

3 Especificação do RUP para o desenvolvimento de groupware baseado no Modelo 3C de Colaboração

Este capítulo apresenta o “RUP-3C-Groupware”, processo elaborado nesta tese para o desenvolvimento de groupware baseado no Modelo 3C de Colaboração. Além da revisão da literatura, este processo foi elaborado a partir da experiência acumulada com o desenvolvimento do AulaNet e, principalmente, a partir da abstração e generalização das boas práticas aprendidas com o desenvolvimento do Mediated Chat, abordadas no capítulo anterior. Para sistematizar e formalizar o processo aqui elaborado, foi usado o RUP, Rational Unified Process, que consiste num *framework* de processo de desenvolvimento de software. A seção 3.1 apresenta uma visão geral da estrutura do RUP e são discutidos os fatores que levaram à escolha do RUP como base para a especificação do processo de desenvolvido de groupware elaborado nesta tese. Na seção 3.2 são apresentadas as modificações realizadas nos Fluxos de Engenharia do RUP padrão, evidenciando os papéis, atividades e artefatos especificados ou introduzidos para a efetivação das boas práticas apresentadas na seção 2.4.

3.1. Rational Unified Process

Rational Unified Process, RUP, é um processo proprietário de desenvolvimento de software criado pela IBM Rational Software Corporation. O RUP é um processo bem estruturado para desenvolver software com alta qualidade de modo repetível e previsível (Kruchten, 2003).

Na Figura 53 é representada a arquitetura global do RUP, que é organizada em duas dimensões. O eixo horizontal evidencia o aspecto dinâmico do processo, descrevendo como ocorre o desenvolvimento ao longo do tempo em termos de fases, iterações e marcos. Também mostra como a ênfase varia ao longo do tempo. Por exemplo, nas iterações iniciais, gasta-se mais tempo com modelagem de negócio, requisitos, análise e projeto; enquanto nas iterações finais gasta-se

mais tempo com implementação, teste e distribuição. Embora os nomes dos fluxos de engenharia possam evocar as fases seqüenciais do modelo em cascata, estes fluxos são revisitados ao longo do ciclo de vida, variando de intensidade a cada iteração.

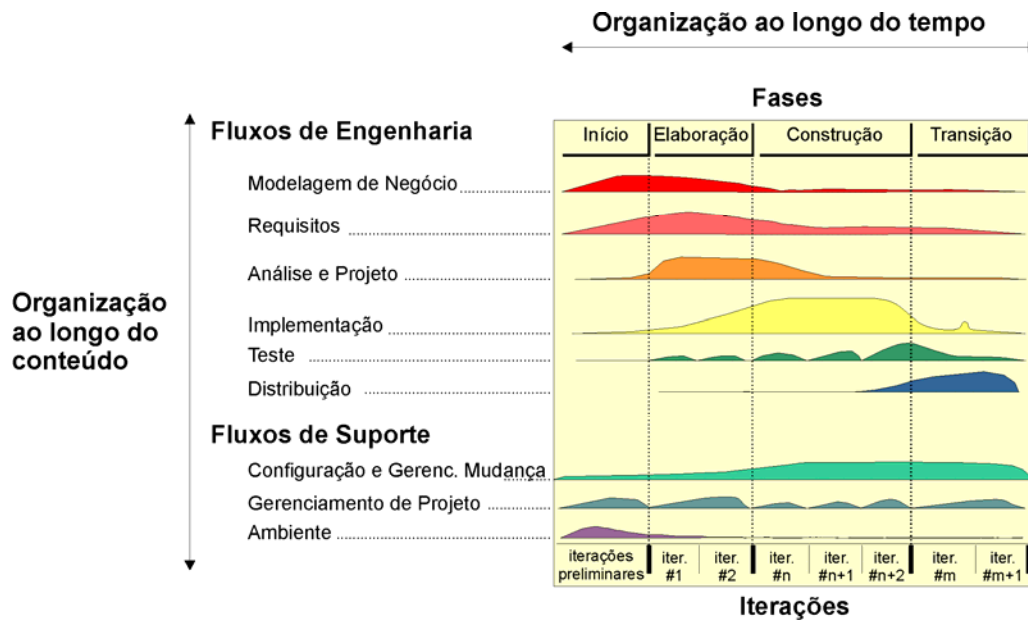


Figura 53. Arquitetura do Processo RUP (2006)

Na Figura 53, o eixo vertical representa o aspecto estático do processo, organizado em termos de disciplinas. No RUP, processo é definido como sendo uma descrição de *quem* está fazendo *o quê*, *como* e *quando* – estes quatro elementos estruturais, correspondem a Papel (quem), Atividade (como), Artefato (o quê) e Fluxo (quando). Na Figura 54 são apresentados todos os conceitos-chave, os elementos estruturais estáticos, definidos no RUP.

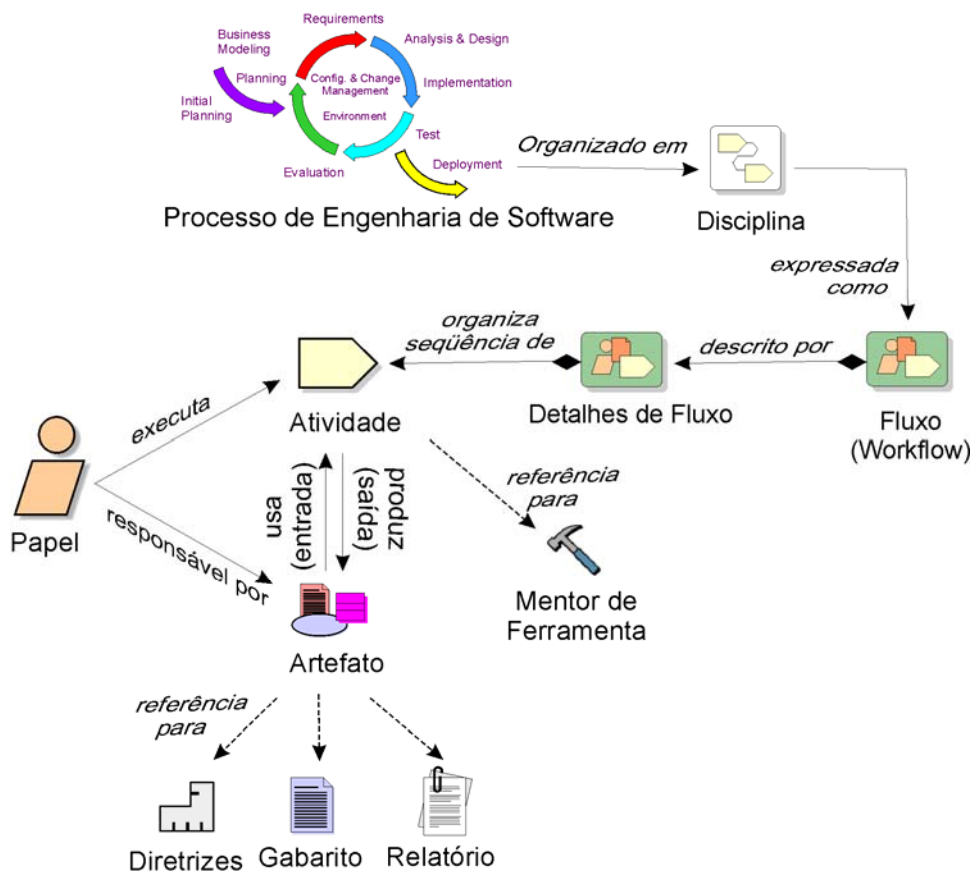


Figura 54. Estrutura Estática do RUP (2006)

Fluxo: é a seqüência de atividades que produz um resultado de valor observável. No RUP, o fluxo é expresso como um diagrama de atividade da UML. Há muitas maneiras de se organizar o conjunto de atividades em fluxos num processo de engenharia de software. No RUP, os fluxos são organizados em dois níveis: *Fluxo* central (*Disciplina*) e *Detalhes de Fluxo*.

Atividade: é o trabalho executado para produzir um resultado significativo no contexto do projeto; consiste, geralmente, na criação ou atualização de artefatos. Toda atividade é atribuída a um papel específico. *Mentor de Ferramenta* fornece diretrizes de como usar uma ferramenta de software específica na execução da atividade.

