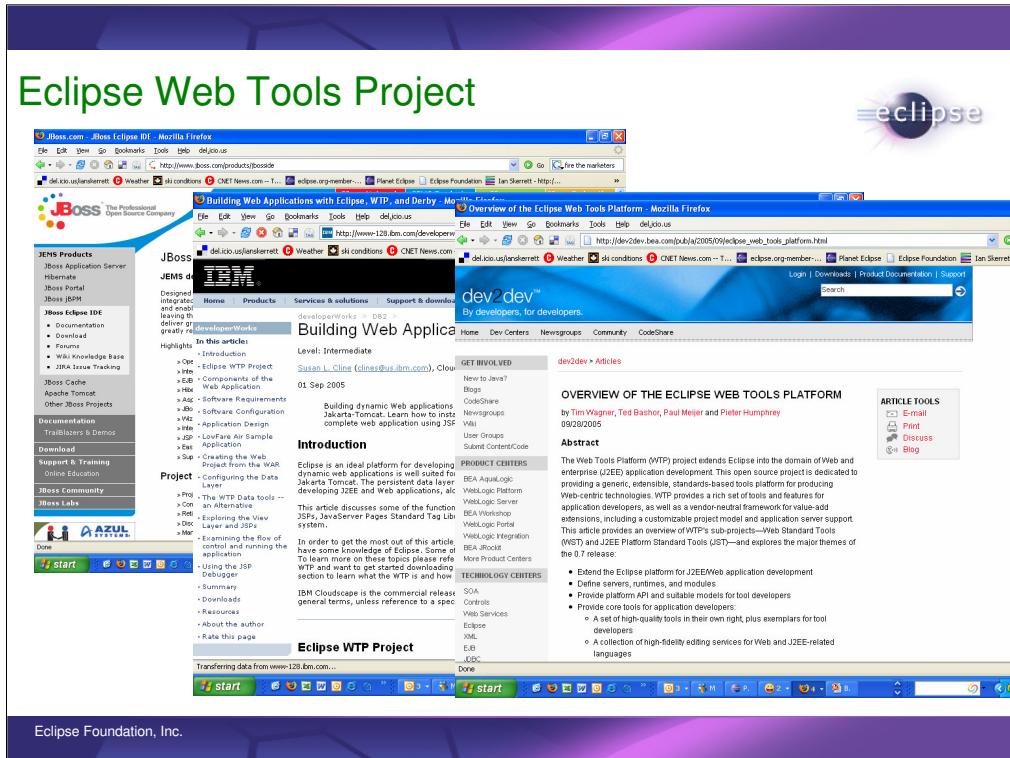


Get your competitors involved

- Allows you to link into other established communities

Eclipse Foundation, Inc.

This is more difficult for vendors as opposed to open source communities but if you can get your competitors involved you instantly start to expand your community.



For example the Web Tools project has IBM, BEA and JBoss involved. One each of the respective web site has information about WTP. This expands the reach of WTP to these vendors developer communities.

Thank the contributors

- Reference blog postings; bloggers love links
- Acknowledge community leaders
- This also helps Google results

Eclipse Foundation, Inc.

You need to think about giving back to the community. Link to blogs, give out awards, highlight other successes. Btw, this also helps with your Google results

Giving back to the community



Promote companies that
use or contribute to the
community

Eclipse Foundation, Inc.

Eclipse is very passionate about ensuring people adopt the technology for commercial adoption. Help promote the fact that other people are using your technology and they will help promote the base technology. A lot of this is an ingredient brand strategy.

Be Part of the Community



If you love something, set it free

➤ Win by letting go

Eclipse Foundation, Inc.

The vendor needs to be part of the community. It can't be a command and control system. Your community will know if you are authentic. You win by letting go.

Pitfalls



- Not all developers are good communicators; or want to be
- Marketers still want to be in control
 - Remember you win by giving up control
- Too easy to fall back into the old way of doing things
- How do you know when you are successful?

Eclipse Foundation, Inc.

What is the marketing role



- Still need to do top-down marketing
 - High level messaging for non-technical decision makers
 - Provide support for developers trying to sell up the chain
- Be part of the community
 - Developers want their community to be successful
 - Make sure you speak their language
- Create a partnership between marketing & development

Summary



- Start the conversation
- Need to make it easy to join the conversation
- Feed the conversation
- Invite others to be part of the community
- Remember to thank the contributors
- Be part of the community

Eclipse Foundation, Inc.



Thank You

Eclipse Foundation, Inc.