

# Gerenciamento da Qualidade: uma nova disciplina para o RUP

**Lívia N. Amorim**

[livia.mia@unifor.br](mailto:livia.mia@unifor.br)

[lnjoza@bnb.gov.br](mailto:lnjoza@bnb.gov.br)

**Arnaldo D. Belchior**

[belchior@unifor.br](mailto:belchior@unifor.br)

Universidade de Fortaleza - UNIFOR  
Mestrado em Informática Aplicada – MIA  
Av Washington Soares, 1321  
CEP 60.811.341 – Fortaleza – CE – Brasil

## **Abstract**

As the globalized society is dependent on software more and more, there is a major concern with how to get the software quality. A solution that the organizations have been fetching is the adoption of disciplined approach in order to get the guarantee of the quality of software process like ISO 9001 and CMMI. The RUP is a software engineering process that provides a disciplined approach to assigning tasks and responsibilities in the software life cycle, aiming to ensure the production of quality software. This work analyses the approach of software quality of RUP and proposes a new discipline for itself, the Quality Management, whose objective is establish the action flow that contributes effectively to the quality of software process.

**Keywords:** Software Engineering, Software Quality, Software Quality Assurance, RUP

## **Resumo**

É crescente a dependência da sociedade globalizada em relação ao software, havendo uma maior preocupação em como atingir a qualidade de software. Uma solução que as organizações têm buscado é a adoção de uma abordagem disciplinada para garantir a qualidade de software, baseadas em processos como ISO 9001 e CMMI. O RUP é um processo de engenharia de software, que provê um enfoque disciplinado para designar tarefas e responsabilidades ao longo do ciclo de vida do software, objetivando produzir software de qualidade. Este trabalho analisa a abordagem de qualidade de software do RUP e propõe uma nova disciplina para o mesmo, o Gerenciamento da Qualidade, cujo objetivo é estabelecer o fluxo de ações que contribuem efetivamente para a qualidade do processo de software.

**Palavras-Chave:** Engenharia de Software, Qualidade de Software, Garantia da Qualidade de Software, RUP

## 1. INTRODUÇÃO

Há ainda uma grande inabilidade por parte das organizações em lidar com o complexo processo de desenvolver software. Estatísticas do chamado “Estudo do Caos” demonstram isto: 30% dos processos de software são cancelados antes do término, 189% estouram os custos, e 222% ultrapassam os prazos estabelecidos [26].

À medida que cresce a dependência da sociedade em relação ao software, a qualidade desponta como um fator essencial no desenvolvimento de produtos de software, com uma maior disposição para esse tipo de investimento [7].

As freqüentes discussões sobre a importância da qualidade para o processo de desenvolvimento de software, e para o produto dele resultante, levaram a indústria a se aliar à academia para o desenvolvimento de padrões e modelos, que pudessem garantir a qualidade de software desejada para as suas necessidades [19, 20, 22]. Modelos têm proposto a existência da função de Garantia de Qualidade de Software ou SQA (Software Quality Assurance) [4, 5, 19, 20, 23].

Este trabalho analisa a abordagem de qualidade de software do RUP, e propõe uma nova disciplina (fluxo de trabalho) para o mesmo. A disciplina denominada Gerenciamento da Qualidade objetiva estabelecer o fluxo de ações que contribuam efetivamente para a qualidade do processo e do produto de software. Essa disciplina está baseada no CMMI (Capability Maturity Model Integration) [4; 5], SWEBOK (Software Engineering Book of Knowledge) [1], e em alguns padrões para garantia de qualidade, como o IEEE Std 1028-1997 [10] e IEEE Std 1061-1998 [11].

Este trabalho está organizado como se segue. A seção 2 discorre sobre a função de Garantia da Qualidade de Software. A seção 3 apresenta a visão de qualidade de software do RUP. A seção 4 propõe a nova disciplina para o RUP: o Gerenciamento da Qualidade. A seção 5 mostra o estágio da implantação da proposta. A seção 6 apresenta as conclusões finais.

## 2. A GARANTIA DA QUALIDADE DE SOFTWARE

Garantir a qualidade envolve não apenas a definição do processo de software da organização, mas também a implementação de um processo de garantia da qualidade que seja adequado às suas necessidades, com a realização da garantia da qualidade do processo e do produto [22]. A qualidade do produto de software está estreitamente relacionada com a qualidade do processo que o desenvolveu [3; 25].

A estruturação da função de garantia da qualidade de software ou SQA em uma organização permite acompanhar se o processo de software definido está sendo realmente utilizado e quão aderentes estão as práticas dos projetos ao processo estabelecido. A forma de como essa função é definida depende dos objetivos da organização, assim como do conceito de qualidade de software por ela adotado [17].

