

# Speaker Support Notes

*Speaker Support Notes*



Welcome.

<Introduce yourself and other presenters. Go through any "house-keeping" information before you begin>

During this presentation, we will describe a concept called Quality By Design – a software development and test solution that includes best practices, process, and tools that help you deliver your applications faster and with higher quality than ever before.

# Speaker Support Notes

*Speaker Support Notes*

## Agenda

- ◆ Quality by Design
- ◆ O Problema de Negócio do Desenvolvimento de Software
- ◆ A Solução para o Desenvolvimento de Software
  - Processo Claro e Adaptável
  - Comunicação Completa para Toda a Equipe
  - Gerenciamento Eficaz de Requisitos e Mudanças
  - Casos de Uso e Modelagem de Negócios
  - Teste Unitário Abrangente
  - Teste Iterativo de Sistemas

**Rational**  
the e-development company

Let's take a look at the agenda. We are going to start with a definition of quality by design, so we're all on the same page. Then we'll move on to a brief setup of the problem we are addressing here. Finally, and this is where we'll spend most of our time today, we will talk about the solution. Specifically, a solution that includes process, team communication, business modeling, testing, and change management.

# Speaker Support Notes

*Speaker Support Notes*

## Agenda

- ◆ Quality by Design
- ◆ O Problema de Negócio do Desenvolvimento de Software
- ◆ A Solução para o Desenvolvimento de Software
  - Processo Claro e Adaptável
  - Comunicação Completa para Toda a Equipe
  - Gerenciamento Eficaz de Requisitos e Mudanças
  - Casos de Uso e Modelagem de Negócios
  - Teste Unitário Abrangente
  - Teste Iterativo de Sistemas

**Rational**  
the e-development company™

First, let's define Quality By Design

# Speaker Support Notes

*Speaker Support Notes*

## Quality by Design... definição

Uma solução de desenvolvimento e teste de software que abrange melhores práticas, processo e ferramentas que o ajudam a entregar aplicativos com maior qualidade mais rápido que nunca.



**Rational**  
the e-development company

Think of quality by design, as quality by "intent".

Quality By Design is a pro-active approach to software development.

It embracing best practices, process and tools that help you deliver higher quality applications faster than ever before



# Speaker Support Notes

Speaker Support Notes

## Quality by Design: É Possível?

- ◆ Boeing 777
  - Projeto sem papel permitiu aos engenheiros simular a montagem do 777
  - O sistema funcionou tão bem que apenas um boneco do nariz foi construído antes da montagem da primeira aeronave
- ◆ A qualidade pode ser medida em tempo de projeto
- ◆ Testabilidade é um aspecto chave da qualidade



**Rational**  
the e-development company

It Quality by Design possible?

We'll, lets take a look at an example outside of the software industry.

The Boeing 777. One of the world's largest and most sophisticated passenger aircrafts. In fact, it was the first one designed *without* a physical prototype. It was done on a computer. Why is that possible? Because sufficient information was captured in the design process at Boeing to allow *assessment* of the *quality* of the *design*. This is where we need to go in software, capturing the information at the beginning of the process. Validating the design before moving on to the prototype or production phases.

# Speaker Support Notes

*Speaker Support Notes*

## Agenda

- ♦ Quality by Design
- ♦ O Problema de Negócio do Desenvolvimento de Software
- ♦ A Solução para o Desenvolvimento de Software
  - Processo Claro e Adaptável
  - Comunicação Completa para Toda a Equipe
  - Gerenciamento Eficaz de Requisitos e Mudanças
  - Casos de Uso e Modelagem de Negócios
  - Teste Unitário Abrangente
  - Teste Iterativo de Sistemas

**Rational**  
the e-development company™

So, what are the impediments to quality in software development?

# Speaker Support Notes

*Speaker Support Notes*

## Bloqueios da Qualidade

- ♦ Requisitos mal compreendidos
- ♦ Projeto com Falha (Arquitetura e Outros)
- ♦ Mudanças Descontroladas
- ♦ Falta de Métrica de Qualidade do Produto
- ♦ Avaliação de Artigos Centrada em Recursos e Muito Manual
- ♦ Montagem Incorreta
- ♦ Riscos de Desenvolvimento Não Gerenciados
- ♦ Esforços/Componentes de Terceiros Não Gerenciados
- ♦ Recursos Humanos Inadequados
- ♦ Cronograma de Projeto Muito Restrito
- ♦ Pouco Rigor no Processo



**Rational**  
the e-development company™

Obviously, from this long list, there are a lot!

# Speaker Support Notes

## *Speaker Support Notes*

### Bloqueios da Qualidade

- ◆ Requisitos mal compreendidos
- ◆ Projetos com Falha (Arquitetura e Outros)
- ◆ Mudanças Descontroladas
- ◆ Falta de Métricas para Qualidade do Produto
- ◆ Avaliação de Artefatos Centrada em Recursos e Muito Manual
- ◆ Montagem Incorreta
- ◆ Riscos de Desenvolvimento Não Gerenciados
- ◆ Esforços/Componentes de Terceiros Não Gerenciados
- ◆ Recursos Humanos Inadequados
- ◆ Cronograma de Projeto Muito Restrito
- ◆ Pouco Rigor no Processo



**Rational**  
the e-development company

Let's just take a few and discuss them.

First is misunderstanding of the requirements. Development projects that have not been plagued with this problem are few and far between. Without a systematic way to capture, analyze and track changes to an application's requirements you may find yourself at the end of a software development cycle that with a beautiful application.... That doesn't satisfy the business need!

#### Faulty Design

As applications become bigger and more complex, more than a single person is usually needed to design an application. *Distributed design capture lends itself to defective designs.*

*With faulty design comes costly rework when faults are eventually uncovered. And design defects not discovered by designers/developers (unit testing), will be much harder to uncover during system testing.*

#### Uncontrolled change

*The cycle of build – test – rework – respec that tends to go on in any normal development process can spiral out of control if there are not tight reigns on the process. Uncontrolled change leads to an inability to know what the final application really looks like, and difficulty knowing if it is really working.*

*And finally, without metrics to measure your progress toward quality goals – the best you'll have is a "feeling" about how the quality of the application.*

# Speaker Support Notes

*Speaker Support Notes*

## O Problema do Negócio

Os custos de teste são o calcanhar de Aquiles dos projetos modulares. Uma explosão de combinações resulta quando muitas experiências em vários módulos precisam ser testadas...

*"A menos que os projetistas consigam romper a barreira do custo de teste dos sistemas, as opções... podem nem existir."*

*Baldwin & Clark (2000)  
Design Rules,  
The Power of Modularity  
MIT Press*

**Rational**  
the e-development company™

This quote is poignant. [READ QUOTE].

For many organizations, testing is when they begin to focus on quality and start shaking out the problems in the application. This may be way too late.

So what does this quote mean? It means that with complex systems come complex tests. And massive complex testing is expensive. Until designers find a way to inject a quality focus in the design process, the cost of building the complex systems that we can design, will be prohibitive.

# Speaker Support Notes

Speaker Support Notes

## O "Paradoxo do Desenvolvimento de Software"



And this brings us to the heart of what we call the software development paradox.

Traditionally, speed and quality have been opposing forces in software development. And they still are. What has changed, though, is that you can't trade one for the other. In the new economy, you have no choice: you must build it fast *and* build it right.

Software quality has never been more important -- or more visible to your customers. So you best design it in from the beginning.

# Speaker Support Notes



Before we start talking specifics about our solution, let me explain why we are qualified to stand up here today and talk to you about this. Rational has been around for 20+ years and over that time we have identified 4 critical success factors to successful software development. First you need good people – we can't help you there. But with the other 3 we can. We provide process, in the form of an organized, proven approach to developing software -- a methodology or set of engineering best practices. Next, you need to equip your teams with the right tools for the job. You need tools that are optimized for the individual practitioners on your team, along with tools that allow the entire group to collaborate more easily and work more productively as a team. Finally, you need professional services you can call upon to accelerate your success.

And that's exactly what Rational brings to the table – and no other company can make that claim. I'd like to start by talking about best practices.

# Speaker Support Notes

Speaker Support Notes

## Melhores Práticas da Rational: O Padrão De Facto

### Melhores Práticas

*Processo Tornado Prático*

- Desenvolver Iterativamente
- Gerenciar Requisitos
- Usar Arquiteturas Componentizadas
- Modelar Visualmente (UML)
- Verificar Continuamente a Qualidade
- Gerenciar Mudanças

### Fortune 1000

Merrill Lynch, Wells Fargo, Bank of America, Arthur Andersen, Charles Schwab

### Consultoria de e-business

IBM Global Services, Deloitte Consulting, iXL, Lante, ICON Media Lab, CBSI, Keane


### Ponto-coms

Tickets.com, eCAMPUS.com, ImageX.com, Food.com, SubmitOrder.com

### Rational Unified Process

*"Estamos desenvolvendo soluções empresariais de e-business robustas e de alta qualidade em prazos inacreditáveis. ...Rational Unified Process – acelera a entrada no mercado em três a cinco vezes."*

Edwin J. Jaufmann, Jr., C CP, CIO, Provider Solutions Corp.



These basic principles about how to develop complex software that we have learned over the past 20 years are codified in the Rational Unified process. RUP includes 6 tenants of practical software development, shown here on the left. And if you select the RUP then you are not alone. More than a thousand organizations are using the RUP today and many Universities are teaching the RUP in their software engineering curriculums. Among customers we find large companies such as Merrill-Lynch, Ericsson, Microsoft, Motorola, AT&T, Prudential, Citicorp, Xerox, E&Y, Oracle, Deloitte, Aetna, etc.



# Speaker Support Notes

## *Speaker Support Notes*

Over the last 20 years, we also discovered that tools are not enough, and there is more leverage in reducing complexity, improving processes and accelerating the proficiency of teams. We invested heavily in professional services to help customers transition to modern process. Rational has over 1000 software engineering professionals participating 1<sup>st</sup> hand and onsite with organizations that depend on software.

We have also invested heavily in leading edge software technologies and pushed the software engineering state-of-the-art. Rational authors have published over 50 books on software engineering topics and driven many of our lessons learned into industry standards such as the UML and RUP.

Finally, Rational is itself, a large scale software development organization and has leveraged these techniques, technologies, tools and processes internally with outstanding results as evidenced by our product leadership in the market. We have a large development organization which practices what we preach to develop and deliver a very powerful set of software products.

# Speaker Support Notes

*Speaker Support Notes*



I mentioned over 50 books published by Rational authors. Shown here are just a few...

# Speaker Support Notes

*Speaker Support Notes*

## Agenda

- ◆ Quality by Design
- ◆ O Problema de Negócio do Desenvolvimento de Software
- ◆ A Solução para o Desenvolvimento de Software
  - Processo Claro e Adaptável
  - Comunicação Completa para Toda a Equipe
  - Gerenciamento Eficaz de Requisitos e Mudanças
  - Casos de Uso e Modelagem de Negócios
  - Teste Unitário Abrangente
  - Teste Iterativo de Sistemas

**Rational**  
the e-development company™

Hopefully, that gives you some context and comfort to know that we are uniquely qualified to solve the problem of delivering high quality applications in Internet Time. Now, let's transition to talk about the specific solution we propose for a Quality By Design solution.

# Speaker Support Notes

Speaker Support Notes

## O Desenvolvimento de Software é Complexo

- ✓ Arquiteturas de projeto mal definidas exigem mudanças fora de hora
- ✓ Requisitos são indefinidos ou mudam no meio do projeto
- ✓ Descoberta de defeitos no final do projeto ou brechas na arquitetura e no projeto = baixa satisfação do cliente
- ✓ Falta de comunicação entre membros de equipes distintas
- ✓ Artefatos não disponíveis para todos os membros da equipe

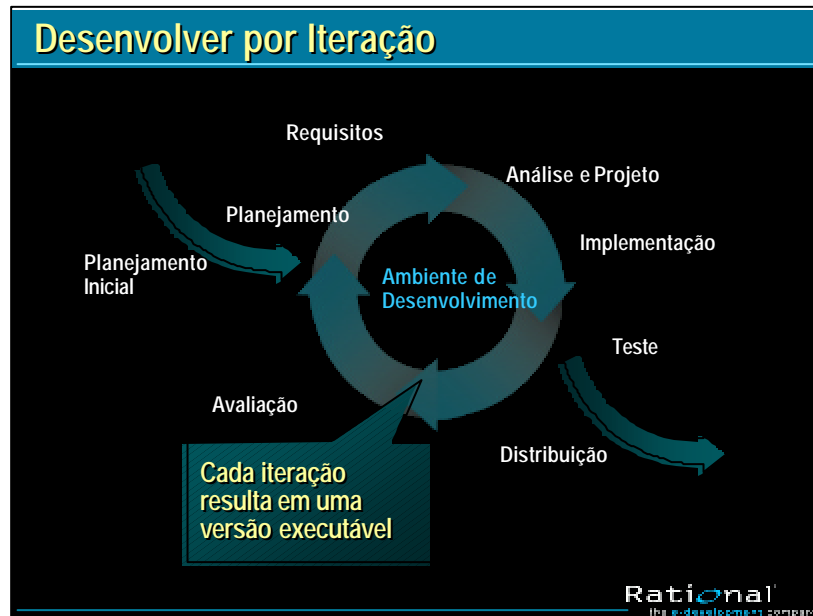
Mau Gerenciamento = CAOS

**Rational**  
the e-development company

As we just identified, we live in a complex world that is constantly changing. Our challenge as software professionals is to incorporate change into our systems. In order to do that we must manage our projects carefully. That means we must have architectures that are flexible, we need to follow a requirements management process that deals with change and we need to effectively communicate between all members of the project team. If we don't we'll have a poorly managed project which eventually will lead to chaos..