Papel: define o comportamento e as responsabilidades de um indivíduo ou grupo de indivíduos trabalhando em equipe. O comportamento é expresso em termos de atividades a serem executadas. Responsabilidades são expressas em termos de artefatos que o papel cria, modifica ou controla.

Artefato: é um produto do projeto; pode ser um documento, um modelo, um código-fonte, um programa-executável etc. Um artefato é de responsabilidade

de um único papel, embora possa ser usado por vários papéis. Artefatos são usados, produzidos ou modificados em atividades. *Gabarito (Template)* é um ‘modelo’ do artefato a ser usado em sua criação. Os gabaritos são ligados à ferramenta que será usada. Por exemplo, um *template* do Microsoft Word® pode ser usado como gabarito de um artefato que seja um documento ou relatório. Um *Relatório* consiste em informações que são extraídas de um ou vários artefatos. *Diretrizes* são informações sobre como desenvolver, avaliar e usar os artefatos. Uma atividade representa o trabalho a ser feito enquanto as diretrizes expressam como fazer o trabalho. São regras, recomendações ou métodos para auxiliar a realização de atividades. Descrevem técnicas específicas para criar certos artefatos, transformar um artefato em outro, ou avaliar a qualidade de um artefato.

O RUP foi escolhido como base para a especificação do processo proposto nesta tese, principalmente pelos seguintes fatores:

É um *framework* para processos. O RUP é um meta-modelo de processo, provê uma linguagem para definição de elementos de processo baseada no SPEM, Software Process Engineering Metamodel Specification (OMG, 2002), que estende a UML para especificação de processos. O RUP é declaradamente elaborado para também ser estendido como ilustram as extensões RUP for J2EE² e RUP for Extreme Programming (XP)³. O RUP também é distribuído enquanto produto na forma de um website (disponível para *trial-download*⁴). Também existem ferramentas específicas para implementar as extensões deste produto (Bencomo, 2005).

Possui credibilidade acadêmica e empresarial. O RUP tornou-se uma referência na literatura sobre Processos de Desenvolvimento de Software, sendo um exemplo específico e detalhado do Processo Unificado, elaborado por Jacobson, Booch e Rumbaugh (1999). O RUP também tem grande credibilidade

² <http://www-128.ibm.com/developerworks/rational/library/4224.html>

³ <http://www-128.ibm.com/developerworks/rational/library/4156.html>

⁴ http://www-128.ibm.com/developerworks/downloads/r/rup/index.html?S_TACT=105AGX28&S_CMP=DLMAIN

empresarial, tendo sido adotado por mais de mil empresas até 1999, dentre elas: Ericsson, Xerox, Visa, Oracle etc. (Kruchten, 2003, p.18).

É iterativo e baseado em componentes. Estes dois aspectos técnicos são contemplados pelo RUP e eram exigidos de um processo a ser usado como base para esta tese (uma decisão de projeto). O RUP também contempla outros aspectos que eram desejáveis, mas não indispensáveis, tal como a modelagem baseada na UML. Contudo, o RUP não contempla alguns outros aspectos técnicos de desenvolvimento genérico de software que também eram desejáveis, tais como Análise de Domínio e Desenvolvimento Baseado em Reuso – aspectos que tiveram que ser incorporados na extensão proposta nesta tese.

3.2. Fluxos de Engenharia do RUP-3C-Groupware

Nesta seção são apresentadas as modificações elaboradas no RUP para torná-lo específico para o desenvolvimento de groupware e para incorporar as práticas descritas na seção 2.4. Como referência para a elaboração do processo aqui proposto, foi usada a versão RUP 2003.06.15, que era a versão disponível mais atualizada até a data de escrita desta tese. Para evidenciar o que foi modificado do RUP padrão, os elementos definidos no RUP-3C-Groupware são apresentados em destaque nas figuras e em negrito no texto.

As mudanças realizadas, apresentadas nas próximas subseções, são restritas aos fluxos de engenharia do RUP, sendo mantidos inalterados os fluxos de gerenciamento e de suporte, pois assume-se que um projeto de groupware não apresenta gerenciamento e suporte diferenciado de um projeto de software genérico.

3.2.1. Modelagem de Negócio

No RUP padrão, o fluxo Modelagem de Negócio é executado com o objetivo de entender o domínio empresarial antes de iniciar o projeto de engenharia de software (Kruchten, 2003). É preciso gerar conhecimento sobre o domínio empresarial no caso genérico em que o software é construído sob medida para uma empresa. Procura-se gerar um vocabulário comum entre clientes,

usuários e desenvolvedores do sistema, capturar a estrutura e a dinâmica da organização na qual o sistema será implantado, e identificar os problemas e as possíveis melhoras que serão derivadas em requisitos do sistema de software a ser desenvolvido.

A modelagem de negócio não é recomendada para todo trabalho de engenharia de software (Kruchten, 2003). Por exemplo, não é preciso considerar a modelagem de negócio quando o objetivo é acrescentar apenas uma nova característica a um software existente, sem modificá-lo radicalmente. Mas se estiver sendo construído um novo sistema para dar suporte ou automatizar alguns procedimentos existentes na organização, neste caso a modelagem de negócio pode ser útil para entender como o novo sistema afeta o modo de administrar o negócio.

O fluxo de Modelagem de Negócio é apresentado na Figura 55. De acordo com o fluxo do RUP padrão, pode-se seguir diferentes caminhos dependendo do propósito da modelagem de negócio e da posição no ciclo de vida do desenvolvimento do projeto de software. Na primeira iteração, avalia-se o estado da organização e são determinadas as áreas a serem melhoradas. Com base nos resultados desta avaliação, toma-se a decisão em que sentido continuar no fluxo (no RUP padrão são fornecidos cenários típicos para apoiar esta decisão). Assim, dependendo do estado do negócio, pode-se determinar que:

- Modelagem do Negócio é necessária. Se o objetivo é melhorar ou fazer a reengenharia de um negócio existente, será preciso modelar o negócio atual e o novo negócio. Se o objetivo é desenvolver um novo negócio, deve-se saltar o “Descrever o Negócio Atual”.
- Apenas um Modelo de Domínio é suficiente, não sendo necessários modelos completos de negócio. No RUP, um modelo de domínio é considerado um subconjunto do modelo de análise do negócio focando apenas nas entidades daquele modelo (conceitos, produtos, e eventos que são importantes para o domínio do negócio).

O Modelo 3C de Colaboração é útil para analisar a colaboração, mas nesta tese não foi realizada a especificação do fluxo da Figura 55 para modelar a colaboração, pois isto não foi praticado em projetos anteriores. Em vez de modelar a colaboração realizada num grupo em particular, a prática aprendida no projeto desta tese parte da análise das ferramentas existentes que sejam candidatas a serem usadas para dar suporte à dinâmica do grupo. É uma abordagem mais baseada na tecnologia e fundamentada na Análise de Domínio. Análise de Domínio é o termo cunhado por Neighbors (1981) para designar a “tentativa de se identificar objetos, operações e relacionamentos entre tudo o que os especialistas em um domínio julgam importante para este domínio”. Outras definições (Werner e Braga, 2005) são:

- “É o processo de identificar e organizar o conhecimento sobre alguma classe de problemas – o domínio de problema – para apoiar a descrição e a solução destes” (Pietro-Diaz e Arango, 1991)
- “É o processo de identificar, capturar e organizar a informação usada no desenvolvimento de software, com a finalidade de permitir a sua reutilização” (Pietro-Diaz, 1987)
- “É a identificação de objetos e operações de uma classe de sistemas similares em um domínio de problema particular” (Freeman, 1983)
- “É a atividade de identificação, aquisição e avaliação da informação reutilizável no domínio do problema, para ser utilizada na especificação e construção do sistema de software” (Arango, 1989)
- “É o processo de identificar, colecionar, organizar e representar um modelo de domínio e a arquitetura de software dos estudos dos sistemas existentes, da teoria de apoio, da tecnologia emergente e dos históricos de desenvolvimento, dentro do domínio de interesse” (Peterson, 1991).

Com a Análise de Domínio, objetiva-se organizar o conhecimento sobre um conjunto de aplicações para facilitar o posterior reuso das informações no processo de desenvolvimento de um novo software daquele domínio. O resultado da Análise de Domínio é uma taxinomia de sistemas que compartilham características comuns, evidenciando as similaridades e diferenças entre os sistemas estudados (Arango, 1994).