Dependendo da política estabelecida pela organização, os responsáveis pela execução de SQA podem trabalhar de forma exclusiva ou em dedicação de tempo parcial [23], tendo esta atribuição institucionalizada na organização por meio da criação do grupo de SQA [19, 20, 23].

A discussão em torno do caráter de disciplina ou da definição da função de SQA na organização é tratada por Baker [2], que chega à conclusão de que é difícil caracterizar SQA como uma disciplina (do ponto de vista da formalização do processo de SQA), devido às diferentes práticas das instituições em relação à SQA, e que nem sempre SQA é uma organização (grupo).

Independente da formalização da função de SQA, a qualidade de software é responsabilidade de todos os envolvidos direta e indiretamente com o processo de software da organização, e possui impacto direto na qualidade do produto gerado. Os gerentes devem estar cômicos da importância da qualidade para se comprometerem com suas atividades e seus resultados. Os atores do processo, como gerentes de projeto, analistas, projetistas e desenvolvedores, devem contribuir para a qualidade em cada ação realizada. Também é importante o papel do cliente e/ou usuário, definindo os requisitos de qualidade para o software, e validando suas especificações em relação ao produto, o que ocorre com mais freqüência em processos de ciclo de vida iterativo [15, 27].

A garantia da qualidade do processo de software visa prover evidências sobre a capacidade do processo em produzir determinado produto, identificando falhas nesse processo e buscando resolvê-las antes que impactem no produto. A garantia da qualidade do produto de software, por sua vez, visa garantir que o software produzido esteja em conformidade com requisitos funcionais e de desempenho especificados; atenda aos padrões de desenvolvimento documentados; seja o mais isento possível de erros; e atenda às características implícitas esperadas pelo usuário.

As atividades de qualidade do processo de software compreendem a definição do processo, a verificação da utilização do processo definido, e a preocupação com a melhoria do processo utilizado. Primeiramente, o processo da organização deve ser definido, atendendo os objetivos da organização. No CMMI, por exemplo, este trabalho é conduzido pela equipe de processo de engenharia de software (SEPG), com a participação da equipe de SQA [4, 5, 19, 20]. Enquanto o SEPG é responsável pelo processo, o SQA verifica a aderência das práticas de engenharia de software dos projetos ao processo definido.

São utilizadas técnicas de SQA para verificar se o processo definido está sendo seguido, para encontrar defeitos diretamente no produto, ou para indicar a necessidade de um exame mais detalhado no mesmo. As principais técnicas utilizadas são as Revisões de Software, que são classificadas como Auditorias, Inspeções, Walkthroughs, Revisões Técnicas e Revisões Gerenciais [10]. As Revisões de Software avaliam o processo de desenvolvimento, seu gerenciamento, e o produto de software [1, 10]. Além das Revisões, são também realizados Testes, os quais constituem um dos elementos críticos da garantia de qualidade de software e representam a última revisão das especificações de projeto e código do produto de software.

Quando o foco é a melhoria do processo estabelecido, são utilizados as Avaliações (Assessment), que examinam um processo para determinar a sua capacidade, por meio da comparação desse processo em relação a um padrão de melhores práticas [8, 19, 20].

Alguns modelos e normas internacionais tratam a qualidade do processo de software, e trazem, em seu escopo, a figura da função de qualidade. O CMMI [4, 5] serve para determinar a capacidade e maturidade de uma organização para a produção de software. A norma ISO 9001 trata da definição, implementação e manutenção do Sistema de Gestão da Qualidade para uma organização [13]. A ISO/IEC 15504 constitui-se de um framework para modelos de avaliação de processos de software, podendo ser utilizado também em estratégias para melhoria de processos [14].

A literatura em qualidade de software relata ainda algumas propostas para formalização do processo de qualidade de software. Marczak et al. [17] propõem um modelo de organização da função de SQA, considerando aspectos práticos da implantação da garantia de qualidade em uma organização de desenvolvimento de software, no contexto de implantação do CMM-SW [17], baseado na área chave de processo SQA desse modelo. Unhelkar [27] estabelece um processo de qualidade de software para organizações que trabalhem com projetos baseados na UML (Unified Modeling Language) [27]. Segundo Unhelkar [27], a arquitetura do processo de qualidade de software é constituída por três componentes: a Gestão de Qualidade, a Garantia da Qualidade, o Controle de Qualidade.

No mesmo enfoque de tratar a qualidade do processo de desenvolvimento, para que se gerem produtos de qualidade, surgiu a abordagem do framework RUP, que visa assegurar o desenvolvimento de produtos de software de qualidade, que satisfaçam as necessidades de seus usuários finais, dentro de cronogramas e orçamentos previsíveis. Para alcançar este objetivo, o RUP diz utilizar-se de melhores práticas em desenvolvimento de software, de forma a ser adaptado para uma grande variedade de projetos e organizações. São elas [15]: desenvolver software iterativamente; gerenciar requisitos; usar arquitetura baseada em componentes; modelar software visualmente; verificar a qualidade de software; controlar mudanças no software [15, 21].