# Speaker Support Notes

*Speaker Support Notes*



The first, and arguably most important, best practice toward a Quality By Design solution is to develop your software iteratively. This means that you break up your system development time into small iterations where you make a pass through all development phases from planning to test. At the end of each iteration the project is assessed and the results of the assessment are input to the next iteration.

# Speaker Support Notes

*Speaker Support Notes*

## Benefícios do Desenvolvimento Iterativo

- ✓ Resolve os grandes riscos antes de fazer grandes investimentos
- ✓ Aumenta a qualidade pelo teste contínuo
- ✓ Fornece marcos de referência frequentes, objetivamente verificáveis
- ✓ Fornece feedback contínuo dos usuários a partir do início do projeto
- ✓ Permite a reação rápida aos novos requisitos e às solicitações para mudanças

**Rational**  
the e-development company

There are many benefits to an iterative life cycle. I've listed a few of them here. The bottom line is that you are building quality into your system throughout the *whole* development life cycle allowing you to build better software that is on time.

# Speaker Support Notes

*Speaker Support Notes*

## Perspectiva da Empresa

Desenvolvimento de software: um processo de negócio essencial

### Rational Unified Process

- ✓ Um processo de negócio de desenvolvimento de software de qualidade comprovada e pronto para uso
- ✓ Fácil de adotar
- ✓ Automatizado pelas ferramentas Rational



**Rational**  
the e-development company™

Due to the Internet revolution, software development process is not only important to the success of most companies - it is critical. Whether you have your own process or not isn't the important factor here – it is the existence and adoption of the process. With RUP, we provide you with a proven out-of-the-box development process. This allows you to focus on building your applications, instead of reinventing the wheel and defining process for your organization. However, we realize that many organizations have processes in place already. Some of these organizations may not be up to date regarding iterative development and OO/component based development. If that's the case, you can view RUP as a process framework that can be used in areas where your process is weak. If you already have process in place, RUP is completely customizable to fit into your existing frameworks.

# Speaker Support Notes

*Speaker Support Notes*

## Perspectiva do Profissional

Ferramentas e tecnologia evoluem rapidamente  
É difícil encontrar diretrizes de processo relevantes

### Rational Unified Process

- ✓ Acessível 24 horas x 7 dias
- ✓ Diretrizes, checklists e modelos
- ✓ Integrado com as ferramentas Rational



**Rational**  
the e-development company™

From a practitioner's perspective, the RUP is a tool that can guide you and help you to master the technologies and steps necessary for successful s/w development and deployment. It is always available because it is HTML based; it provides guidelines and specific implementation tips for any particular activity; and, it is integrated directly with all of Rational tools. This helps solve some common problems that today's teams face.

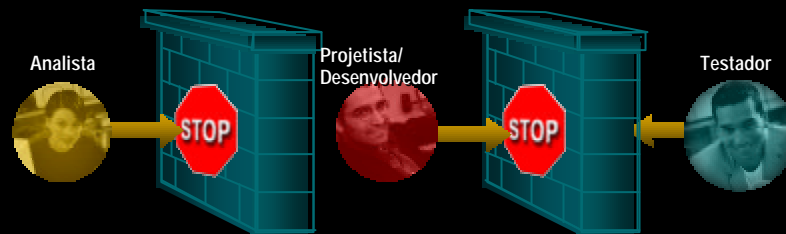


# Speaker Support Notes

*Speaker Support Notes*

## O Problema: Equipes Funcionais Jogam as Coisas por Cima do Muro

Os dados de uma equipe não são utilizados pelas outras



**Rational**  
the e-development company

Let's examine one of those problems:

Team communication is a challenge. What is produced by the analysts is not necessarily useful to the designers so they do it their way. This communication problem leads to communication problems to testers who need to test the system according to the set of requirements that are supposed to be implemented. This goes on and on ... walls everywhere. Do you recognize problems like this in your organization? Software development is all about team communication and situations like this cause delays and mis-matched testing and business requirements.

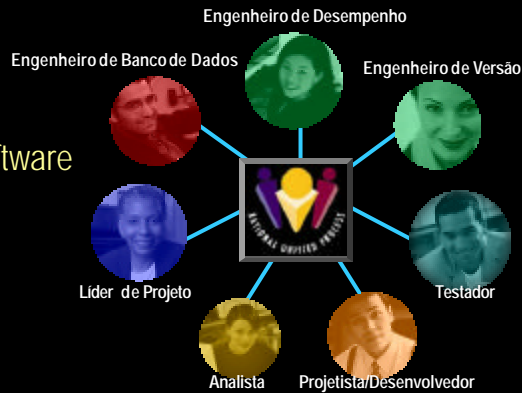
# Speaker Support Notes

*Speaker Support Notes*

## A Solução: Unificar Suas Equipes

Todos os membros de equipes compartilham

- ♦ Conhecimento
- ♦ Processo
- ♦ Visão de como desenvolver o software
- ♦ Linguagem para modelagem



**Rational**  
the e-development company

The power of one common view on software development and test is important. The RUP focuses on team communication. It ensures that the results of the analyst is really useful for the developer and tester. For example, we describe how you produce a use-case that the customer can understand, while at the same time provide the designer enough information to use it to drive the design. The same asset will also allow the tester to create test cases from it, directly. One view makes the team more productive, is easier to work with in distributed teams, and removes the problem of redundant assets. You can reuse items from task to task and user to user. This practice also makes it is easier to get new hires up to speed. How does it do all this...?

# Speaker Support Notes

Speaker Support Notes



The RUP describes *Who* produces *What* , *When* they should produce it, and guidance on *How* to produce it.

Who is described by the concept of a Role that define team member responsibilities. A role is responsible for a number of artifacts. An artifact is something you produce in the project, like a use case, glossary, or project plan. It is the *What*. The RUP also provide detailed step by step instructions on how to produce an artifact and when it should be produced and modified. This is described in the activities. This is the *When* and *How*. The RUP also include guidelines, Word and FrameMaker templates, and checklists that can be used for review meetings. Activities and additional guidelines give all the information you need to produce an artifact.

# Speaker Support Notes

*Speaker Support Notes*



The benefits to your team are quick and dramatic:

Improves team communication - artifacts produced by some are used as input by others so the information will flow between team members

Reduces development risk - RUP is risk driven and focus on identifying and reducing risk early in the project.

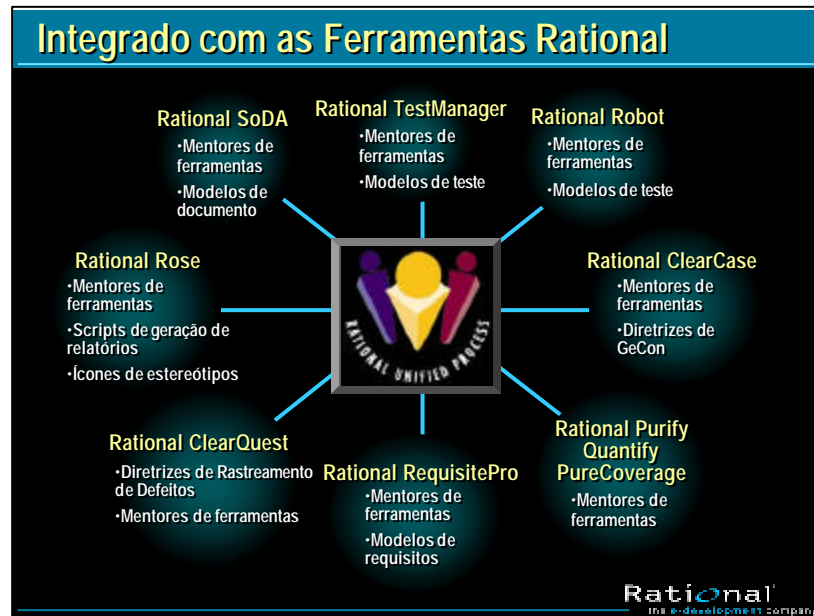
Improved development productivity - With clear defined roles and responsibilities team can work more productive

Improved quality - get requirements right, focus on architecture, use review on design, code etc. Testing can start early in the project

Lowered overall cost of software production - Risk is mitigated early, people do not step on each others toes.

# Speaker Support Notes

*Speaker Support Notes*



Tool support is essential to make process practical, and the RUP is integrated with Rational tools supporting the full software development lifecycle. Let's take a look at the product in a brief demo....

# Speaker Support Notes

*Speaker Support Notes*



See demo script part 1.

# Speaker Support Notes

*Speaker Support Notes*

## Agenda

- ◆ Quality by Design
- ◆ O Problema de Negócio do Desenvolvimento de Software
- ◆ A Solução para o Desenvolvimento de Software
  - Processo Claro e Adaptável
  - Comunicação Completa para Toda a Equipe
  - Gerenciamento Eficaz de Requisitos e Mudanças
  - Casos de Uso e Modelagem de Negócios
  - Teste Unitário Abrangente
  - Teste Iterativo de Sistemas

**Rational**  
the e-development company

OK, that first section talked about process and team communication, specifically addressed by RUP. Next I'd like to talk about requirements and change management – a very important component of quality applications. Again, let's start with a definition to level set everyone.

# Speaker Support Notes

Speaker Support Notes

## Qual É a Dificuldade no Gerenciamento de Requisitos?

### Requisitos:

- ✓ São difíceis de comunicar (palavras podem ser ambíguas)
- ✓ São numerosos (podem ser difíceis de gerenciar)
- ✓ Nunca são iguais (alguns são mais difíceis, mais arriscados, mais importantes, mais estáveis)
- ✓ Estão relacionados entre si
- ✓ São ligados à outras entidades no processo de desenvolvimento
- ✓ Mudam ao longo do ciclo de desenvolvimento

**Rational**  
the e-development company

Requirement. What is it? Simply put, a requirement is some specific thing the application must do. For example, process all transactions in 8 seconds or less – perfectly acceptable requirement. Now you might wonder... what's so hard about this? We defined requirements management and it seems relatively straightforward.

But requirements are more difficult to communicate than you might think. Our natural languages are inherently ambiguous. When I write something down in a spec, you may interpret it completely differently when you read it. In addition, users find it difficult to visualize the system by reading a written spec. They are uncertain whether it will meet their needs. Also, there are often a *lot* of requirements to keep track of. Software is becoming increasingly complex. You could literally have tens of thousands of requirements. Add to this the fact that not all requirements are created equal – some are more important, more difficult to implement, more risky to our project, deliver more benefit, more costly etc. Requirements are more than just text on a page, they have details or attributes.

Want more reasons? OK [click] requirements are related to one another: you might need to fulfill one requirement before addressing another one. And requirements are related to other entities of the development process, e.g., test cases, user models, change requests, design artifacts, project schedules, etc. And here's the real kicker: REQUIREMENTS WILL CHANGE! Change in itself isn't bad; however, unmanaged change is chaotic. Requirements change often and it is very difficult to keep track of requirement evolution and assess the impact on the rest of the system.