Como discutido no capítulo anterior, a Análise de Domínio possibilitou a organização dos elementos 3C que deveriam ser considerados no desenvolvimento das versões Mediated Chat. A importância da Análise de Domínio foi particularmente notada para a caracterização do problema, para o levantamento de possíveis soluções, e para auxiliar na revisão das versões desenvolvidas.

Para incorporar esta prática, o fluxo Modelagem de Negócio do RUP padrão foi alterado conforme destacado na Figura 55 (o que foi mantido do RUP padrão é representado sem preenchimento e com linhas em cinza claro, enquanto as modificações introduzidas aparecem em destaque, com preenchimento e com linhas em preto). No RUP-3C-Groupware, ao avaliar o Estado do Negócio deve-se tomar a decisão sobre qual abordagem seguir: Análise do Domínio ou Modelagem da Colaboração. Se for decidido seguir a Análise de Domínio, devem ser executadas as atividades e produzidos os artefatos especificados no detalhe do fluxo “Analisar Domínio” abordado adiante. Se for decidido seguir a abordagem de Modelagem da Colaboração, deve-se seguir os procedimentos estabelecidos no RUP padrão, estando alerta para o fato de que estes procedimentos podem não ser suficientes uma vez que não são específicos para o desenvolvimento de groupware.

Ao optar pela abordagem Análise de Domínio, deve-se “Analisar Domínio” de acordo com os detalhes do fluxo apresentado na Figura 56. É adequado assinalar, contudo, que não são claras as fronteiras de um domínio: groupware pode ser visto como um domínio único; ou então, as aplicações de bate-papo podem ser vistas como um domínio mais específico; já no projeto Mediated Chat, foi útil considerar o domínio das aplicações de comunicação síncrona. As fronteiras do domínio a ser analisado dependem do projeto em desenvolvimento.

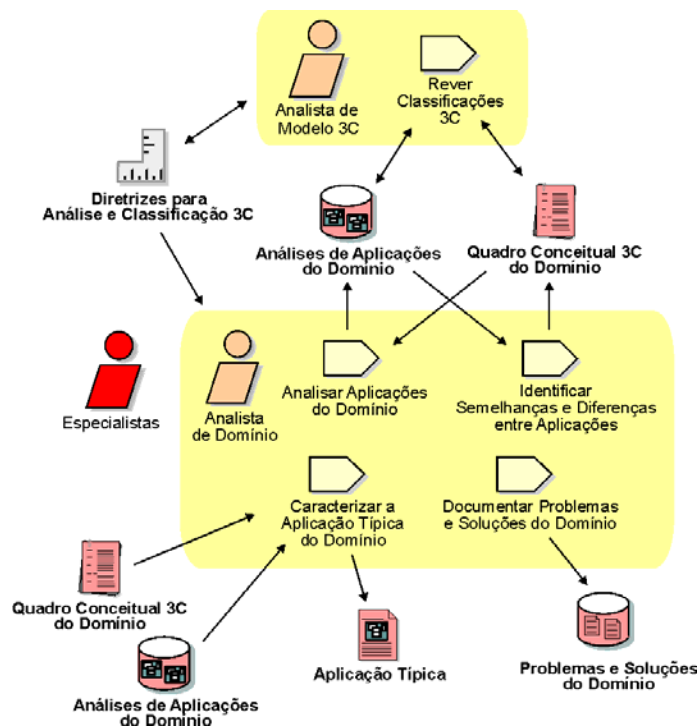


Figura 56. Detalhes do fluxo Analisar Domínio, proposto no RUP 3C Groupware

O papel **Analista de Domínio** é responsável por analisar e comparar as diferentes aplicações do domínio para o qual o novo groupware está sendo desenvolvido, e também por caracterizar a aplicação típica do domínio em análise. Para realizar estas atividades, o Analista de Domínio deve contar com diversas fontes de informação tais como especialistas do domínio, aplicações existentes, publicações e outros documentos (Kang et al., 1990; Werner & Braga, 2005).

A atividade **Analisar Aplicações do Domínio** tem por objetivo documentar diferentes aplicações do domínio. De cada aplicação, deve-se documentar as principais funcionalidades e classificá-las de acordo com o Quadro Conceitual 3C. A ausência de uma classificação adequada para uma funcionalidade indica que o Quadro Conceitual precisa ser revisto; ou então, indica que aquela funcionalidade não está relacionada à colaboração. O objetivo desta atividade é alimentar o repositório de Aplicações do Domínio. Esta atividade deve ser executada nas fases de Início e de Elaboração, e revista sempre que preciso nas fases subsequentes.

O artefato **Análises de Aplicações do Domínio** é um repositório para armazenar a análise de diversas aplicações do domínio. Como exemplo de análise, no Anexo 1 está disponível a documentação da análise da aplicação PalTalk. Alimentar o repositório é uma forma de promover o conhecimento sobre o domínio. O repositório é usado em outras etapas do projeto de groupware em

desenvolvimento. Deve possibilitar a recuperação das diferentes aplicações que implementam uma dada funcionalidade, o que é útil para comparar os diferentes mecanismos de implementação. Também deve possibilitar a recuperação das funcionalidades relacionadas a um elemento do Quadro Conceitual 3C, o que é útil para auxiliar a compreensão dos conceitos abstratos do quadro conceitual.

Na atividade **Identificar Semelhanças e Diferenças entre Aplicações**, o Analista de Domínio deve estabelecer comparações entre as aplicações analisadas buscando abstrair os elementos de comunicação, coordenação e cooperação do domínio. Como resultado desta atividade, objetiva-se construir um Quadro Conceitual 3C do domínio, ou aperfeiçoar algum que já esteja em uso no projeto. Esta atividade deve ser executada nas fases de Início e de Elaboração, e revista sempre que preciso nas fases subseqüentes.

O artefato **Quadro Conceitual 3C** é a organização, em função do Modelo 3C de Colaboração, dos elementos relevantes constituintes de aplicações no domínio do groupware em desenvolvimento. Como exemplo deste artefato, na seção 2.1 foi discutido o Quadro Conceitual 3C do domínio Ferramentas de Comunicação Síncrona elaborado no projeto Mediated Chat. Sugere-se que o Analista de Domínio não desenvolva um Quadro Conceitual 3C a partir do zero, devendo partir, preferencialmente, de alguma proposta que já tenha sido documentada. Por exemplo, como alternativa ao Quadro Conceitual 3C desenvolvido no projeto Mediated Chat, o Analista de Domínio poderia partir da proposta apresentada por Miranda et al. (2005), desenvolvido para o domínio de aplicações de comunicação – neste caso, o Analista de Domínio ainda teria que rever aquele quadro reorganizando os elementos em função do Modelo 3C de Colaboração.

Na atividade **Caracterizar a Aplicação Típica**, deve-se caracterizar o que é uma aplicação típica do domínio em questão, atividade a ser executada logo no início do projeto. Com uma boa quantidade de exemplos de aplicações do domínio, e com a abstração das semelhanças e diferenças organizadas no Quadro Conceitual, o Analista de Domínio deve ser capaz de identificar qual é o conjunto mínimo de funcionalidades para caracterizar uma aplicação naquele domínio. Se o objetivo for construir uma aplicação totalmente nova, que não tenha aplicações que possam servir como referência nem mesmo como uma primeira aproximação (o repositório de aplicações do domínio estará vazio e não existirá um Quadro

Conceitual 3C), então há duas estratégias. A primeira é supor quais seriam os elementos mínimos para dar suporte à Comunicação, Coordenação e Cooperação daquele trabalho colaborativo, gerando assim a especificação de uma Aplicação Básica, que substituirá a Aplicação Típica (não existe a aplicação típica já que não existem aplicações semelhantes). A segunda estratégia é desistir de usar a abordagem de Análise de Domínio e realizar a Modelagem da Colaboração.

O artefato **Especificação da Aplicação Típica** serve como uma referência, um registro do conjunto mínimo dos elementos relevantes que seja suficiente para caracterizar uma aplicação naquele domínio. A aplicação-típica serve como base para que sejam realizadas sucessivas modificações em busca da solução do problema. Se está sendo desenvolvido um novo sistema, sugere-se que toda a primeira iteração concentre-se no desenvolvimento da versão típica, que resultará numa arquitetura essencial e um conjunto mínimo de componentes para colocar em uso uma primeira versão da aplicação para que possa ser avaliada.