### 3. A VISÃO DE QUALIDADE DE SOFTWARE NO RUP

O RUP é uma abordagem de desenvolvimento de software iterativa. É um processo de engenharia de software, que estabelece o que deve ser feito, como deve ser feito, quando e por quem, abrangendo o ciclo de vida de software de um projeto. Provê um framework de processos de software, podendo ser utilizado por uma organização em seus projetos de desenvolvimento de software, por meio da adaptação desses processos à sua realidade [15].

O RUP compõe-se de duas dimensões ou estruturas: uma estática e outra dinâmica, que podem ser visualizadas na Figura 1.

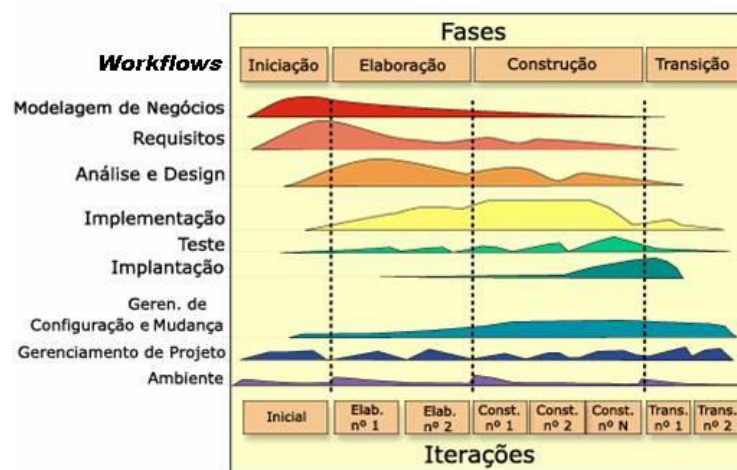


Figura 1: Dimensões do RUP

A dimensão estática representa a estrutura estática do processo, descrevendo como os elementos do processo são agrupados logicamente em disciplinas (workflows). Disciplinas são agrupamentos lógicos de papéis, atividades, artefatos e outros guias para a descrição de um processo, e são representadas por um fluxo de trabalho.

Tabela 1: Elementos do processo do RUP

Elemento	Conceito
Papel	Definição do comportamento e das responsabilidades de um indivíduo ou grupo de indivíduos trabalhando juntos em equipe
Atividade	É uma unidade de trabalho, executada por um papel com um propósito definido, geralmente criando ou atualizando um artefato. Pode ser dividida em passos ( <i>steps</i> ).
Artefato	Um pedaço de informação que é produzido, modificado ou criado por um processo. São usados como entrada na execução de atividades e como produto das atividades executadas.
Conceito	Introduzem definições e princípios chave.
Guia	Prover regras, recomendações e heurísticas que suportam atividades, passos e artefatos.

A dimensão dinâmica do RUP é representada pelo tempo e expressa o processo por meio de ciclos, decompostos em fases, que são divididas em iterações com marcos de conclusão. Essas fases são: Concepção, Elaboração, Construção, e Transição [18]. Cada fase possui objetivos de negócio e contém iterações que focam na produção de artefatos técnicos para atingir aqueles objetivos. As descrições de cada fase são sumarizadas na Tabela 2.

No que diz respeito à qualidade de software, a visão apresentada pelo RUP vem evoluindo em relação à qualidade do processo e à qualidade do produto. Nele, a qualidade é responsabilidade de todos os envolvidos no processo, mas o papel do gerente de projeto possui destaque como responsável pela gestão da qualidade [15, 21]. A qualidade do processo no RUP é implementada ao longo das disciplinas, fases e iterações do ciclo de vida do projeto, estando especialmente ligada à disciplina de Gerenciamento do Projeto. O gerente de projeto é o papel responsável por estabelecer as atividades relacionadas à qualidade de software no projeto. Há revisões técnicas e revisões gerenciais. As primeiras são realizadas sobre os artefatos, como requisitos, modelos de casos de uso e código. As outras são conduzidas pelo gerente de projetos, para avaliar o status das atividades de projeto pelo menos uma vez a cada iteração, e pelo PRA (Project Review Authority) para revisar a conformidade do projeto com obrigações contratuais e com padrões organizacionais [21].