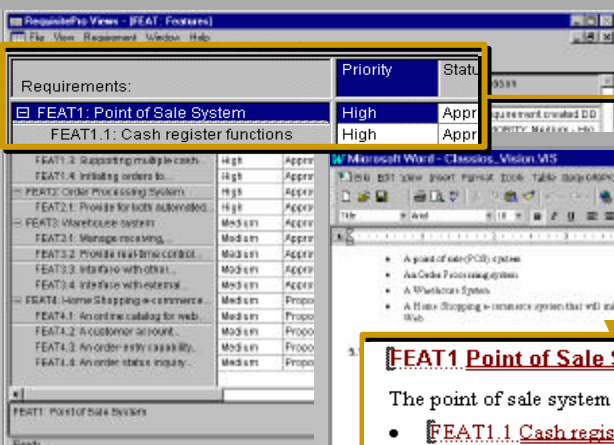
Scared? You should be. Now, let me ask a question: How many people here are engaged in active requirements management today?



# Speaker Support Notes

Speaker Support Notes

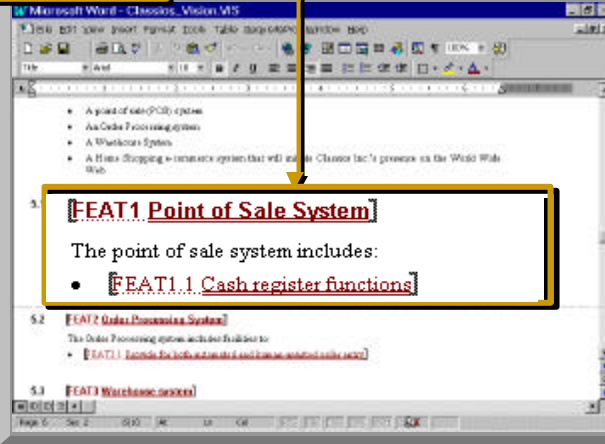
## Capture e Gerencie Requisitos com Eficiência



Requirements:	Priority	Status
FEAT1: Point of Sale System	High	Appr
FEAT1.1: Cash register functions	High	Appr

Classifique e consulte informações

O texto permanece no contexto



Rational  
the e-development company

I think I just proved that requirements need to be documented and organized... ideally in a version controlled database so all changes can be captured and managed, right? Well, the problem is that most of us document information in a word processor like Microsoft Word. Documents are a natural choice to capture and elaborate your requirements because they provide the context to fully understand the requirements – words. The most frequently used word processor is MS Word. By connecting Word to a database system you could get the best of both worlds – the best environment to document free form information and the best environment to organize and manage that information. Well, that's exactly what we do with Rational Suite. We allow you to use the freedom of Word, while delivering to you the power of a database. The tool is called RequisitePro and it come standard with every copy of Rational Suite. It is THE vehicle for documenting and managing requirements, iteratively, throughout your entire software development lifecycle.

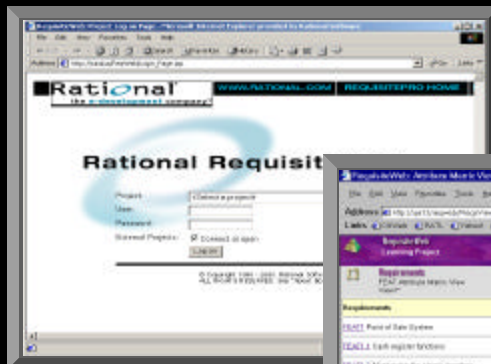
If you are not using Word, you can directly input requirements into the database.

# Speaker Support Notes

Speaker Support Notes

## Gerencie Requisitos via Web

Fácil de acessar



Recursos poderosos

ID	Priority	Status	Case	Difficulty	Stability	Assigned To	Origin
REQ1	Low	Proposed	High	Low	Medium	Patricia	
REQ2.1	Medium	Proposed	Medium	High	High	Patricia	
REQ2.2	Low	Incorporated	Medium	Medium	Medium	Patricia	
REQ3.1	Medium	Proposed	High	High	High	Patricia	
REQ3.2	Medium	Incorporated	Medium	High	High	Patricia	
REQ4	Low	Incorporated	Low	Low	High	Patricia	
REQ5	Medium	Proposed	High	High	High	Patricia	
REQ6	High	Incorporated	Medium	Medium	High	Patricia	
REQ7	Medium	Proposed	Medium	High	High	Patricia	
REQ8	Low	Proposed	High	Medium	High	Patricia	

**Rational**  
the e-development company

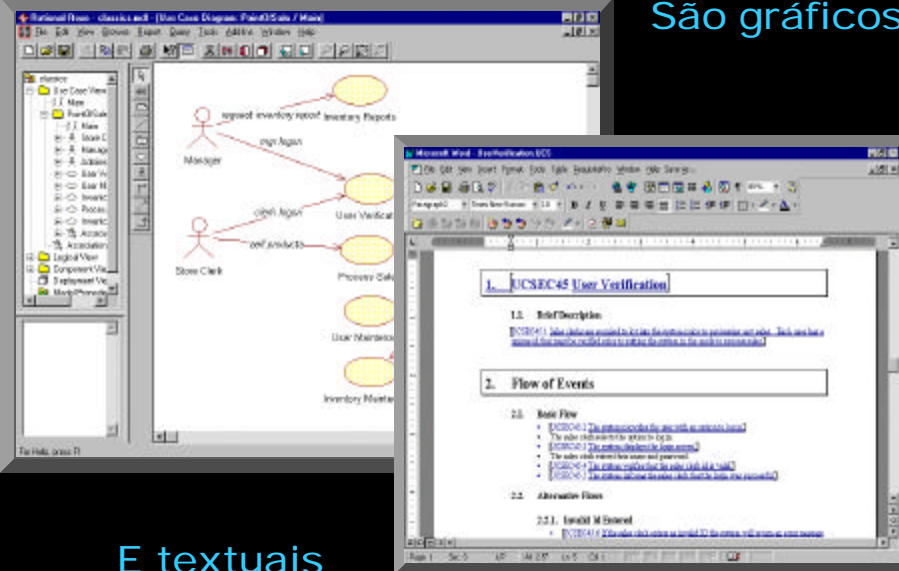
I know what you're thinking – not all requirements come from internal constituents, right? Sometimes customers or partners have very valuable input that we want to capture as input into our next development cycle. Or maybe your team is distributed and you need to practice geographically distributed requirements management. No problem. All the power and flexibility that is available in our client-server requirements management tool is also available from your Web browser with no additional install necessary. Your users (or anyone on your team) can just point and click to create new requirements, modify those that already exist, participate in discussions, or view the impact of change. RequisiteWeb is the most functional Web requirements management tool in the market.

# Speaker Support Notes

Speaker Support Notes

## O Que São os Casos de Uso, Afinal?

São gráficos



E textuais

Rational  
the e-development company

Now that we've discussed the importance of requirements management, let's talk about another element of quality designs – use cases. Like requirements, let's start with a definition. What is a use case? Simply put, a use case is the description of an action a user would take on your application. It is a specific way of using the system from a user's perspective. And a use case can be graphical or textual. The graphical part of a use case is represented in a use case diagram as shown on the left. The simplicity of a use case diagram is what makes it so powerful. The diagram describes interactions between users and actions.

Behind each line between an actor and a use case is an entire dialog – a sequence of transactions. That sequence of actions is typically documented into a document (shown on the right). Most of the use case information is stored in that documents – flow of events (basic and alternate), pre and post conditions, potentially non functional requirements for that use case.

So use cases are **graphical but mostly textual**, which as we know, creates no challenge since automation of such assets are managed automatically with our RequisitePro tool. Since use cases can be graphical or textual (but usually both) you can use either a visual design tool like Rational Rose, a document-based management tool like RequisitePro, or ideally – both. In fact, you can even link the visual description of the use case in Rose to an underlying text document, fully managed and version controlled by RequisitePro – both included as part of Rational Suite.

# Speaker Support Notes

Speaker Support Notes

## O Poder dos Casos de Uso

### Poupe tempo e reduza os erros

- ◆ Obtenha acordo com os clientes no início
- ◆ A equipe tem uma compreensão comum do sistema a ser construído
  - Uma descrição evita a redefinição de requisitos
  - Não-técnico o bastante para que os usuários entendam
  - Detalhado o bastante para que os desenvolvedores comecem a projetar
- ◆ Se expandem pelo projeto, teste e documentação
- ◆ A natureza centrada no usuário reduz as restrições de projeto

**Rational**  
the e-development company

What is the real power of use cases? Use cases save us time and help reduce errors early on – and we need to do everything we can to reduce these errors early on. How do use cases save time and reduce errors?

First off – because they use natural languages to describe the system from a user-centric view, use cases help define the solution in terms that less technical stakeholders can understand – and often these less technical folks happen to be the ones holding the project budget or doing the project testing. By providing a way to express requirements and user activity within a managed context we achieve the best possible communication mechanism between any two parties. By gaining agreement early on, you stand a chance of avoiding expensive reworks down the road because your stakeholders sent you the wrong path by not understanding the requirements early on.

Second of all, use cases provide a common understanding of the system to build. They provide a single description of the system to build, rather than using a user requirements document and a system requirements document, each for different audience, and one being restated from the other. Use cases are easy enough for non-technical folks to understand the solution we are about to build, yet they are technical enough to designers to understand the desired functionality. Plus, use cases are the start of the **design** activities. By describing the functional requirements in use cases, we automatically jumpstart designers who no longer have to convert a set of shall statements into a set of use cases to start designing the solution.

Finally, because they tend to treat the system as a black box (by following a user-centric approach to describing the system functionality), they focus on WHAT the system should do rather than HOW it should do it... again, a design activity.

# Speaker Support Notes



Now, we have talked about many of the virtues of use cases; however, here is the most valuable piece – they are reusable. 100% reusable, and that saves time and money in the long run. How? Well, in the traditional approach, the product manager would come up with the initial requirements, and describe it in a product requirement document (PRD). After prioritization, product requirements would then be refined into a Software Requirements Specification, or product spec. The information in the spec will be restated by the documentation writers in a user manual, the test engineers will restate the SRS in a format that is useful for testing the system, etc., etc.. Each time the requirements are restated, information is lost and the intent of the design is compromised

Let's contrast this with the use case driven approach.<mouse click>

In the use case driven approach, the product manager would come up with the initial use case model, one in which use cases and actors would be described. After prioritization, use cases scoped in the release would then be refined into a use case specification. That use case specification can then be reused by both doc writers and testers since it is written in a sequential fashion, just like user's manuals and test scripts. Plus, the use case has immediate impact because it is graphical (easy access for testers, esp. non-technical testers) and textual (doc value).

Use cases provides a single way to describe the requirements that everyone can be jumpstarted with.

# Speaker Support Notes

Visualize o Impacto da Mudança

Rastreabilidade garante cobertura

Detecção de mudanças em tempo real

Rational  
the e-development company

Traceability is a key facet to quality. You want to be able to see the relationships between related requirements or requirements and use/test cases. In the screen shot shown, traceability views show the relationships between the **product features**, across the top, and the **use cases** that fulfill those features down the left. The blue arrow indicates these items are related. But what happens if a change has occurred in the text of the requirement in the document?

[CLICK]

The link between related items becomes "suspect", warranting a closer look. The red slash over the arrow indicates a *suspect link*. This mechanism has been designed to appear simple to the user, but in fact, it is extremely powerful when used to assess the impact of a potential change and communicate changes to your team.

# Speaker Support Notes

## Speaker Support Notes

**Plano de Teste: Requisitos e Casos de Uso como Entradas**

- ◆ **Entradas do Teste**
  - Qualquer coisa de que o testador dependa para conduzir o teste
  - Não limitado a artefatos Rational, podendo incluir:
    - Requisitos (ReqPro)
    - Casos de Uso / Modelos de Diagramas de Sequência (Rose)
    - Células de Planilha
    - Solic. de Mudança / Defeitos
    - ? "Entrada Personalizada"

Aberto a todos os tipos de entrada  
Permite rastreabilidade mais flexível



**Rational**  
the e-development company

So great, we've talked about requirements and use cases, but how do you map that to your test plan? Well, it's easy. A Quality By Design solution depends on re-use. Requirements and use cases are assets you generate early and manage throughout your project.... so why duplicate any of that information when you go to test against it. A good QBD solution allows for immediate and unfiltered reuse of assets. Rational TestManager, the test planning component of Rational Suite, accepts inputs directly from any source. There are built-in adapters for requirements and use cases generated from Rose and RequisitePro, so you can simply reference those assets as test inputs and TestManager handles the rest. You can associate test case to those input and generate coverage and distribution reports to show where your test plan stands with respect to requirements or use cases at any point in time.

And by the way, if you want to use a custom input, e.g., Excel spreadsheet cells, we have an open API that lets you integrate directly to TestManager. But more about that later....