Na atividade **Documentar Problemas e Soluções do Domínio** deve-se alimentar o repositório **Problemas e Soluções do Domínio** com os problemas já conhecidos e com as soluções já avaliadas naquele domínio. Este trabalho é útil, por exemplo, para especificar para o novo groupware uma solução (ou variação da solução) que já se saiba ser adequada ao menos em outras aplicações.

O papel **Analista de Modelo 3C** deve possuir um conhecimento mais aprofundado sobre o Modelo 3C de Colaboração, sendo responsável pelo uso consistente deste modelo ao longo do processo de desenvolvimento do groupware. Nos detalhes do fluxo em questão, o Analista de Modelo 3C executa a atividade **Rever Classificações 3C** realizadas pelo Analista de Domínio. O Analista de Modelo 3C não precisa ser um especialista no domínio do groupware em desenvolvimento, mas precisa saber realizar análise e classificação em função do Modelo 3C de Colaboração, sendo responsável por modificar, se necessário, o artefato contendo as Diretrizes para Análise e Classificação 3C.

A atividade **Rever Classificações 3C** deve ser realizada sempre que uma nova aplicação é analisada ou quando o Quadro Conceitual 3C é modificado, objetivando garantir o uso consistente do Modelo 3C.

O artefato **Diretrizes para Análise e Classificação 3C** documenta o Modelo 3C de Colaboração e seu uso na Classificação das Aplicações Groupware, como abordado na seção 2.1. Neste artefato também são fornecidas classificações

de diversos elementos, o que auxilia, por comparação e aproximação, a classificação de um novo elemento.

3.2.2. Requisitos

No RUP padrão, o fluxo Requisitos objetiva estabelecer o que o sistema deve fazer. Este fluxo é executado para sistematicamente elicitar, documentar, organizar e acompanhar as mudanças dos requisitos de um sistema. “Um requisito é definido como ‘uma condição ou capacidade para a qual um sistema tem que se conformar’.”(Kruchten, 2003, p.132). No RUP padrão, os requisitos são descritos em função do modelo FURPS (Grady, 1992): Funcionalidade, Usabilidade, Confiabilidade (Reliability), Desempenho e Suporte. Os requisitos funcionais descrevem o comportamento do sistema, aquilo que o sistema faz para o usuário, e são descritos no RUP através de Casos de Uso. Os demais requisitos são denominados não-funcionais e são descritos no RUP como Especificação Suplementar.

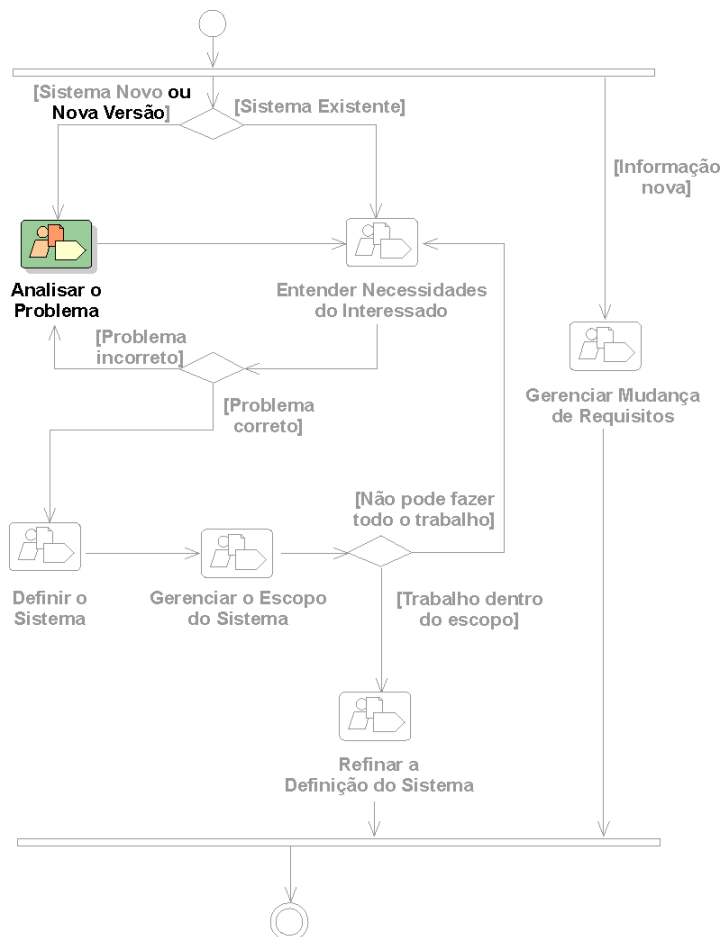


Figura 57. Fluxo de Requisitos

Na Figura 57 é apresentado o fluxo de Requisitos. De acordo com este fluxo, devem ser executadas as atividades descritas a seguir.

Analisar o Problema, para entender as necessidades iniciais do interessado, entender o problema e propor soluções em alto-nível. Objetiva-se obter um acordo sobre qual é o verdadeiro problema(s) e quem são os interessados – um interessado pode ser um usuário final, um cliente (comprador ou contratante), um gerente de projeto “ou qualquer outro que se preocupe bastante ou de quem as necessidades devam ser satisfeitas pelo projeto” (Kruchten, 2003, p.134). Também são delimitadas as fronteiras da solução a ser implementada.

Entender Necessidades do Interessado, para entender mais claramente as solicitações dos interessados, usando técnicas como entrevistas, brainstorming, prototipagem conceitual, questionários, e análise competitiva. O resultado deve ser uma lista de solicitações descritas textualmente ou graficamente (através de casos de uso) com prioridade relativa entre elas.

Definir o Sistema, traduzindo as necessidades do interessado para uma descrição significativa do sistema, usando linguagens natural e gráfica. O objetivo é convergir dos requisitos de alto-nível para requisitos detalhados, incluindo os não funcionais. Parte desta atividade inclui modelos e protótipos iniciais diretamente relacionados às solicitações mais importantes do interessado.

Gerenciar o Escopo do Sistema, priorizando os requisitos e escolhendo cuidadosamente os que serão implementados a cada iteração visando diminuir os riscos conhecidos no projeto. É preciso negociar com os interessados o escopo de cada iteração do projeto, o que requer boa habilidade em gerenciar expectativas de resultado do projeto em suas diferentes fases.

Refinar a Definição do Sistema, de tal forma que os interessados possam entendê-la, concordar com, e assiná-la. É sugerido que sejam usados casos de uso combinado com protótipos visuais como uma forma adequada de comunicar o objetivo e os detalhes do sistema. Casos de uso auxiliam na contextualização dos requisitos; contam uma história de como o sistema será usado.

Gerenciar Mudanças de Requisitos, para avaliar o impacto das mudanças de requisitos solicitadas, e para gerenciar o impacto da mudança em cadeia a ser realizada nos artefatos.

O fluxo de Requisitos definido no RUP padrão não foi estruturalmente alterado. Como destacado na Figura 57, somente o detalhe de fluxo **Analisar o Problema** foi modificado. No RUP-3C-Groupware, é pressuposto o desenvolvimento através de sucessivas versões, sendo resolvido um problema a cada versão, conforme a boa prática aprendida com o desenvolvimento das versões Mediated Chat (subseção 2.4.2). Portanto, o detalhe de fluxo Analisar o Problema deve ser executado a cada nova versão e não somente perante um novo sistema como original definido no RUP padrão. Este detalhe de fluxo foi alterado conforme destacado na Figura 58 e discutido a seguir.