Tabela 2 - Fases do RUP

Fase	Descrição
Concepção	Estabelece uma boa compreensão do sistema a ser construído, por meio de requisitos de alto nível e do estabelecimento do escopo do projeto. Obter acordo com os envolvidos ( <i>stakeholders</i> ), para prosseguir ou não com o projeto, também caracteriza essa fase. Também é preparado o ambiente para o projeto, estabelecendo o caso de desenvolvimento.
Elaboração	O propósito desta fase é compreender como construir, analisando o domínio do problema, estabelecendo uma arquitetura básica, desenvolvendo um plano de projeto, e eliminando os maiores riscos do projeto.
Construção	Os componentes da solução são desenvolvidos e integrados em um produto. Custos, cronograma e qualidade são otimizados pelo gerenciamento dos recursos e o controle das operações. Os componentes são desenvolvidos e testados baseado em critérios de avaliação definidos. <i>Releases</i> do produto também são avaliadas.
Transição	O principal objetivo desta fase é construir a versão final do produto e disponibilizá-la para os usuários finais. São desenvolvidos os materiais de suporte à instalação e uso pelo usuário final.

Também relacionado à qualidade do processo, há o papel do SEPA (Software Engineering Process Authority), que auxilia a organização na identificação dos processos dos projetos, avaliando-os continuamente com o intuito de melhorá-los. Para estabelecer o processo a ser utilizado no projeto, o RUP é adaptado para o projeto pelo engenheiro de processos, pelo gerente do projeto ou até mesmo por um mentor, que é alguém com experiência em implementações do Processo Unificado [21].

No entanto, a ênfase de qualidade no RUP está voltada para o produto. O RUP aborda a qualidade do produto como sendo algo verificável por meio dos testes realizados nos releases ao longo do ciclo de vida do projeto. Como a abordagem é iterativa e os testes são realizados em cada fase, a qualidade do produto é verificada continuamente, sendo possível identificar defeitos mais cedo [15]. Doria [6] afirma que existem amplas referências na literatura de que, dependendo do caso, as inspeções podem ser mais efetivas que os testes e a um custo bem menor.

O modelo FURPS+ [9], aplicado para a caracterização de requisitos, é usado no RUP também para avaliar a qualidade do produto. O RUP faz uma distinção entre os dois tipos de produtos: o software executável, versão final disponibilizada ao cliente, e os produtos de software, que resultam do processo de desenvolvimento, como os planos, as especificações e outros artefatos (byproducts), que estão associados ao processo [15].

Apesar de considerar a importância da qualidade do processo e das atividades de revisão, Kroll & Kruchten [15] afirmam que para o RUP, “at the end of the day, what really counts is how good your code is, not how good your byproducts of software development are”.

Os principais artefatos relacionados à qualidade de software no RUP [21] são:

- Plano de Garantia de Qualidade: elaborado por projeto e estabelecido pelo gerente do projeto, para registrar as ações de qualidade previstas e realizadas, especificando como assegurar a qualidade do produto, dos artefatos e do processo.
- Caso de Desenvolvimento: elaborado por projeto, contém a definição do processo a ser seguido pelo projeto, a partir das adaptações do RUP.
- Registro de Revisão: captura os resultados da atividade de revisão no qual um ou mais artefatos dos projetos são revisados. Segundo o RUP [21], “é mais importante realizar a revisão do que documentá-la”, sugerindo que um e-mail sirva como registro da revisão.

Alguns pesquisadores apontam deficiências no RUP na abordagem de qualidade de software. Manzoni & Price [16] analisam que o RUP não estabelece como tratar os problemas identificados nas revisões e nem como os resultados das revisões são comunicados aos demais envolvidos com o projeto de software e aos envolvidos com o processo de software.

Para Smith [24], que analisou as características do RUP em relação ao padrão IEEE 1074 [12], o RUP possui várias carências abrangendo: a formalização das avaliações de qualidade, especialmente em termos dos tipos de revisões realizadas; a ausência de aprovação de desvios identificados em relação ao processo padrão (framework do RUP); e ao uso de avaliações com foco na melhoria do processo de software utilizado. Esta análise sugere que os aspectos de qualidade abordados na disciplina de testes, que são bem cobertos, sejam aliados à abordagem das avaliações de revisão ao longo do processo, que são tratadas informalmente no RUP, e constituam uma nova disciplina de avaliação.

Neste contexto, este trabalho propõe uma nova disciplina para o RUP: o Gerenciamento da Qualidade, cujo objetivo é estabelecer o fluxo de ações que conduzam à qualidade do processo de software, suprimindo algumas das deficiências encontradas no RUP em relação à garantia da qualidade de software.