# Speaker Support Notes

## Speaker Support Notes

### Rastreabilidade entre Casos de Uso e Casos de Teste

- ◆ Os casos de uso podem ter um caso de teste associado
- ◆ Em geral, há um relacionamento de um para muitos entre casos de uso e casos de teste
- ◆ Mudanças nos casos de uso geram mudanças nos casos de teste

	Test Case 1	Test Case 2	Test Case 3	Test Case 4	Test Case 5
Use Case 1	↗				
Use Case 2		↗		↗	
Use Case 3			↗	↗	
Use Case 4			↗	↗	

**Rational**  
the e-development company™

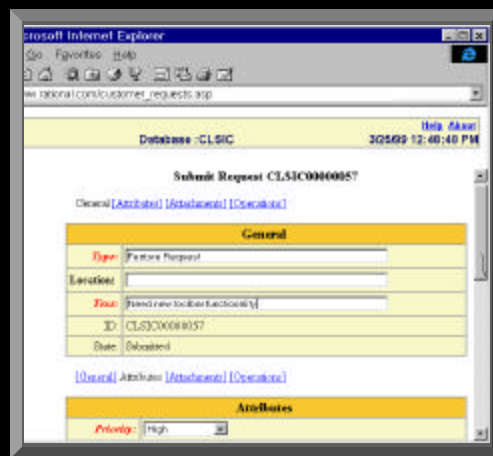
Like the traceability you had between use cases and features or requirements, you should also consider tracking relationships between use cases and test cases. After all, a use case gives you a description of what activity to follow – why not verify that the test case used to verify that use case has passed? Also, you can see if one use case translated to multiple test cases, and vice versa.



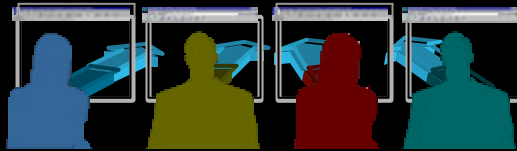
# Speaker Support Notes

Speaker Support Notes

## Colete as Mudanças pela Web



- ◆ Alcance todas as fontes de solicitações
- ◆ Simplifique a classificação e a priorização de solicitações
- ◆ Garanta que os requisitos reflitam precisamente as necessidades dos envolvidos



**Rational**  
the e-development company

Now let's talk for a moment about change. I addressed a moment ago, the importance of reusing assets. This applies here, too. Changes comes in many ways and in many formats, e.g., email, voice mail, surveys etc. Managing these change requests can be a very time consuming task. And if you do it manually, it is prone to error.

One way to save time and promote quality is to provide a consistent and complete change request form and make that form widely available to your stakeholders. Doing so lets you collect the information you need to make decisions, prioritize requests & defect, etc. Rational Suite provides a consistent, easily accessible approach to gather and manage all the input via the web – a now universal interface that can be used by all your stakeholders, internal and external. You can setup access from a password-protected URL. Once they click on the submit button, their input is funneled into a centralized change request database. Again, easy capture and re-use of assets is the key!

# Speaker Support Notes

Speaker Support Notes

## Gerencie as Solicitações de Mudança

Consistente

Enviado por um membro da equipe

Enviado por um usuário

Facilmente Gerenciável

Classif. e Consultar

#	general_text	priority	State	assigned_to
	Need request for functionality	High	Submitted	alex
	Needs needs to accommodate groups	High	Validated	chris
	Need support for Windows 2000	High	Rejected	chris
4	CLSTC00000093 Implement controls as ActiveX	Medium	Submitted	alex
5	CLSTC00000092 Testing must plan for 1000 users	Medium	Submitted	alex
6	CLSTC00000091 Test for all database backends	Medium	Submitted	alex
7	CLSTC00000044 E-commerce implementation issues	Medium	Proposed	alex

Rational  
the e-development company

But it goes even further. Not only can you establish a hyperlink from your web site allowing customers to submit change requests, but this technology is also used by your developers, testers and help desk to collect input. This ensures you will be collecting all of the change requests in real-time, in a similar format, which will make you gain time at reviewing it all. From the database where change requests are logged, you can prioritize, review, sort, separate, and make assessments – quickly – with all the information at your easy disposal.

This practice helps you control scope and feature creep by putting *you* in control of changes.

# Speaker Support Notes

Speaker Support Notes

## UCM Aprimora Comunicação e Eficiência da Equipe

### ◆ O gerente de projeto é capaz de

- Executar métricas de projeto
- Equilibrar cargas de trabalho
- Comunicar altas prioridades

### ◆ Todos na equipe sabem

- Quais são as atividades de alta prioridade
- Quais atividades estão sendo trabalhadas



**Rational**  
the e-development company

As I mentioned earlier, work in RUP is organized around activities. Activities have details attached to them such as priority, who's working on them, state, etc. It is much easier for everyone on the team to know what they're supposed to be working on, as well as what's going in the project as a whole – big picture / little picture.

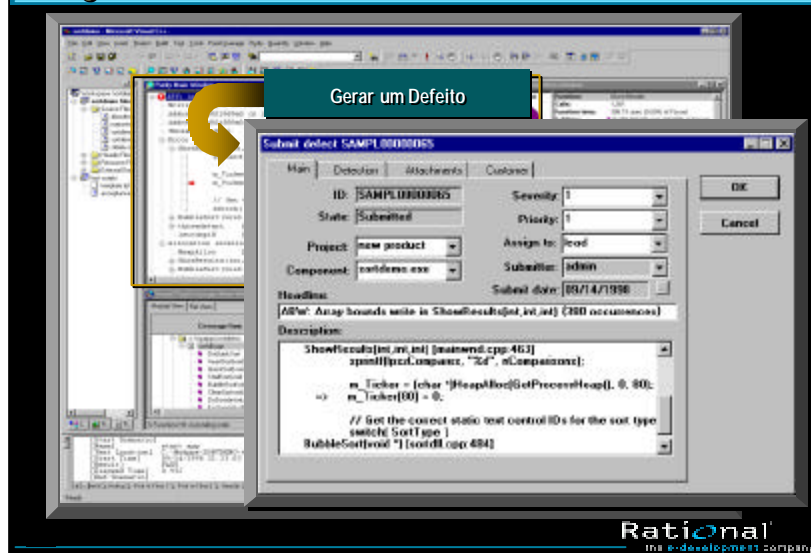
With UCM, Unified Change Management, project managers can better analyze projects, workloads, and activities and communicate cross-functional priorities. Plus everyone on the team knows what are a project's priorities, what is being worked on by the group and which specific activities are associated with a baseline. Through the powerful reporting and charting capabilities of ClearQuest (the tool you saw on the last few slides), team leads can easily review project status, assess and distribute workload and estimate completion dates.

Working with ClearQuest and ClearCase, UCM sets up each individual's personalized activity-based to do list. So, whatever your role on the team, you've got a clear understanding of the project's scope, priorities, and all of the activities associated with the project.

# Speaker Support Notes

## Speaker Support Notes

### Integração ao Ambiente de Desenvolvimento

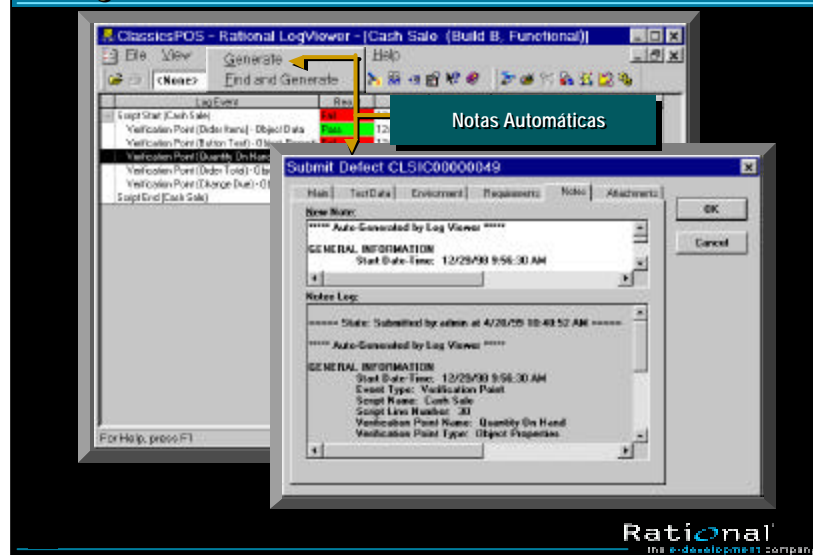


Finally, some specific change control talk for developers. What if you could integrate your change management system directly into your IDE? Wouldn't that save time and effort – and again allow you to reuse assets and increase efficiency? Sure it would. If you're debugging code or running code for unit testing and come across a defect, it would be great if there was a tool watching that could automatically trap those errors and populate a defect report for you. That's exactly what tools like Purify do.... And yes, such tools are included in Rational Suite. Shown here is a screen shot of the Purify integration with Microsoft VisualStudio and ClearQuest. Notice how your developers work directly in their IDE, no need to learn a new interface. Also notice how the ClearQuest fields are automatically filled in with the Purify generated information.

# Speaker Support Notes

*Speaker Support Notes*

## Integração com o Rational Suite TestStudio



And here is a screen shot of the integration with our test environment. Notice how easy it is to generate and log a defect directly from your test log. All the information you need to re-create this defect is automatically entered and attached, saving time for developers and testers.

# Speaker Support Notes

*Speaker Support Notes*



Let's look specifically at managing change requests and the requirements of your application.

See demo script part 2

# Speaker Support Notes

*Speaker Support Notes*

## Agenda

- ◆ Quality by Design
- ◆ O Problema de Negócio do Desenvolvimento de Software
- ◆ A Solução para o Desenvolvimento de Software
  - Processo Claro e Adaptável
  - Comunicação Completa para Toda a Equipe
  - Gerenciamento Eficaz de Requisitos e Mudanças
  - Casos de Uso e Modelagem de Negócios
  - Teste Unitário Abrangente
  - Teste Iterativo de Sistemas

**Rational**  
the e-development company™

OK – next let's move on to talk for a moment about modeling.

# Speaker Support Notes

Speaker Support Notes

## Modele Visualmente

- ◆ A Modelagem Visual lhe permite construir certo na primeira vez:
  - Compreenda os requisitos do usuário
  - Garanta que o projeto atende às necessidades
  - Visualize a interface do usuário, a lógica do negócio e os dados separadamente
  - Separe domínios de negócio como apropriado
  - Visualize todas as dependências
  - Garanta o desempenho antes do início da codificação



**Rational**  
the e-development company

Visual model is another best practice identified in RUP. And there are two main benefits of modeling:

- The first is communication. Visual models can be used throughout the life cycle as a communication tool by everyone involved in your project... from customers to analysts to data analysts to requirements analysts to designers, developers, project managers, testers and technical writers. Everyone will know the state of the project. Just like putting together a model of car or laying out office space, modeling software applications help you accelerate the process by removing on-the-fly design changes. Modeling early and then constructing your application will improve the quality you are able to design into your system.
- The second benefit is that visual modeling makes you plan out your system before you build it. In other words. Modeling helps you build your system right the first time.



# Speaker Support Notes

*Speaker Support Notes*

## Automatize com Ferramentas

- ✓ Melhora a produtividade humana
- ✓ Elimina as fontes de erros humanos
- ✓ Permite aprimoramentos no processo



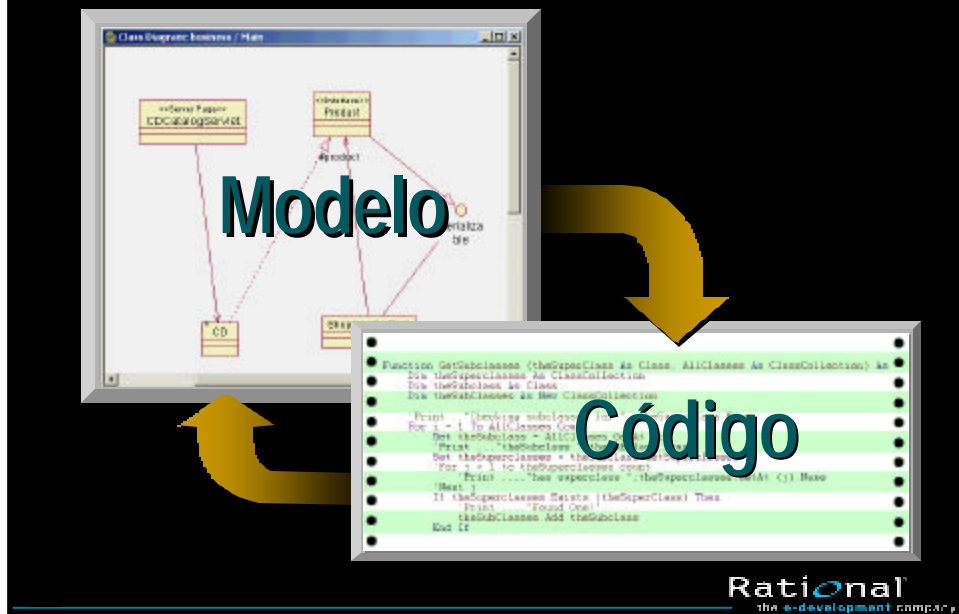
**Rational**  
the e-development company™

... and whenever you can automate a process with tools, you have the opportunity to improve human productivity. Removing the tedious, repetitive tasks will enable your team to spend more time thinking, planning, and engineering – and less time doing manual-intensive things that have little pay-off and high costs. You also remove the human error factor, which is bound to creep into any human-intensive process.