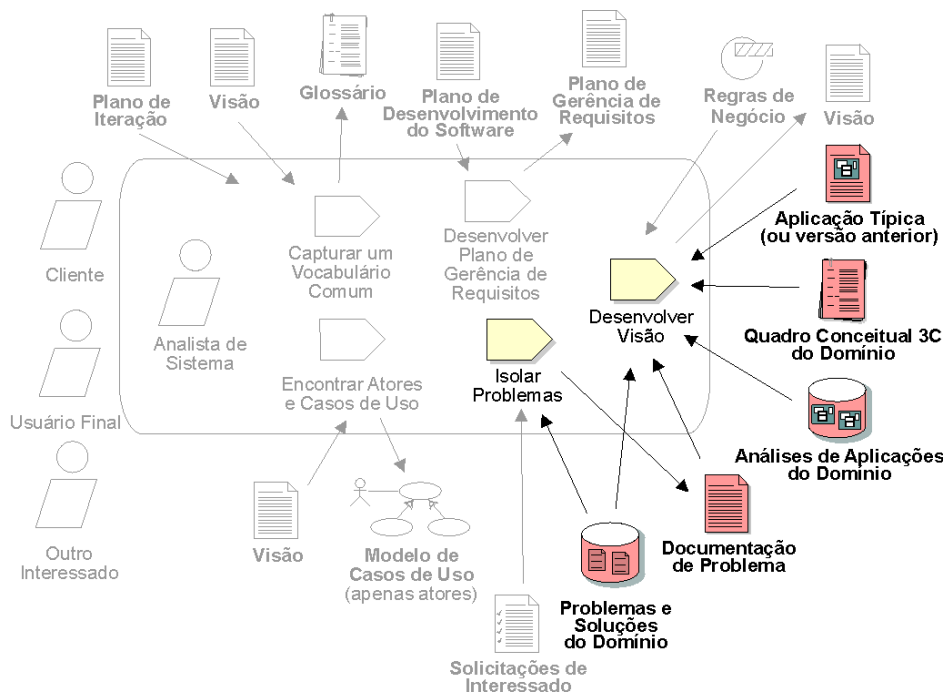


Figura 58. Detalhe do fluxo “Analisar Problema” modificado no RUP-3C-Groupware

No RUP padrão, ao realizar a atividade **Desenvolver Visão**, o Analista de Sistema deve partir das Regras de Negócio e das Solicitações do Interessado (*Stakeholder Requests*) para gerar o artefato **Visão** (*Vision*) que contém as principais necessidades de interessados e características do sistema a ser desenvolvido. No RUP-3C-Groupware, primeiro deve-se executar a atividade **Isolar Problema** para identificar e formalizar os problemas relatados pelo interessado. Deve-se consultar o repositório de Problemas e Soluções do Domínio para comparar os problemas relatados com outros já conhecidos. Se for identificado um problema novo, deve-se produzir o artefato **Documentação de Problema** e cadastrá-lo no repositório. O Anexo 1 contém exemplos dos artefatos Solicitações de Interessado e Documentação de Problema.

Formalizados os problemas relatados pelo interessado, deve-se selecionar um único problema (o de mais alta prioridade) para desenvolver uma nova versão do groupware. Na atividade **Desenvolver Visão**, o artefato visão a ser produzido deve explicitamente estar relacionado com o problema selecionado. Para especificar as características da versão a ser desenvolvida, que serão usadas para definir os Requisitos, deve-se partir de uma solução já conhecida, se existir, ou propor uma variação de alguma solução relacionada (consulta-se o repositório Problemas e Soluções do Domínio). Ao estabelecer as características da versão a

ser desenvolvida a partir da solução proposta, deve-se considerar as outras Aplicações do Domínio, o Quadro Conceitual 3C, e os elementos já existentes na Aplicação Típica (caso seja a primeira versão a ser desenvolvida) ou numa versão desenvolvida anteriormente. O Anexo 1 contém exemplo do artefato Visão a ser produzido de acordo com estes procedimentos estabelecidos no RUP-3C-Groupware.

3.2.3. Análise e Projeto

O fluxo Análise e Projeto, apresentado na Figura 59, objetiva traduzir os requisitos numa especificação que descreve como implementar o sistema. A análise foca os requisitos funcionais do sistema e, por causa da simplicidade, expressa uma imagem quase ideal do sistema. O projeto é um refinamento da análise, e foca na adaptação dos resultados da análise às restrições impostas pelos requisitos não-funcionais tais como os requisitos de desempenho, ambiente de implementação etc. (Kruchten, 2003).

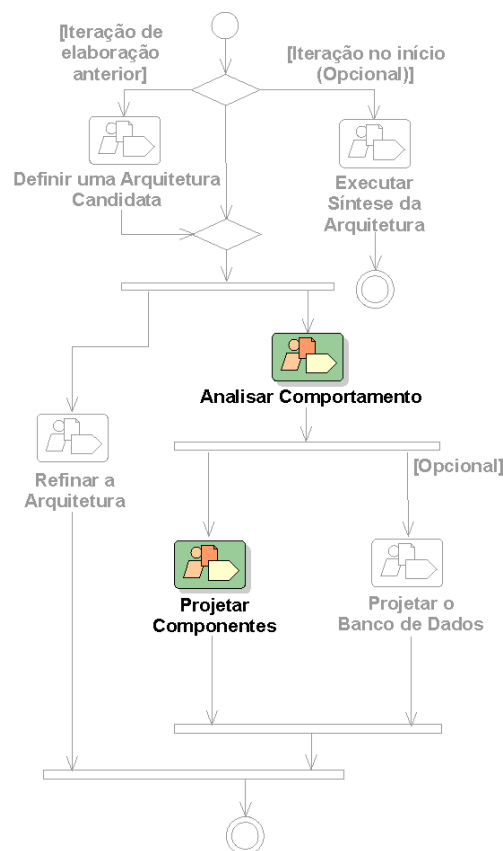


Figura 59. Fluxo de Análise e Projeto

Na fase inicial, no fluxo Análise e Projeto deve-se determinar se o sistema tal como vislumbrado é viável e deve-se avaliar o potencial das tecnologias para a solução – objetivos do detalhe do fluxo Executar Síntese da Arquitetura. Se há poucos riscos (por exemplo, o domínio é bem conhecido), este detalhe de fluxo pode ser dispensado.

No início da Fase de Elaboração, deve-se criar uma arquitetura inicial para o sistema – Definir uma Arquitetura Candidata – estabelecendo um ponto de partida para o trabalho principal de análise. Se a arquitetura já existir, seja porque foi produzida em iterações anteriores, em projetos anteriores, ou obtida de um framework de aplicação, o trabalho é redirecionado para Refinar a Arquitetura, e para Analisar o Comportamento para criar um conjunto inicial de elementos que satisfazem o comportamento esperado.

Os elementos inicialmente identificados são posteriormente refinados. Projetar Componentes produz um conjunto de componentes que satisfazem o comportamento esperado e satisfazem os requisitos do sistema. Se o sistema inclui um banco de dados, então Projetar Banco de Dados deve ser executado em paralelo. O resultado é um conjunto inicial de componentes que serão posteriormente refinados na disciplina Implementação.

Uma modificação realizada no RUP padrão foi realizada no detalhe do fluxo **Analisar Comportamento**, como destacado na Figura 60. No RUP-3C-Groupware foi especificado que para **Projetar e Prototipar a Interface-com-Usuário**, o projetista deve considerar também os artefatos desenvolvidos na Análise de Domínio: Aplicação Típica (ou Versão Anterior); Análises das Aplicações de Domínio; e Quadro Conceitual 3C do Domínio. Por exemplo, dado uma funcionalidade especificada no artefato Visão, o projetista consulta o repositório Análises das Aplicações de Domínio para extrair os diferentes mecanismos que implementam aquela funcionalidade em outras aplicações. Estes mecanismos fornecem algumas possibilidades para o projetista selecionar ou adaptar para o groupware em desenvolvimento um mecanismo já conhecido, possibilitando assim o reuso de elementos de interface-com-usuário.