#### 4. GERENCIAMENTO DA QUALIDADE: DISCIPLINA DE QUALIDADE DE SOFTWARE NO RUP

No RUP, a qualidade do produto é tratada de forma abrangente pela disciplina de Testes, e é não dada a relevância necessária ao tratamento dado à qualidade do processo de software. Neste contexto, é proposta a disciplina de Gerenciamento da Qualidade para o RUP, que visa estabelecer o fluxo de ações que contribuam efetivamente para a qualidade de software, por meio do planejamento, da garantia, e da monitoração da qualidade ao longo do processo de desenvolvimento.

A disciplina de Gerenciamento da Qualidade (GQ) está estruturada de modo a atender os seguintes objetivos:

- Prover visibilidade apropriada para a qualidade do processo de desenvolvimento no RUP, destacando a importância da mesma em cada iteração.
- Sistematizar e formalizar ações de qualidade do processo de software, executadas mediante o planejamento da qualidade realizado em paralelo com o planejamento do projeto. São estabelecidas as principais técnicas de SQA, conforme o padrão IEEE Std 1028-1997, que são utilizadas durante as iterações para verificar a qualidade do processo de software, suprimindo a deficiência relatada por Smith [24].
- Tratamento sistemático das não conformidades, facilitando sua identificação, seu relato aos envolvidos, e a identificação de ações corretivas e preventivas (quando possível). Isto não abordado no RUP, segundo Manzoni & Price [17].
- Possibilitar que as ações de qualidade estejam alinhadas aos objetivos de qualidade especificados para a organização. A execução das atividades de SQA é realizada de acordo com o planejamento da qualidade de software para o projeto e para a iteração. O planejamento da qualidade, por sua vez, considera os objetivos

de qualidade especificados para a organização, para o projeto e ainda para a iteração. Isso possibilita que as ações em prol da qualidade sejam realizadas de forma contínua, estritamente alinhadas aos objetivos estabelecidos, e tendo seu alcance monitorado.

- Permitir a utilização de métricas de processo relacionadas aos objetivos de qualidade para a iteração, para o projeto e para a organização.
- Contribuir para as ações de melhoria do processo de software no RUP. A disciplina proposta estabelece o registro das lições aprendidas ao longo do processo. Esta atividade, aliada ao uso sistemático de métricas, facilita a identificação da aderência de boas práticas de engenharia de software do projeto em relação ao processo configurado a partir do framework do RUP, municiando a organização e, especialmente, o SEPA (Software Engineering Process Authority) e o Engenheiro de Processos do RUP com as informações necessárias para propor a instância de processo mais adequada às necessidades do projeto.

Para que a nova disciplina proposta para o RUP, Gerenciamento da Qualidade, exerça efetivamente seu papel nos projetos de software, urge que seja estabelecido um conjunto de atividades de “gerenciamento da qualidade” no âmbito organizacional. Essas atividades devem ser realizadas necessariamente antes da instanciação da disciplina ao nível de projeto. As principais atividades que devem ser realizadas no âmbito institucional são:

- definição dos objetivos de qualidade e de políticas de qualidade para a organização;
- estabelecimento de procedimentos, padrões e métricas institucionais de qualidade;
- especificação de papéis e da função da qualidade na organização;
- formatação do perfil para os profissionais envolvidos no processo de software;
- realização de treinamentos para os envolvidos no processo, para a disseminação dos conceitos de qualidade para a organização;
- o estabelecimento de controles sobre o custo das atividades de qualidade nos projetos.

Para a disciplina Gerenciamento da Qualidade são propostos dois novos papéis para o RUP: o Gerente de qualidade, e o Engenheiro de qualidade.

O Gerente de Qualidade deve administrar a função de qualidade de software da organização, mantendo os objetivos organizacionais da qualidade alinhados aos objetivos de negócios organizacionais e avaliando se os projetos estão alcançando os objetivos de qualidade. Deve possuir experiência com gestão de processos e de pessoas, conhecimento da organização e conhecimento sobre modelos e padrões de qualidade de software.

O Engenheiro de Qualidade deve ter forte embasamento em engenharia de software, mormente em modelos, padrões e técnicas de SQA, e como estes elementos estão relacionados com o processo de desenvolvimento de software.

Ambos, Gerente de Qualidade e Engenheiro de Qualidade, devem estar cientes do papel da função de qualidade de software no contexto organizacional e dos projetos, podendo agir como consultores de engenharia de software para projetos da organização, e provendo orientação a respeito dos padrões organizacionais e de qualidade.

O fluxo da disciplina Gerenciamento da Qualidade é apresentado na Figura 2, e possui as macro-atividades a seguir: (i) Planejar Qualidade do Projeto; (ii) Planejar Qualidade da Iteração; (iii) Garantir Qualidade; (iv) Monitorar Qualidade; e (v) Registrar Lições Aprendidas. O conjunto de atividades referentes aos itens (ii), (iii) e (iv) deve ocorrer para cada iteração do RUP ao longo do processo de desenvolvimento. A seguir, são explanados os detalhes de cada parte do fluxo da disciplina.