# Speaker Support Notes

Speaker Support Notes

## Exemplo: Geração Automática de Código



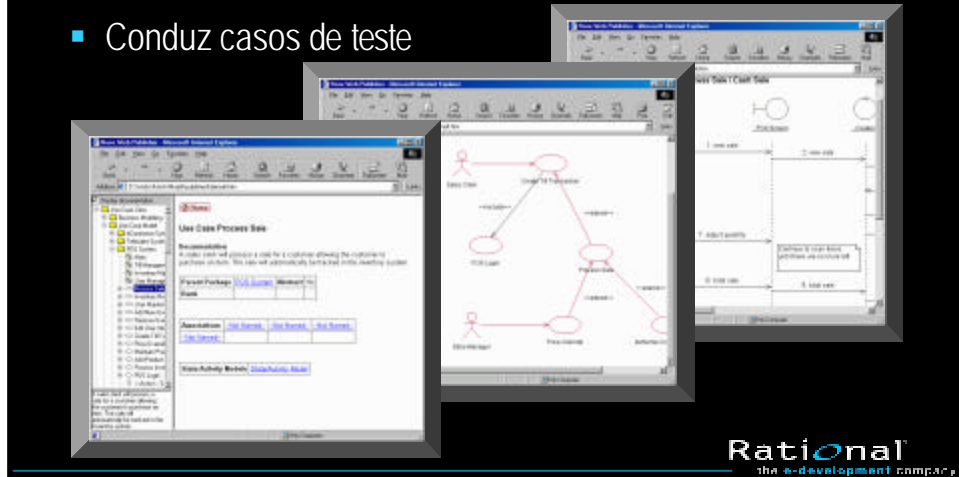
An example of how automation can help you is through code generation from design models. This allows you to create a blueprint of your application working on a higher level of abstraction, and then automate the creation of a large number of code lines in a desired language based on your design. Many lines of code are hence generated from a fewer number of human generated specifications (blue prints).

# Speaker Support Notes

Speaker Support Notes

## Comunique a Funcionalidade

- ◆ Verifique as necessidades dos usuários - Web publishing
  - Captura requisitos sem ambigüidades
  - Destaca erros e omissões
  - Conduz casos de teste



Once we have verified the basic functionality of the system, we then need to communicate the functionality of the system. Again, calling on a web-based interface gives you the benefits of pervasiveness while re-using as many existing assets as possible. Here, for example, using the web publishing capabilities of Rational Rose you can publish descriptions of the system, use cases, and activity diagrams that will help your test team by documenting the precise steps and actions they need to validate.... Thus driving your test cases from existing material and avoid costly rework where you might lose something in the translation. Finally, this also allows you to get "buy in" from all the project stakeholders assuring you that you are building the system that they want.

# Speaker Support Notes

*Speaker Support Notes*

## Agenda

- ◆ Quality by Design
- ◆ O Problema de Negócio do Desenvolvimento de Software
- ◆ A Solução para o Desenvolvimento de Software
  - Processo Claro e Adaptável
  - Comunicação Completa para Toda a Equipe
  - Gerenciamento Eficaz de Requisitos e Mudanças
  - Casos de Uso e Modelagem de Negócios
  - Teste Unitário Abrangente
  - Teste Iterativo de Sistemas

**Rational**  
the e-development company™

# Speaker Support Notes

Speaker Support Notes



As we discussed earlier, reusing as much work as possible saves time and improves efficiency. If you're building component-based systems (and most companies are moving in this direction), you need to validate functionality of those components. But how do you do that? Components generally have no GUI. Plus, they often need to interact with other components that may not yet be finished.

I don't think anyone would argue that component testing helps mitigate risks early in the development cycle. Problems can be found and fixed while the component is still isolated, thus resulting in a higher predictability of an integrated system; unfortunately, component testing is easier said than done... in fact, up to 50% of a developer's time can be spent creating and maintaining drivers and stubs just meant to test a component – and most of this work gets tossed.

**Rational** QualityArchitect helps you ensure the quality of your applications by testing software components quickly and easily. It

# Speaker Support Notes

Speaker Support Notes

## Garanta a Confiabilidade do Software: *Rational Purify*

- ✓ Funciona com ou sem fonte para que até os componentes de terceiros possam ser testados
- ✓ Funciona dentro de processos existentes
- ✓ Resultados rápidos porque não é preciso recompilar



Indica os bugs difíceis de localizar

**Rational**  
the e-development company

QualityArchitect validated functionality in your design. Now let's look at reliability. How many times has an application executed fine on one machine and the minute it is moved to a testers machine it crashes? All the time, right? And unfortunately, trying to manually find runtime errors like these is next to impossible. Fortunately, Rational Purify helps you find reliability defects which can cause application instability and often trigger unpredictable failures. With automated debugging, Purify detects, locates and diagnoses run-time and memory-related bugs in C/C++ code and garbage collection-related memory management issues in Java code.

# Speaker Support Notes

*Speaker Support Notes*

## Teste Todo o Código : *Rational PureCoverage*

- ✓ Garante a qualidade do pacote de testes revelando caminhos de código não exercitados
- ✓ Funciona com ou sem fonte para que até mesmo os componentes de terceiros possam ser testados
- ✓ Funciona dentro de processos existentes
- ✓ Resultados rápidos porque não é preciso recompilar

Código não testado antes da liberação  
será "testado" pelo seu cliente

**Rational**  
the e-development company

Speaking of testing your code, you need to make sure that you test **all** of your code, because I guarantee that your customers will find the one line of untested code in your application. Finding out what code has been tested and what code has not been tested is a tedious process. This is why we include coverage tools in our suite, too. Rational PureCoverage provides code coverage verification and shows developers how much of their code is exercised during a testing session. Even a test run that is 100% successful is not a valid metric of software quality, unless you know that your test run actually touched all of your code. PureCoverage not only provides this information, it actually shows the lines or functions not called. It integrates automatically with the IDE which gives you fast, reliable, automated access to code coverage information.

# Speaker Support Notes

Speaker Support Notes

## Isole Problemas de Desempenho: *Rational Quantify*

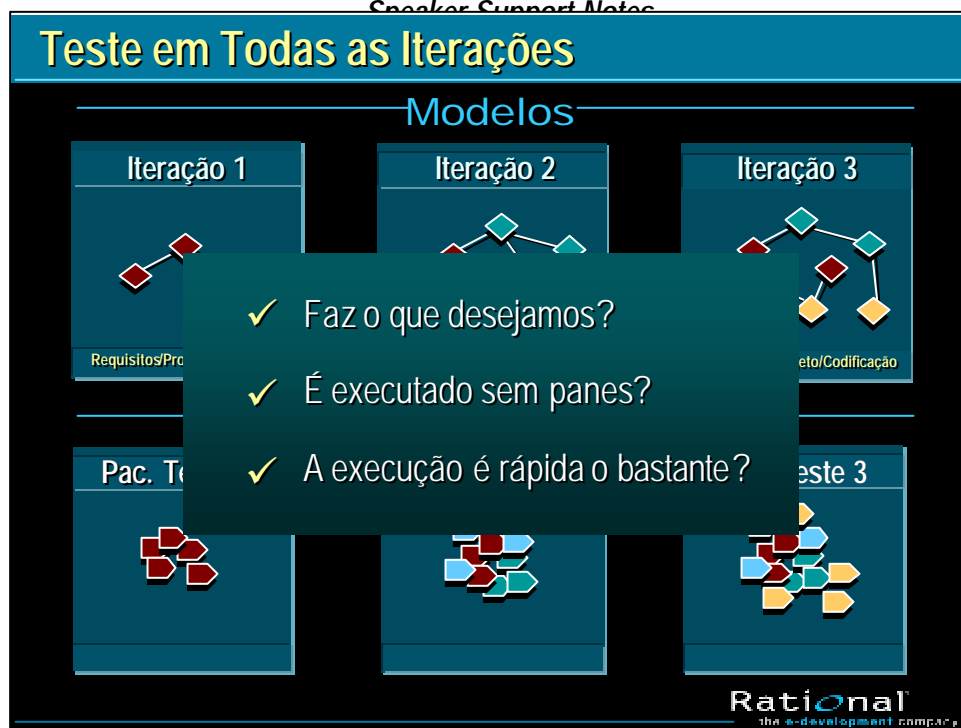
- ✓ Destaca visualmente os gargalos de desempenho e indica sua gravidade relativa
- ✓ Funciona com ou sem fonte para que até mesmo os componentes de terceiros possam ser testados
- ✓ Funciona dentro de processos existentes
- ✓ Resultados rápidos porque não é preciso recompilar



A final quality measure during development activity is procedure performance. You can have very reliable code that is fully tested but if it runs so slowly that even you won't use it, you can be guaranteed that your customer won't use it. Rational Quantify automates the process of isolating application performance bottlenecks for you. It not only pinpoints the bottlenecks in your code but also measures the severity of each, allowing you to decide what to work on first, and what to let go by. If it is an area of code that is infrequently accessed - fixing it may not be justified. Rational Quantify helps you make the decision.



# Speaker Support Notes



So throughout the development cycle, testing early and often is quite painless – yet the benefits are tremendous. Higher quality applications, delivered in shorter time. To maintain quality by design throughout the development lifecycle, and throughout the life of the software, you need to test every iteration for functional compliance, performance, and reliability.

# Speaker Support Notes

*Speaker Support Notes*



So let's take a look at some of the tools we were just talking about...

See demo script part 3

# Speaker Support Notes

*Speaker Support Notes*

## Agenda

- ◆ Quality by Design
- ◆ O Problema de Negócio do Desenvolvimento de Software
- ◆ A Solução para o Desenvolvimento de Software
  - Processo Claro e Adaptável
  - Comunicação Completa para Toda a Equipe
  - Gerenciamento Eficaz de Requisitos e Mudanças
  - Casos de Uso e Modelagem de Negócios
  - Teste Unitário Abrangente
  - Teste Iterativo de Sistemas

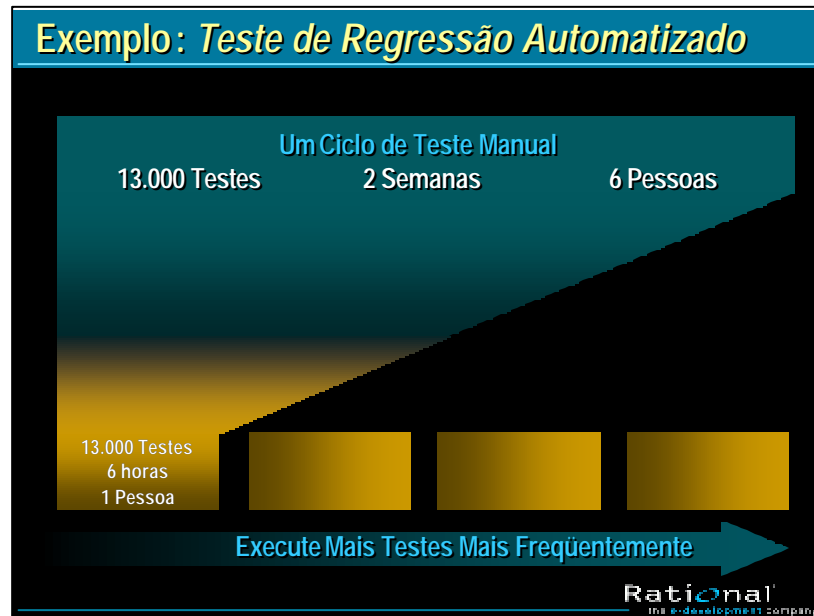
**Rational**  
the e-development company

You may be surprised that we've been talking for 1-1/2 to 2 hours about quality by design, yet we haven't yet mentioned traditional test activities! This should be obvious to you know, but what that means is that quality by design has to do with ALL parts of the software development lifecycle – not just testing.