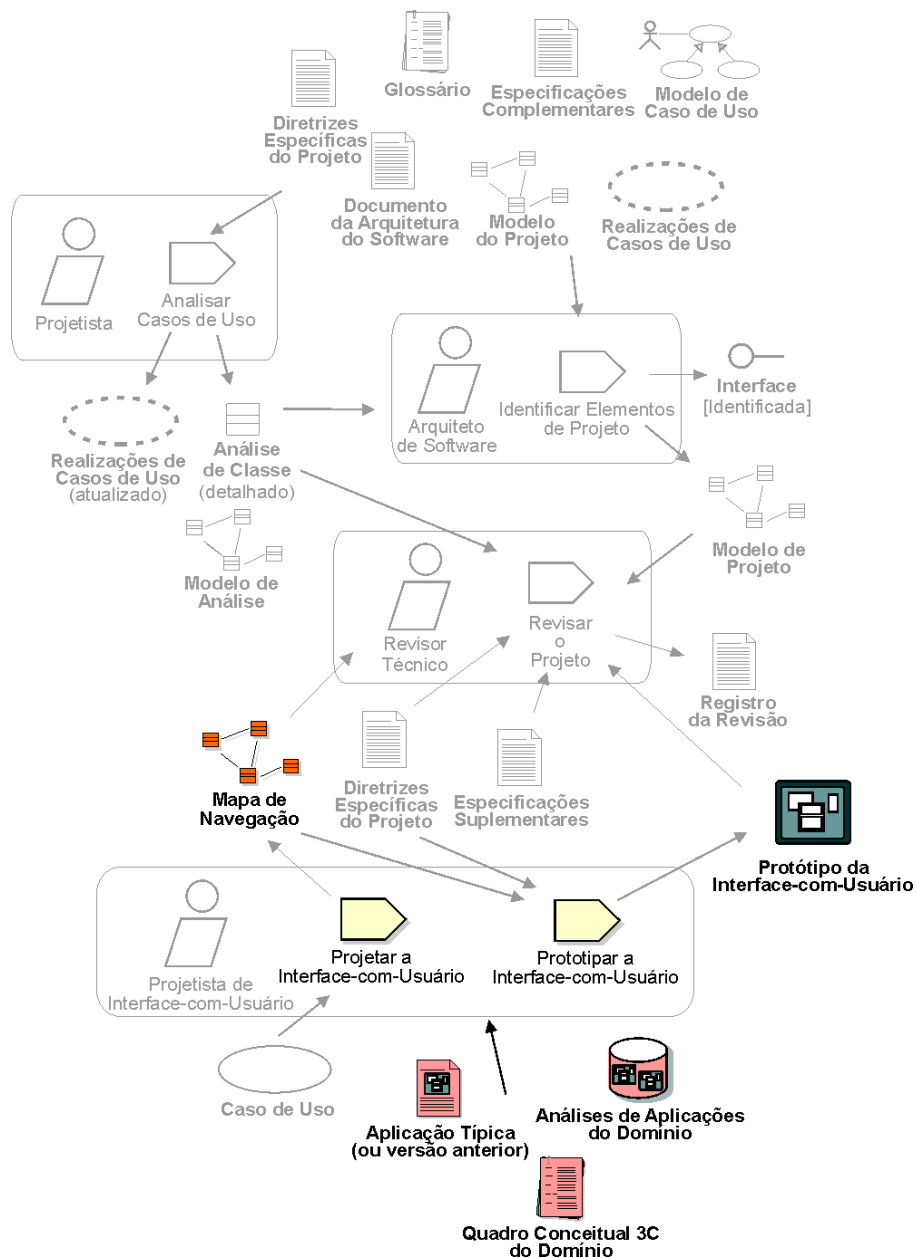


Figura 60. Detalhe do fluxo “Analisar Comportamento” do RUP-3C-Groupware

O artefato **Mapa de Navegação**, também previsto no RUP padrão, foi especializado no RUP-3C-Groupware sendo adotado especificamente o Diagrama de Páginas, que consiste na estereotipação do Diagrama de Estados da UML, para a construção do mapa. A partir da prática adquirida com o projeto AulaNet, identifica-se que este diagrama é útil para discutir as páginas da aplicação e os eventos que possibilitam a navegação entre elas. O anexo 1 contém um exemplo deste artefato.

isoladamente, devem ser integrados num subsistema que, por sua vez, após ser testado e validado, um subsistema é integrado ao sistema.

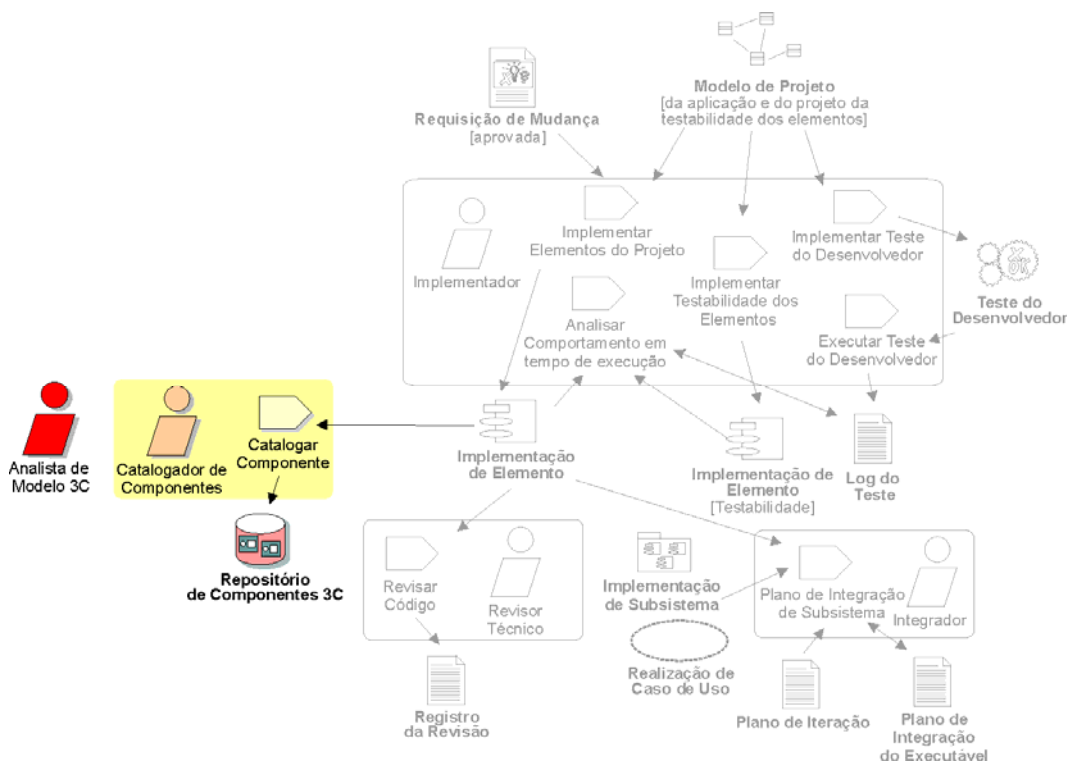


Figura 63. Detalhes do fluxo “Implementar Componente” do RUP-3C-Groupware

No RUP-3C-Groupware o detalhe de fluxo **Implementar Componente** foi modificado conforme destacado na Figura 63. De forma simétrica à modificação no **Projetar Componente** descrita na subseção anterior, cada novo componente implementado deve agora ser armazenado no **Repositório de Componentes 3C**. O papel **Catalogador de Componentes** é responsável por **Catalogar Componente** com auxílio do **Analista de Modelo 3C**.

É adequado assinalar que a implementação de um componente deve seguir uma arquitetura técnica – por exemplo, no projeto AulaNet 3, os componentes serão implementados seguindo a arquitetura definida por Barreto (2006) onde o código faz uso de *frameworks* como Spring e Hibernate (ver subseção 1.3.1).

3.2.5. Teste

O fluxo Teste, apresentado na Figura 64, objetiva principalmente avaliar a qualidade de produto. A função do teste não é assegurar a qualidade, mas avaliá-la, encontrar as fraquezas do produto de software.