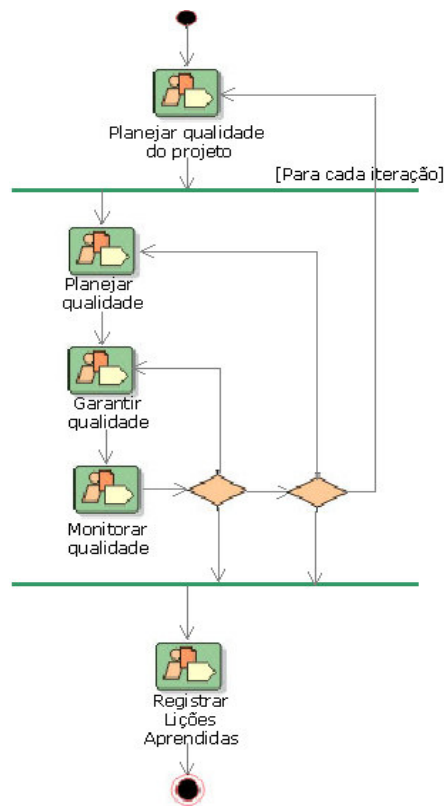


Figura 2 - Fluxo da disciplina Gerenciamento da Qualidade

A Figura 3 detalha as atividades do fluxo Planejar Qualidade do Projeto, que devem ser executadas em paralelo com as atividades de planejamento do próprio projeto. Nesse fluxo, são definidos os objetivos de qualidade de software para o projeto, com base nos requisitos do projeto, plano de desenvolvimento e padrões de qualidade da organização. São estabelecidas as métricas de qualidade associadas a cada objetivo de qualidade especificado, para que seja possível monitorar se as metas foram atingidas.

O Engenheiro de qualidade planeja, a partir dos objetivos de qualidade, que atividades de qualidade serão executadas ao longo do projeto e constrói o cronograma dessas atividades. Estas informações devem ser registradas no Plano de Garantia de Qualidade de Software do projeto. Ele também pode orientar integrantes do projeto a respeito dos padrões de qualidade adotados e das práticas de engenharia de software aplicáveis.

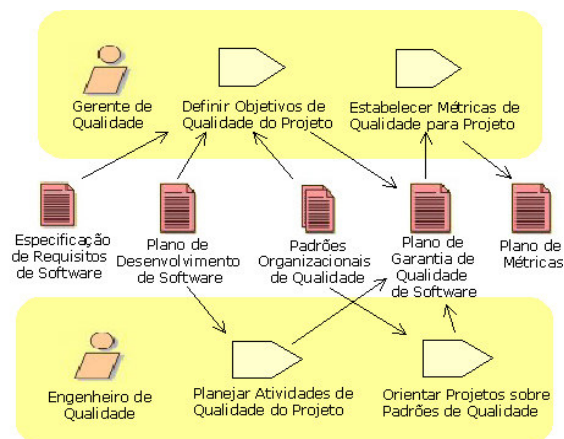


Figura 3 - Detalhe do fluxo: Planejar Qualidade do Projeto

A Figura 4 detalha o fluxo Planejar Qualidade na Iteração. Ao se iniciar uma iteração, devem ser identificados, dentre os objetivos de qualidade do projeto levantados na macro-atividade anterior, quais aqueles que serão abordados durante a iteração. Em seguida, deve ser elencadas as atividades de qualidade de software a serem realizadas durante a iteração, atualizando-se o Plano de Garantia de Qualidade.

A partir da segunda iteração, com os dados provenientes da macro-atividade de Monitorar Qualidade, podem ser especificadas as ações preventivas para as não conformidades (reconhecimento da falta de aderência de determinado item sob garantia de qualidade em relação aos padrões de qualidade) identificadas na iteração anterior, minimizando assim a ocorrência de não conformidades.