But it is now time to talk about a more traditional testing approach, so let's get to it.

# Speaker Support Notes

*Speaker Support Notes*

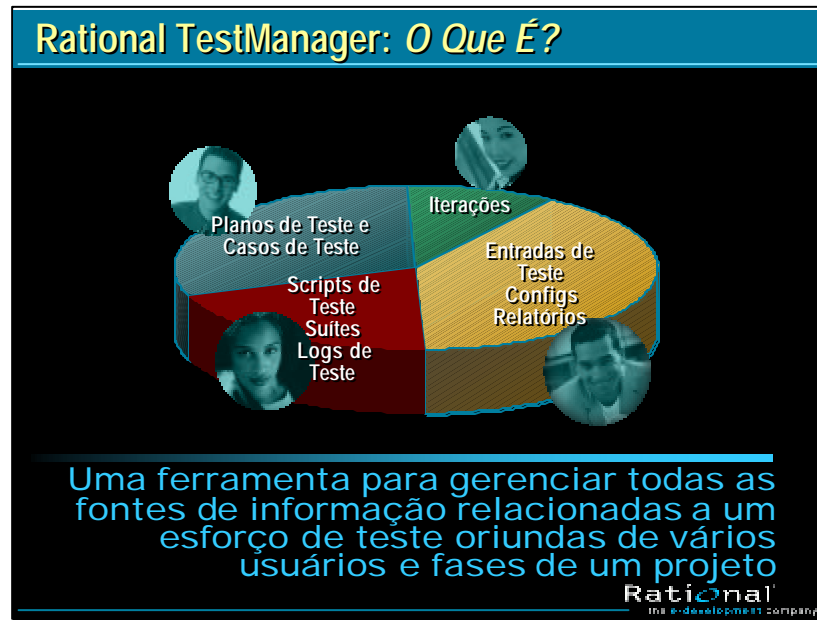


Manual testing is still the most prevalent type of testing today. And the fact is that many tests simply cannot be automated. Automating what you can – smartly – is what is going to generate a return on investment for you and accelerate your testing cycle and quality.

Here is a case of automation ROI: In one benchmark, a manual approach to testing a GUI-intensive application, one test cycle took 6 people 2 weeks to run, making iterative testing difficult, since their release cycle was weeks apart. With automation, this became less than one person days effort, making regression testing of iterations practical and cost effective.

# Speaker Support Notes

*Speaker Support Notes*



To compensate for these challenges, we built a tool called TestManager – included with every Rational Suite. Simply put, TestManager is a tool with which to manage all sources of information related to a testing effort ... regardless of which users or phases of the project generated that information.

# Speaker Support Notes

## Speaker Support Notes



TestManager was built to explicitly support the 4 major activities performed in test:

- 1) Defining inputs/requirements
- 2) Test planning, including lab management, test design, etc.
- 3) Execution – where you actually implement the tests you planned
- 4) Analysis – where you evaluate the tests, generate defects, etc.

Let's make a quick definition of each:

**INPUTS:** A test input, simply put, is the definition of something that needs to be tested or evaluated. It might be a business requirement, e.g., all orders must be processed in 8 seconds or less; it might be a product spec, use case, etc. – all of these are typical test inputs. Now, we have built-in support for input types, e.g., RequisitePro and Rose. However, if you need to use custom inputs we publish the API so you can integrate literally ANY tool and pump output from that tool into TestManager as a test input.

**PLANNING:** The command center of your test management. It is organized around your test plans, assets (like scripts), machines, configurations, etc. This is where you choose what gets tested, where it gets tested, and which data to use in those tests. It is also where you explain how tests get implemented.

**EXECUTION:** Speaking of implementing tests, this is exactly what execution is about. This is where you deploy and execute scripts. More on this in a moment. Just like in the input phase, we publish the API here, too, so you can use any test execution engine you wish to implement your tests.

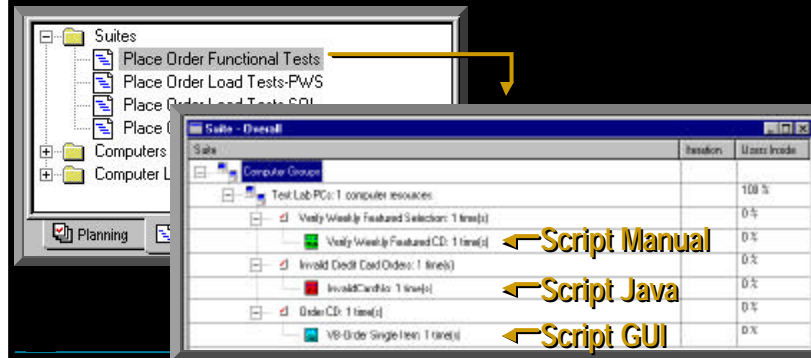
and finally, **RESULTS:** This focuses on evaluating the results of your test run and reporting on those results and test logs.

# Speaker Support Notes

*Speaker Support Notes*

## Arquitetura de Execução de Testes Aberta

- ◆ Permite a execução de uma suíte com vários tipos de scripts (manual, GUI, carga, etc.)
- ◆ Pode criar uma suíte para execução com coleta de recursos
- ◆ Recursos robustos para programação e modelagem de perfis



Once of the nicest features of TestManager is the ability to define, execute, and track any type of test – functional, load, manual, integration, unit, etc. In the example shown here, we run a manual test, Java script, and GUI (or automated functional) script in the same test run. TestManager handles scheduling, execution, and collection of results for every test that gets run under its command. If you had a 3<sup>rd</sup> party test tool you could even run that here and TestManager would collect and collate the results.

# Speaker Support Notes

## Teste de Objetos

Navegação precisa e confiável em objetos (até mesmo guias, árvores, ...)

Teste todos os dados, até os ocultos (ex.: em tabelas Sun Swing J)

Todas as propriedades de objeto disponíveis p/ o desenvolvedor são testadas em tempo de execução

Let me spend a moment talking about functional testing and the importance of object-based testing.

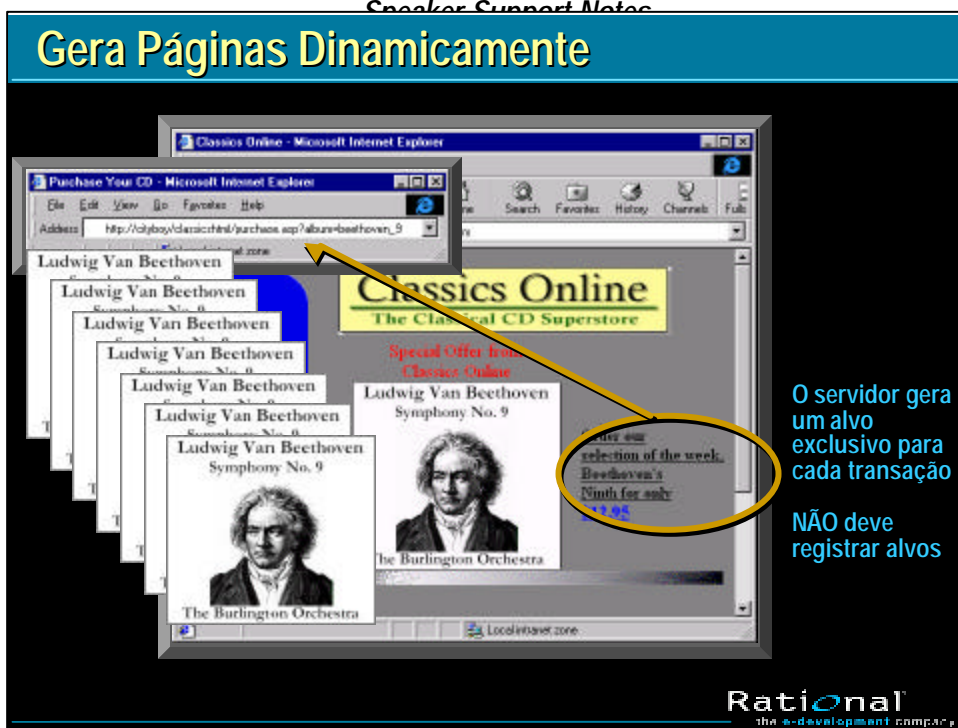
As you know, applications change from build to build. Objects change. They get prettied up, moved about, updated dynamically, and in an n-tier system, their timing is pretty unpredictable anyway. This means that it really isn't possible to reuse scripts recorded by X-Y position, label, or with fixed timing. That's why Rational pioneered a technology called Object



# Speaker Support Notes

Speaker Support Notes

## Gera Páginas Dinamicamente



O servidor gera um alvo exclusivo para cada transação

NÃO deve registrar alvos

Rational  
the e-development company

And here is an example with Dynamic HTML. If you try to test transactive content, something like Amazon, what happens when you place 10 orders? You get 10 books, right? Well, if you do it the way most products test this site, you get 10 copies of the same book. That's because they record the thing you ordered as the first URL. Is that what you wanted? Probably not. If you've got content that is customized based on who the user is, for example, your target should be different each time. Thus, you should NOT record targets.

# Speaker Support Notes

**Teste Dinâmico: A Maneira Correta**

Object Properties Verification Point

Selected object: HTMLLink:HTMLid=hrefSpecialPurchase

Name	Value
hrefSpecialPurchase	hrefSpecialPurchase
innerHTML	Order our CD's...
innerText	Order our Beetho...
nameType	Active Server Docume...
namePop	purchase.asp

Deve registrar origem do link por nome persistente

Permite que o servidor use alvos reais no momento da execução

Rational  
the e-development company

Instead, you should capture the persistent object name, so if you run this test with 10 different users, you get 10 different books. We allow this in Robot because we use the W3C's document object model to look at the object that generates the click, just like the web site does when you actually make the order.

When you test with Rational, you get the 10 books you ordered. Only Rational does this, and it's the benefit you get with object testing.

# Speaker Support Notes



We talked in the last section about the value of tools like Purify and Coverage for tapping memory leaks, GPF's, code coverage, etc. Well, these tools are seamlessly integrated with Robot, the functional testing tool. In fact, it takes functional testing one step further – letting you do things like identify missed lines of code or memory leaks when you're playing back your normal set of regression tests! By simply turning on one of these tools in the background – something like Purify for memory leaks and GPF's or PureCoverage for code coverage – you can collect valuable information usually reserved only for developers.

Using Rational's tools, you can instrument an application prior to performing functional tests.

Once instrumented you now have the capability to generate reports that will tell you what percentage of code is covered during execution of your functional tests. You can, for example, collect metrics on how effective your test case is at exercising a certain section of code. You can also tell if the "place order" test missed the lines of code that check the credit card validation. That would be valuable information to capture, right? Well, it's just a click away on your options tab. This ensure that no major areas of the application are going un touched in your test plan.