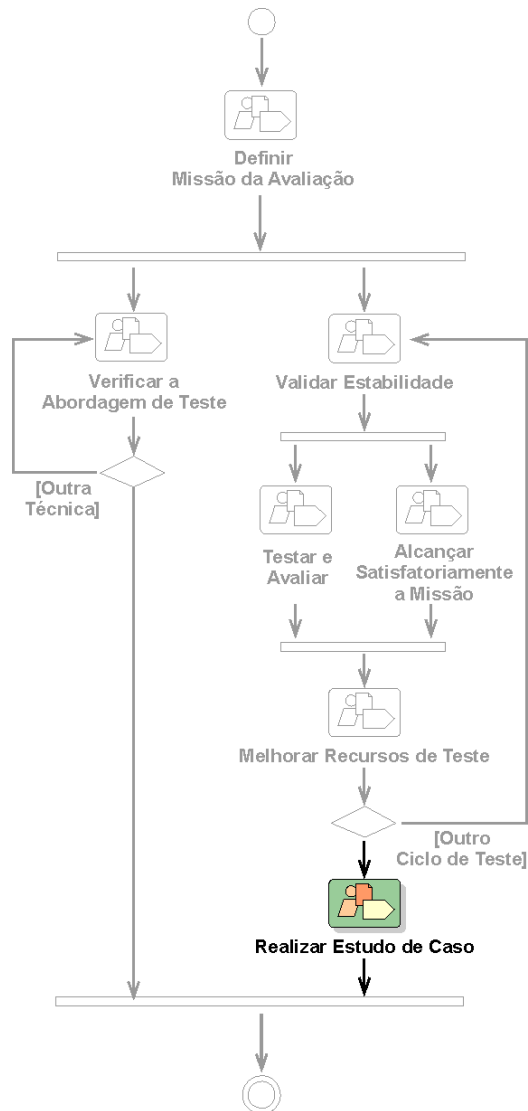


Figura 64. Fluxo de Teste

Definir Missão da Avaliação objetiva identificar qual o foco adequado para o teste durante cada iteração, e também negociar com os interessados quais são as metas que direcionarão os testes. Verificar a Abordagem de Teste é realizado para investigar se a abordagem selecionada irá funcionar, produzir resultados acurados e se é adequada para os recursos disponíveis. E Validar a Estabilidade é executado para inspecionar se a construção está estável o suficiente para iniciar teste e avaliação detalhada, prevenindo assim que recursos sejam desperdiçados.

Testar e Avaliar visa atingir adequada amplitude e profundidade de teste para possibilitar uma avaliação suficiente dos itens alvos (onde “avaliação suficiente” é definido pelos atuais motivadores de teste e missão de avaliação). Alcançar Satisfatoriamente a Missão tem por objetivo entregar para os interessados uma avaliação útil dos resultados dos testes.

Melhorar Recursos de Teste tem o propósito de melhorar os testes, sendo principalmente importante se o objetivo for reusar os recursos desenvolvidos no atual ciclo de testes para os ciclos de teste subseqüentes.

No RUP padrão, os testes são classificados em:

- Teste de Unidade – focados na verificação dos menores elementos testáveis do software. Tipicamente, aplica-se este teste num componente para verificar o fluxo de controle e de dados, e se o componente funciona como esperado. O programador executa os teste de unidade conforme a unidade é desenvolvida. Os detalhes do teste de unidade são descritos na disciplina Implementação.
- Teste de Integração – executado para garantir que os componentes continuam funcionando adequadamente quando são combinados para executar um caso de uso. O alvo do teste é tipicamente um pacote ou um conjunto deles.
- Teste de Sistema – tradicionalmente feito quando o software está funcionando como um todo. Com o ciclo de vida iterativo, é possível executar teste de sistema assim que subconjuntos do comportamento do caso de uso estão implementados. Usualmente, são testados os elementos funcionais do sistema.
- Teste de Aceitação – teste de aceitação do usuário é o teste final feito antes da liberação do software. O objetivo do teste é verificar se o software está pronto para ser usado pelos usuários na execução das funções e tarefas para as quais o software foi construído.

Nos testes, procura-se por erros de implementação e verificação da conformidade com os requisitos. Contudo, não há testes específicos para investigar a adequação da solução proposta para o problema que originou o desenvolvimento da versão, objetivo do detalhe de fluxo **Realizar Estudo de Caso** introduzido na disciplina Teste do RUP-3C-Groupware, conforme destacado na Figura 64 e detalhado na Figura 65. Como discutido na seção 2.3, a realização de estudo de caso influenciou profundamente o desenvolvimento das versões Mediated Chat.

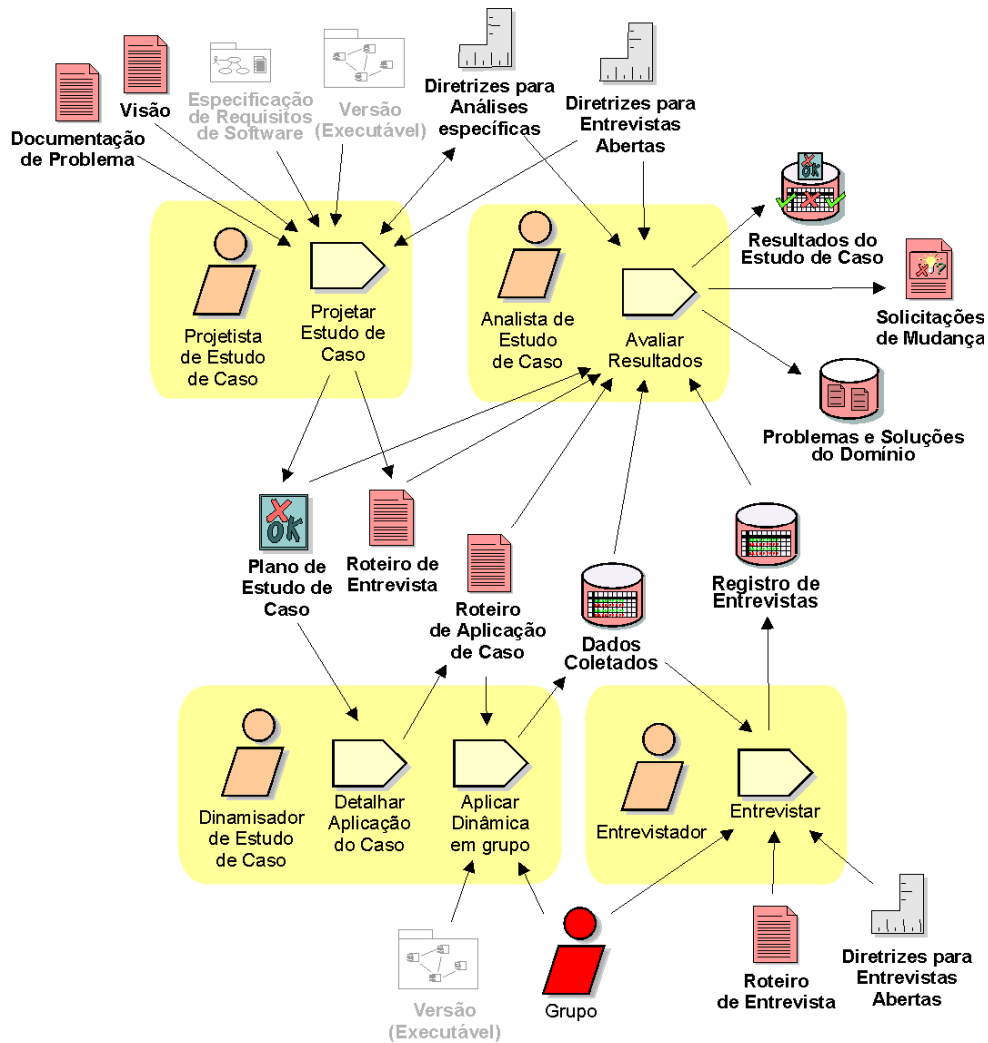


Figura 65. Detalhes do fluxo “Realizar Estudo de Caso” do RUP-3C-Groupware

O **Projetista de Estudo de Caso** deve elaborar um caso para estudar a adequação da versão construída, investigar se a solução implementada de fato resolve ou diminui o problema que está guiando o desenvolvimento da versão. O projetista deve **Projetar Estudo de Caso** considerando: o problema que se encontra documentado no artefato Documentação de Problema; a solução proposta para o problema documentada no artefato Visão; os requisitos documentados em Especificação dos Requisitos de Software (artefato definido no RUP padrão); e a Versão construída do groupware (artefato Build definido no RUP padrão). No artefato **Plano de Estudo de Caso**, o projetista deve especificar como a versão deverá ser usada, por quantas sessões, na realização de qual trabalho, e por qual perfil de grupo. O plano deve ser elaborado de tal maneira que seja possível extrair indícios sobre a ocorrência do problema na nova versão,

sendo esperada a ausência do problema ou uma frequência menor de ocorrência – o anexo 1 contém um exemplo de Plano de Estudo de Caso.

A ocorrência do problema deve ser medida a partir de análises dos resultados obtidos do uso da versão. No artefato Diretrizes para Análises Específicas, encontram-se estabelecidos os procedimentos para extrair indícios da ocorrência do problema. Estes procedimentos podem ter sido estabelecidos em etapa anterior do projeto, em outros projetos ou obtidos da literatura. O projetista deve modificar este artefato na ausência de procedimentos específicos para o problema em questão. Por exemplo, durante o desenvolvimento da versão Mediated Chat 2.0, para investigar a ocorrência do problema “Interrupção da Dinâmica” foram definidos procedimentos específicos derivados da Análise da Conversação (Marcuschi, 1999).