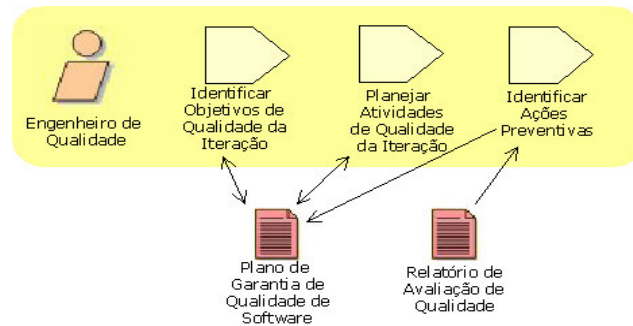


Figura 4 - Detalhe do fluxo: Planejar Qualidade na Iteração

A Figura 5 apresenta o detalhamento do fluxo Garantir Qualidade na iteração, que é realizada através de revisões de atividades do processo, por meio da avaliação de artefatos. As revisões e avaliações são conduzidas a partir do padrão IEEE 1028 [10]. O Relatório de avaliação da qualidade do projeto registra as informações sobre as revisões e avaliações realizadas, os itens avaliados, as não conformidades identificadas, o grau de severidade a elas associadas, o responsável pela resolução da não conformidade, assim como os prazos para que isso seja executado. Esse relatório formaliza o Registro de Revisão do RUP. As não conformidades identificadas e reportadas devem ser acompanhadas pelo Engenheiro de Qualidade até seu desfecho. Ainda são coletadas métricas de qualidade do projeto, e são fornecidas orientações (quando necessário) sobre padrões e práticas de engenharia de software.

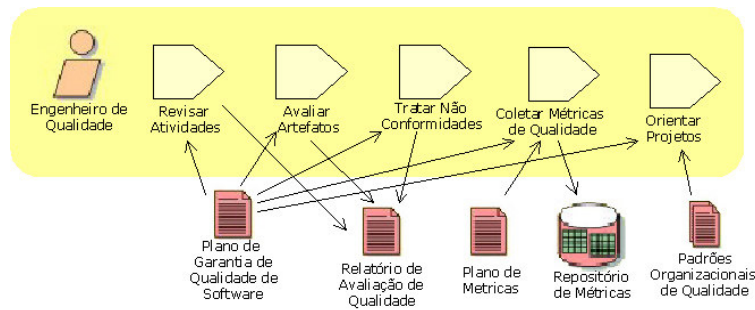


Figura 5 - Detalhe do fluxo: Garantir Qualidade

A Figura 6 apresenta o detalhamento do fluxo Monitorar Qualidade na iteração. O Engenheiro de qualidade analisa dados sobre as atividades de qualidade realizadas (revisões de atividades, avaliações de artefatos e tratamento de não conformidades) e sobre as métricas coletadas, com o objetivo de adequar ou melhorar o processo de qualidade, e permitir que ações preventivas por ser empreendidas.

O Gerente de qualidade avalia o alcance dos objetivos de qualidade da iteração e do projeto, e o desempenho da equipe de qualidade alocada ao projeto. O Relatório de garantia de qualidade é o artefato por meio do qual são reportadas as atividades realizadas pela equipe de qualidade para todos os envolvidos com o processo, dando visibilidade ao trabalho da equipe de qualidade e à importância da função de qualidade de software para a organização e para os projetos.



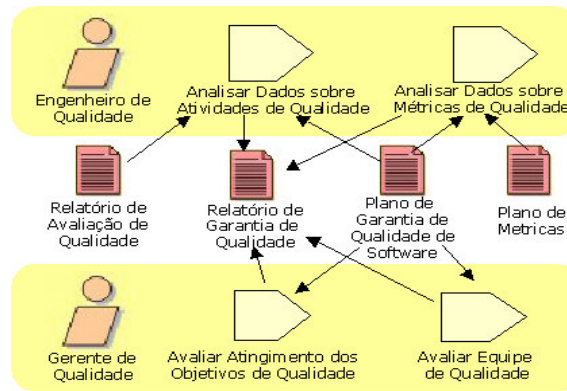


Figura 6 - Detalhe do fluxo: Monitorar Qualidade

Ao final do projeto, são registradas as lições aprendidas (Figura 7) do processo de qualidade do projeto. Essas lições podem:

- auxiliar o Engenheiro de processo, na disciplina de Ambiente, a selecionar a instância do framework do RUP, que constituirá o processo mais adequado para projetos similares, atualizando o artefato Caso de Desenvolvimento;
- identificar tipos de não conformidades mais comuns e como elas podem ser prevenidas e corrigidas;
- possibilitar avaliar como determinados tipos de não conformidades podem ser resolvidos e em que espaço de tempo;
- ajudar a quantificar a equipe de qualidade adequada por tipo de projeto, dentre outras informações válidas.

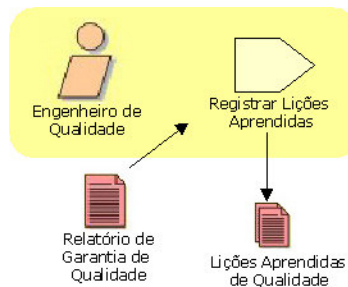


Figura 7 - Detalhe do fluxo: Registrar Lições Aprendidas

Cada uma das atividades constantes da disciplina de Gerenciamento da Qualidade é detalhada em termos de propósitos, passos, artefatos de entrada e de saída, frequência de execução da atividade, e atores envolvidos. No Quadro 1, é descrita a atividade Definir objetivos de qualidade do projeto, do fluxo Planejar Qualidade do Projeto (Figura 3).