# Speaker Support Notes

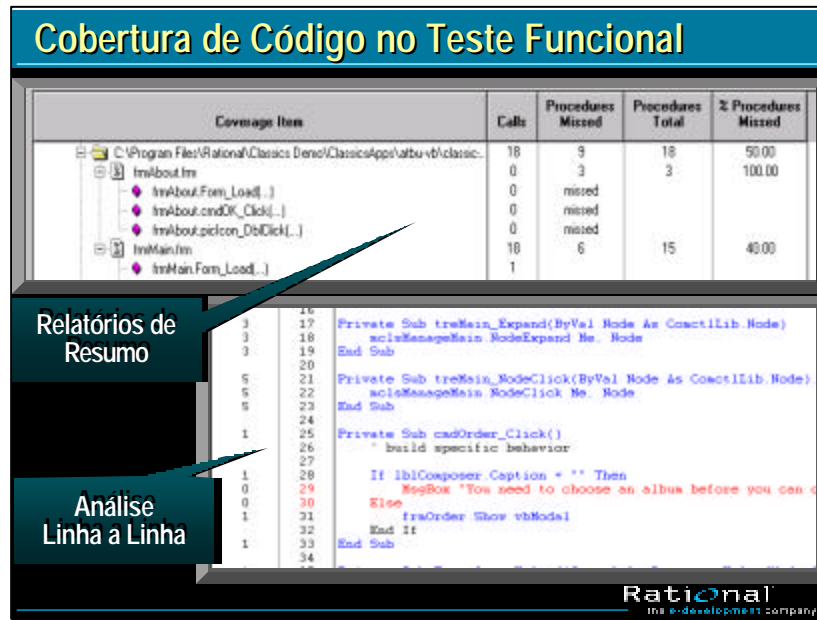
## Speaker Support Notes

Teste Funcional e de Confiabilidade	
Classics Online - Rational LogViewer - [Place Order VB with Cov	
File View Select Tools Window Help	
<None>	
Log Event	Result
Script Start (Place Order VB with Cov)	Fail
Start Application	Pass
Unexpected Active Window - Send (ESCAPE) Key	Fail
Purity Message: I Starting PurityAd ClassicsA.exe at 07/05/2000 16:48:53	Informational
Purity Message: I Starting main	Informational
Purity Message: E ABR: Array bounds read in SetDIBits (1 occurrence)	Fail
Purity Message: E ABR: Array bounds write in snxprintf (1 occurrence)	Fail
Purity Message: I MAF: Memory allocation failure of 1049576 bytes in mjpeg40.dll (1 occurrence)	Informational
Purity Message: E ABR: Array bounds read in SetDIBits (1 occurrence)	Fail
Purity Message: I Searching for all memory leaks...	Informational
Purity Message: W ILK: COM Interface leak from interface pointer 0x12680a08, allocated in SigCreateDocfile	Warning
Purity Message: W ILK: COM Interface leak from interface pointer 0x77bca228, allocated in CoGetMalloc	Warning
Purity Message: W MLK: Memory leak of 784 bytes from 0x00000000, allocated in LoadExpSrvDll	Warning
Purity Message: W MLK: Memory leak of 784 bytes from 0x00000000, allocated in DICanUnloadNow	Warning
Purity Message: W MLK: Memory leak of 116 bytes	Warning
Purity Message: W MLK: Memory leak of 784 bytes	Warning

As you can see here – the information is all captured in a single log file. When you log a defect off this log, all that valuable information is included automatically – the functional errors found in regression testing PLUS the reliability errors that you get for free... just by making that one menu pick and turning on one of these tools.

# Speaker Support Notes

## Speaker Support Notes



Here's a screen shot showing the kind of information you can extract from our code coverage tool. Wouldn't it be great to say to your development staff, "Here is the code coverage that was detected while I was running my regression suite and found this bug?" The test team doesn't have to interpret this information – simply pass it on to your development staff and your value is increased dramatically because you've made THEIR jobs easier by giving them something that shows them where to look in their code for the problem you found. Quality by design.

# Speaker Support Notes

**Programação LoadSmart**

- ◆ Fácil de Usar
  - Modele cenários complexos
  - Programe por "apontar e clicar"



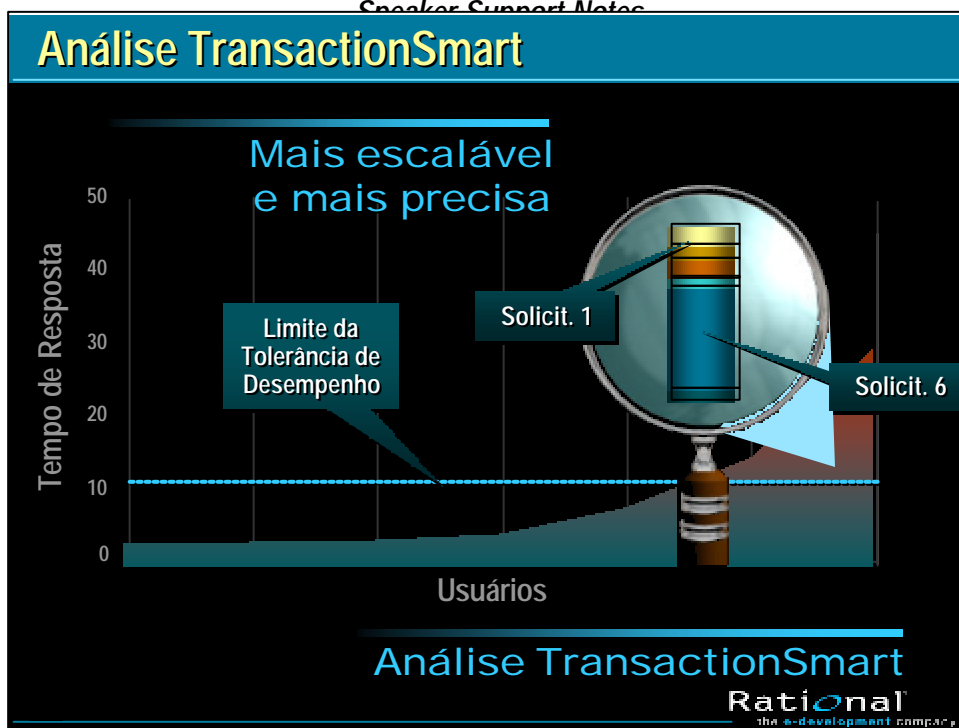
**Programação LoadSmart**

**Rational**  
the e-development company™

OK, let's transition from functional testing into load testing – which, by the way, is simply the process of exercising an application by emulating actual users putting a load on your system. Load testing has historically been very difficult to do correctly. Unpredictable loads, application complexity, and difficulties accurately modeling a real-world user population are factors for this.

These are also the reasons we created a scheduling tool that is all point-and-click. Very intuitive, no programming required. With this tool you can build complex scenarios quickly and maintain the robustness you need for accurate load testing. For example, your schedule can include already-recorded functional scripts (as you might find in your test plan) and/or virtual user scripts used solely for load testing. And because all user groups are percentages of the whole, changing your load test from 100 to 10,000 users is as easy as changing one number. Any data that is passed during these scripts can be specific (read from existing data tables) or parameterized, i.e., read from a random table of data that is created on the fly.

# Speaker Support Notes



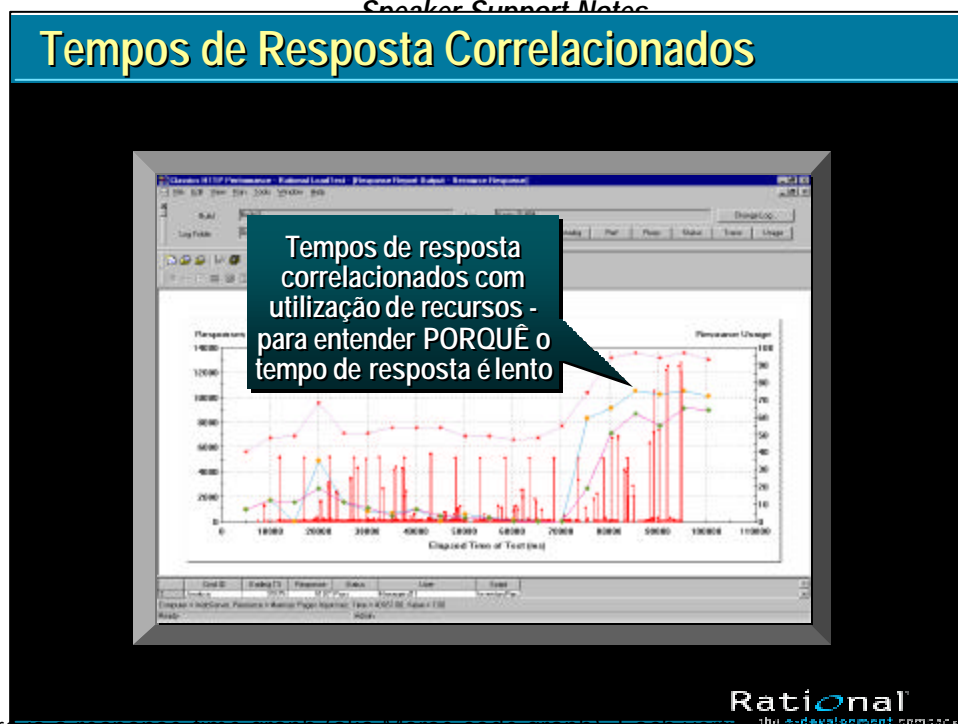
Ease of use is great, but accuracy and scalability are critical success factors in a quality designed and load tested application. When your load test has completed the next step is to analyze the results, determine if you have performance problems and begin the process of resolving those problems. Here's where the accuracy of your test becomes extremely important. With Rational Suite, you can identify precisely where your web application falls down. You can set performance tolerance marks and determine exactly how many users or how many transactions it will take to crash your system. And we can even go a step further.

<CLICK>

With Rational Suite you can peer into each individual transaction (e.g., HTTP Send and Receive calls) if you like - to obtain an unprecedented level of diagnostic capability. This technology, which we call TransactionSmart Analysis, lets you isolate bottlenecks down to the individual statements of client server or server-server interaction.

Only with Rational Suite do you get this ease of use AND unmatched accuracy and scalability.

# Speaker Support Notes



Shown here is a response time graph (aka Morse code graph). Each vertical line represents the time needed for a given transaction to process. That's great – and most load testing tools give you this – however, we go one step further

<CLICK>

We allow you to overlay the monitored resources on top of the transaction graph. That way, you can see at a glance whether you have a hardware constraint. Maybe that's what is causing the slow transaction and maybe it's not. But by correlating response time to resource utilization you can tell very quickly where to look for your bottleneck.

<CLICK>

Beyond that, you can easily correlate resource consumption with response time. This helps you to understand not just where performance is slow - but why it is slow.



# Speaker Support Notes

## Speaker Support Notes

### Avaliação do Teste: Analizando Resultados Detalhados



OK, so let's assume you've done your functional or your load testing and you've found some defects. What do you do? Report them, right? No problem... we provide you with an integrated log viewer which holds all results from a given test run. The log viewer merges the feedback it gets from all the test engines used during playback and builds one full log for each test run – including pass, fail, warning, and informational tags. If you want more info on a certain message, just double-click.

# Speaker Support Notes

Speaker Support Notes

## Avaliação do Teste: Investigando Falhas

✓ Clique duplo na falha p/ obter detalhes

✓ Comparação de dados e descrição textual

The screenshot displays the Rational TestManager interface. A table of test results is visible, with a red row indicating a failure. A callout box points to this row with the instruction 'Clique duplo na falha p/ obter detalhes'. Another callout box points to the drill-down view of the failure, which shows a comparison of baseline and actual data, with the instruction 'Comparação de dados e descrição textual'. The drill-down view shows the following data:

Baseline	Actual
Order Date: 11/11/2008	Order Date: 11/11/2008
Product: Hightech Symplores Max 99.8 101	Product: Hightech Symplores No. 5
Quantity: 5	Qty: 5
Price: 200.00	Unit Price: 200.00
	Order Subtotal

Below the comparison table, the error message is displayed: 'Error not found: 1-Product\*12-Product Symplores Max 99.8 101' and 'Error not found: 1-Product\*12-Product Symplores Max 99.8 101'.

Rational  
the e-development company

The result is exploded in drill-down fashion to reveal the full details of that message. It may be the baseline and actual data of a failed comparison point, like what's shown on our screen. It could also be a message saying that you have a memory leak. Because TestManager can run back any type of test, there is no limit to the kinds of results you can manage with it.

# Speaker Support Notes



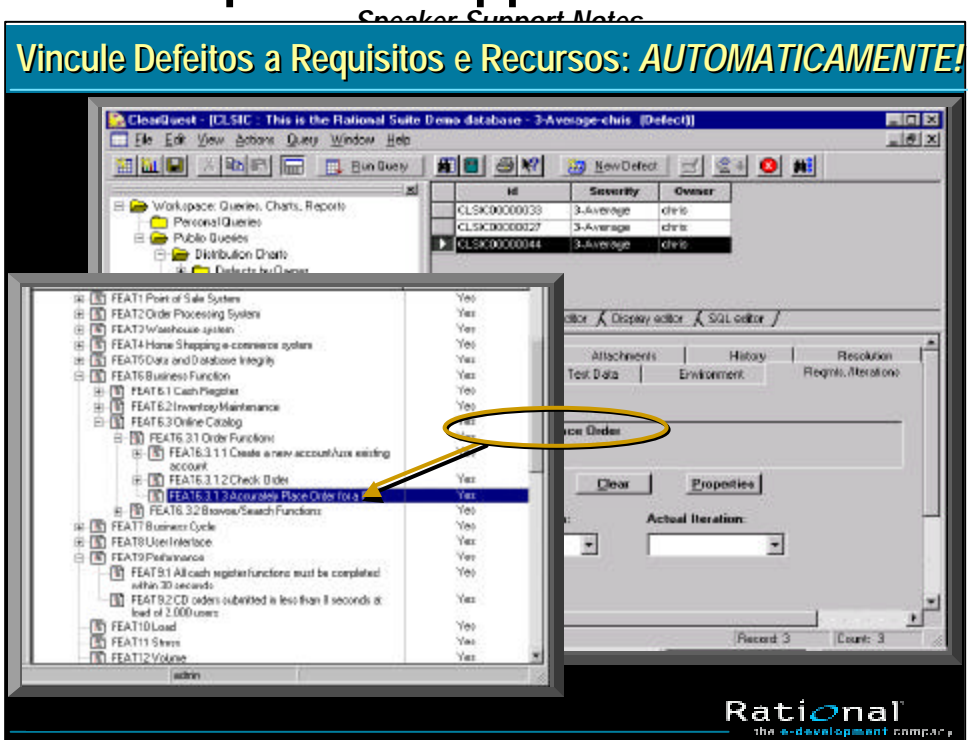
One of the greatest advantages of using Rational Suite TestStudio is the team unification power you gain. Let's take the example of requirements. You may have formal requirements or a product spec in a database or a Word doc. You can look at this requirement from Rational TestManager, the test planning and test management tool. From here you can assign or create test scripts that will be used to measure functional compliance against that requirement.