Dinamizador de Estudo de Caso é o papel responsável pela realização do caso a ser estudado. Em função do Plano de Estudo de Caso, o dinamizador deve especificar os detalhes da instanciação do plano durante a atividade **Detalhar Aplicação de Caso**, registrando no artefato **Roteiro de Aplicação de Caso** as configurações do ambiente, do grupo, da tarefa e os procedimentos a serem seguidos. De posse deste roteiro, o dinamizador executa a atividade Aplicar Dinâmica em Grupo, colocando o grupo para usar a versão e coletando dados durante o uso. Os dados a serem coletados podem ser: produtos construídos em grupo (como as sessões de debate no projeto Mediated Chat), dados obtidos da interação (como a seqüência de caracteres digitados ou dos eventos disparados), vídeos dos participantes, ou quaisquer outros tipos de dados especificados pelo projetista do caso.

A realização de entrevistas-abertas (Nicolaci-da-Costa, 2001) mostrou-se útil ao longo do projeto Mediated Chat enquanto método de apoio ao estudo de caso de groupware. No artefato **Diretrizes para Entrevistas Abertas** foram documentados os procedimentos para o projeto, aplicação e análise deste tipo de entrevista. A partir deste artefato, o projetista de estudo de caso deve construir o Roteiro de Entrevista, organizando os tópicos que serão abordados com os entrevistados através de perguntas simples, objetivas e abertas tais como “O que você achou disso?”, “Como foi participar disso?”, “Quais foram as suas dificuldades?” etc. A entrevista deve ser conduzida pelo **Entrevistador** que, durante a execução da atividade **Entrevistar**, deve aprofundar as respostas dos

entrevistados (membros do grupo) mas deve evitar: influenciar os entrevistados, apresentar uma hipótese, elaborar perguntas fechadas, e ‘colocar palavras na boca’ do entrevistado. O entrevistador deve consultar os Dados Coletados para eventualmente fazer perguntas relacionadas a situações ocorridas. As entrevistas são disponibilizadas no artefato Registro de Entrevistas.

O **Analista de Estudo de Caso** fica responsável por analisar os Dados Coletados e o Registro de Entrevistas, seguindo as Diretrizes para Análises Específicas e para Entrevistas Abertas. Para auxiliar a interpretar alguns dos dados coletados, o analista deve consultar os artefatos do planejamento: Plano de Estudo de Caso, Roteiro de Aplicação de Caso, e Roteiro de Entrevista. Durante a atividade **Avaliar Resultados**, o analista deve confrontar os dados coletados com os resultados esperados, gerando um relatório dos **Resultados do Estudo de Caso** e indicar o que deve mudar na versão, se for o caso, no artefato **Solicitações de Mudança**. Os resultados obtidos devem ser resumidos no repositório Problemas e Soluções do Domínio para futuras consultas.

Depois do estudo de caso, a versão pode ser considerada adequada e liberada para a distribuição. Ou então, podem ser identificadas modificações que resultarão num novo ciclo de iteração para a revisão da versão. Ou ainda, novos problemas podem ser identificados podendo dar início a todo um novo ciclo do processo de desenvolvimento.

3.2.6. Distribuição

O fluxo Distribuição do RUP, apresentado na Figura 66, tem por objetivo tornar o produto de software disponível para o usuário-final. Este fluxo não foi alterado no RUP-3C-Groupware; assume-se que a distribuição de um projeto de groupware é igual à de outro software qualquer.

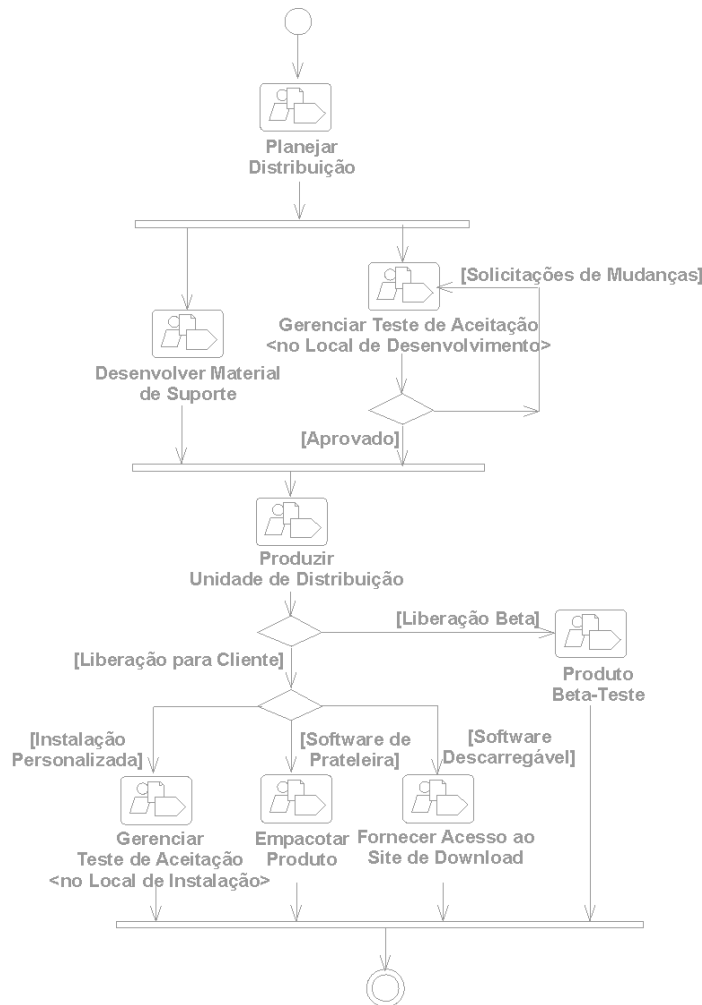


Figura 66. Fluxo de Distribuição do RUP padrão

O detalhe de fluxo Planejar Distribuição é executado logo no início do ciclo de vida do projeto para planejar a distribuição do software e o desenvolvimento de material de treinamento e suporte para o usuário-final. Desenvolver Material de Suporte objetiva a produção de informação para instalação, operação, uso e manutenção do sistema.

A disciplina de Distribuição tem grande ênfase na garantia de que o produto seja bem testado antes de ser liberado para o cliente. O detalhe de fluxo Gerenciar Teste de Aceitação refere-se a dois tipos de ambientes de teste. Primeiramente, a versão precisa ser suficientemente testada no ambiente de desenvolvimento. Depois, precisa ser re-testada no local destino. Após ter sido testada, a versão deve ser liberada para beta-teste. Um programa-beta refere-se ao processo usado por uma organização para solicitar retorno de um subgrupo de usuários sobre o produto em desenvolvimento.

Um cliente pode adquirir o software através de um contrato (instalação personalizada), comprando-o numa loja (software de prateleira), ou baixando-o pela Internet. Para software de prateleira, Empacotar Produto descreve atividades para a produção em massa; embalar o produto de software, o roteiro de instalação e o manual do usuário como qualquer outro produto de consumo. Fornecer Acesso ao Site de *Download* objetiva disponibilizar o produto para compra e *download* pela Internet como um canal de distribuição de *software*.

O RUP-3C-Groupware representa a consolidação e sistematização do que foi aprendido com o desenvolvimento das versões do Mediated Chat e também com o desenvolvimento de outros serviços do ambiente AulaNet. Consiste no esforço para que as boas práticas específicas de desenvolvimento de groupware sejam organizadas num processo de desenvolvimento de software consolidado como o RUP.

Esta tese também dá início a um novo trabalho de pesquisa onde se investiga a adequação do processo aqui elaborado, buscando-se a melhoria contínua. Espera-se que o processo RUP-3C-Groupware já possa ser útil para auxiliar a integração de novos alunos de pós-graduação no grupo de pesquisa Groupware@LES e a integração de novos engenheiros de software na equipe de desenvolvimento do AulaNet. Neste sentido, no capítulo 5 é apresentado o estudo de caso realizado para investigar a repetitividade do processo que foi aqui apresentado.