But, requirements change. They are dynamic, right? No problem with Rational Suite TestStudio. It doesn't matter where you update your requirement - work wherever you are most comfortable. Because the requirement is stored in a central repository, you can edit it from Word, RequisitePro, or TestManager - and it gets automatically updated everywhere.

More importantly, when logging a defect, you can note which requirement this test script was running against. So when the developer or analyst views the defect, they know immediately which business requirement is broken. Rational ClearQuest, the integrated defect management component, automatically pulls in that info when you generate the defect. **The script, verification point, input/requirement, log results, etc. are ALL attached automatically to the defect note, ensuring that the developer can reproduce it quickly and easily.**

Rational ClearQuest has dozens of built-in reports you can run with a single mouse click, so anyone on the team can find out the status of the project at that moment in time!

# Speaker Support Notes

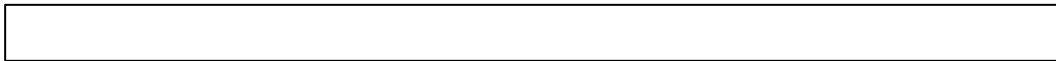


And because everything sits in a central repository (data store), if you want to open a requirement from the defect, it's as easy as clicking a single button. What comes up? The most current version of that requirement. That's the power of a database under the covers.

Let's turn to the demo again....

# Speaker Support Notes

Speaker Support Notes

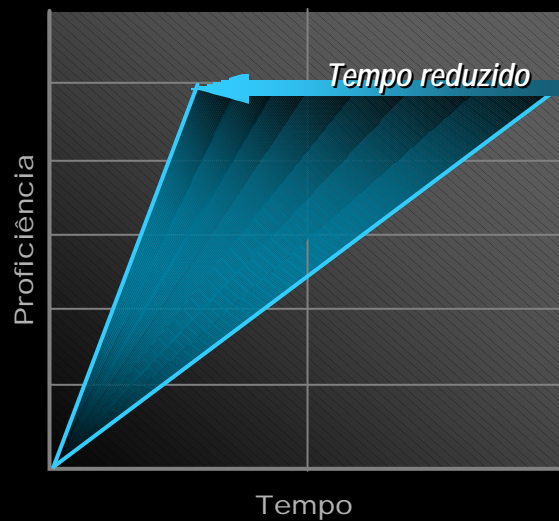


# Speaker Support Notes

Speaker Support Notes

## Aumente a Capacidade da Equipe

Mais de 1.000 profissionais qualificados



Serviços de e-development da Rational

- ♦ QuickStarts
- ♦ Pacotes de Distribuição
- ♦ Avaliações
- ♦ Rational University
- ♦ Consultoria
- ♦ Suporte Técnico
- ♦ Rational Developer Network

**Rational**  
the e-development company™

All Rational products are backed up by our worldwide service and support organization. Customers find that our services are an integral part of their success. They help them adopt new technology faster and accelerate project implementation.

- Over 1000 technical professionals available, worldwide
- Wide array of products and services
- Focus on technology knowledge transfer
- Offerings that are customized for your project or your needs
- Training on both tools and best practices
- The Rational Developer Network, your one-stop resource for white papers, articles, artifacts, and discussions of interest to Rational users\*

\*Available exclusively to Rational Suite users

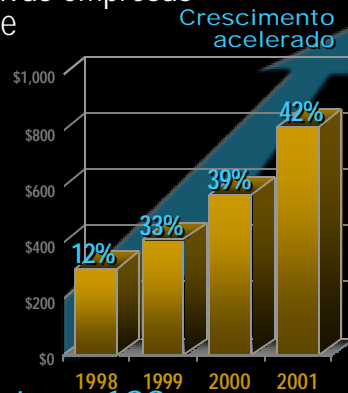
# Speaker Support Notes

Speaker Support Notes

## Sobre a Rational Software

Uma das maiores e mais lucrativas empresas de desenvolvimento de software

- ◆ Fundada em 1981
- ◆ Mais de 3.900 funcionários
- ◆ 70 escritórios no mundo
- ◆ USD 100 milhões gastos em P&D
- ◆ Mais de 500.000 usuários



90 das empresas da *Fortune 100* usam a solução Rational de e-development

**Rational**  
the e-development company™

Here are a few statistics about Rational Software. Rational was founded in 1981, so we've been in business for 20 years. We grew by an astounding 42% last year (year-over-year), and have more than 3,900 employees in 70 offices around the world.

One of the drivers of this growth is our commitment to advancing the discipline of software engineering. Last year we invested nearly \$100 million in research and development.

In fact, the vast majority of the Fortune 100 use the Rational development platform to fuel their own success. These are just some facts to help you understand Rational.

# Speaker Support Notes

Speaker Support Notes

## O "Paradoxo do Desenvolvimento de Software"

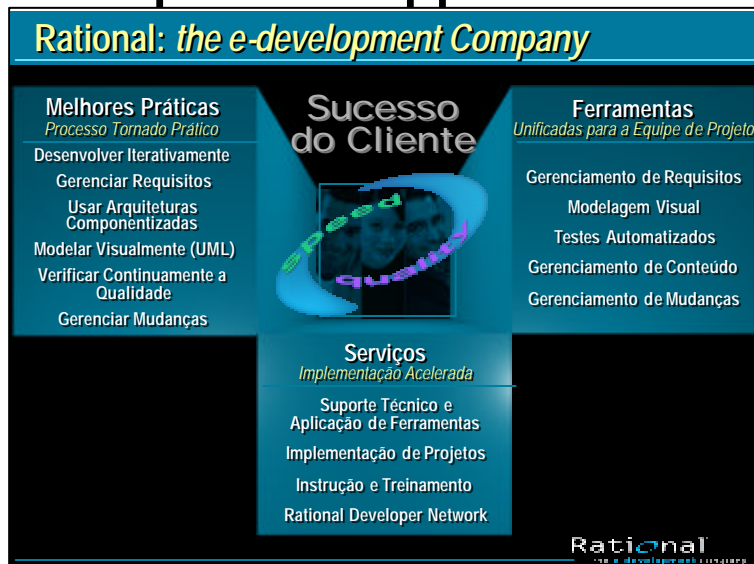


And this brings us right back to where we started this morning – the software development paradox.

Can you overcome it? With Rational behind you – absolutely!



# Speaker Support Notes

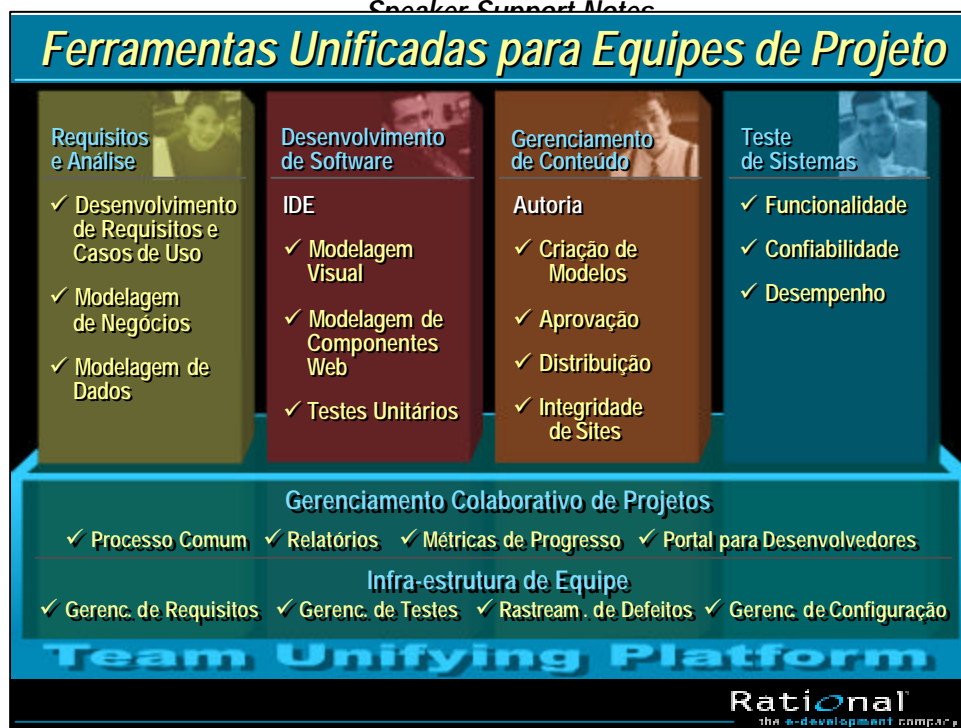


How do our customers overcome the software development paradox?

Several factors contribute to the success of a software project. First, at the heart of the solution, are the people you hire to build software. Now, Rational can't help you in that area. But the three other factors -- the process those teams follow, the skills they use to do their jobs, and the tools they employ-- are all very critical areas in which Rational can provide assistance.

It's the synergy of all three of these critical success factors that enable our customers to produce higher quality software in less time.

# Speaker Support Notes



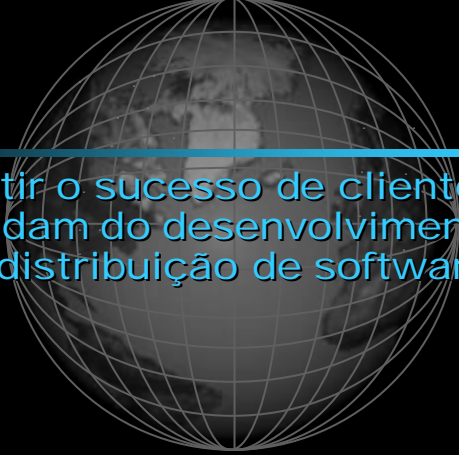
Rational is the only vendor that offers tools that not only fully support the individual but also support the team. Now, the yellow text and checkmarks on this slide indicate the different areas of the software development project team where Rational provides solutions.

As you can see, we cover almost every software development area. In the areas that we don't provide technology, we partner with leading vendors that do. For example, we don't provide scheduling tools for your project management effort, but we integrate with Microsoft Project. We don't provide programming languages, but we integrate with the leaders of integrated development environments like IBM, Microsoft, Sun, HP, and others.

# Speaker Support Notes

Speaker Support Notes

## A Missão da Rational



"Garantir o sucesso de clientes que dependam do desenvolvimento ou da distribuição de software"

**Rational**  
the e-development company

Rational's mission then is to ensure the success of customers who depend on developing or deploying software. It's a mission we've held to for the past 20 years, and one that we plan to continue.... with people like you.

# Speaker Support Notes

Speaker Support Notes

## Onde Obter Mais Informações

- ◆ Cursos da Rational University  
[www.rational.com/university](http://www.rational.com/university)
- ◆ Avaliações de Ferramentas e Downloads  
[www.rational.com](http://www.rational.com) Try It Center
- ◆ Webinars – Seminários On-line Gratuitos  
[www.rational.com/events/webinars](http://www.rational.com/events/webinars)
- ◆ Seminários – Para Locais e Assuntos Próximos de Você  
[www.rational.com/events/seminars](http://www.rational.com/events/seminars)
- ◆ Rational User Conference – Blazing New Trails  
22 a 26 de julho de 2001, Denver, CO  
Inscrições: [www.rational.com/ruc](http://www.rational.com/ruc) ou pelo telefone 888-889-0074



**Rational**  
the e-development company

# Speaker Support Notes

*Speaker Support Notes*



And that concludes our presentation. We thank you for coming and spending the time with us today. At this point I'd like to field any questions you may have at this point.