<b>Atividade: Definir objetivos de qualidade do projeto</b>	
<b>Propósito:</b> estabelece os objetivos de qualidade de <i>software</i> para o projeto, observando as definições organizacionais para a qualidade e as características específicas do projeto (requisitos legais, padrões do cliente do projeto etc.)	
<b>Passos:</b> - verificar os padrões organizacionais de qualidade (Políticas, padrões, procedimentos etc); - identificar requisitos de qualidade do projeto (requisitos do cliente, legislação etc.); - definir os objetivos de qualidade de <i>software</i> do projeto, orientando os projetos quanto às suas necessidades de qualidade de <i>software</i> .	
<b>Artefatos de Entrada:</b> - Padrões organizacionais de qualidade; - Plano de Desenvolvimento de <i>Software</i> ; - Especificação de Requisitos de <i>Software</i> ; e FURPS+.	<b>Artefatos de Saída:</b> - Plano de Garantia de Qualidade de <i>Software</i> .
<b>Frequência:</b> a cada reunião de planejamento do projeto.	
<b>Ator:</b> Gerente de Qualidade	

Quadro 1 - Definir objetivos de qualidade do projeto

## 5. IMPLEMENTAÇÃO DA DISCIPLINA GERENCIAMENTO DA QUALIDADE

A idéia da formalização da função de SQA no RUP nasceu a partir da experiência dos autores com a implementação do RUP em uma organização que, simultaneamente à implantação do RUP, também estava buscando implantar a função de SQA incorporada à Gestão da Qualidade. Essa organização atualmente já é certificada CMM nível 2.

Uma segunda organização, na qual está sendo implementada a nova disciplina proposta (Gerenciamento de Qualidade) adotou o framework do RUP na definição de seus processos de software e está buscando a certificação CMM nível 2.

A implementação da função de SQA nesta organização está utilizando a disciplina proposta como processo de qualidade de software. O processo de implementação está em andamento, no estágio de realização das atividades em âmbito organizacional, tendo já sido definidos os objetivos da qualidade, Procedimentos de Qualidade e a Política de Qualidade da organização. Estão sendo definidos os padrões organizacionais de qualidade.

Foram especificados os papéis necessários ao processo de desenvolvimento e estão sendo realizados treinamentos para disseminação da cultura e conceitos de qualidade na organização.

Foi criada uma unidade organizacional para englobar a função de SQA, que é independente da estrutura que contém os projetos que serão avaliados em relação à qualidade, garantindo a independência de atuação do Engenheiro de qualidade e Gerente de qualidade.

Com a conclusão dos treinamentos e da revisão dos padrões, as atividades da disciplina serão instanciadas em alguns projetos selecionados como piloto. Já é possível observar a importância da formalização da disciplina de qualidade de software no dia-a-dia da organização, por meio da identificação de não conformidades e dos pontos de controle que precisam ser melhorados no processo de software utilizado.

## 6. CONCLUSÃO

A proposta de inclusão no RUP da disciplina de Gerenciamento da Qualidade traz como principais contribuições:

- Formalização de um processo de qualidade de software, dentro de um contexto de ciclo de vida iterativo / incremental, para projetos que sigam o processo de desenvolvimento baseado no framework do RUP;
- Fortalecimento da perspectiva de qualidade do processo de software do RUP, formalizando ações e alinhando-as a um processo de gestão da qualidade;
- Expansão da visão de qualidade do produto, que era mais focada no software executável, também para os demais produtos de software (byproducts) produzidos ao longo do processo de desenvolvimento.
- Estabelecimento do uso e da análise de métricas mais alinhadas aos objetivos da qualidade.
- Enfatiza ações que trabalham a melhoria do processo de software para o projeto, e não apenas as mudanças de configurações de processo, possibilitando o uso do processo melhorado em projetos similares.

- Pode auxiliar a organização na estruturação de seus processos rumo à certificação CMMI nível 2, já que a proposta considera várias práticas exigidas por este modelo.

Como trabalhos futuros, pode-se elencar a avaliação das características de qualidade de software do RUP em relação a modelos de Qualidade de Software, como o CMMI e a ISO 15504, a fim de possibilitar às organizações que utilizam o RUP usufruírem uma implementação robusta de SQA e da Gestão da Qualidade.

### Referências Bibliográficas

- [1] A Guide for Software Engineering Body of Knowledge, IEEE – Trial (Version 0.95) – May, 2001.
- [2] Baker, Emanuel R., Which Way, SQA? IEEE Software January/February 2001.
- [3] Conradi, R. & Fuggetta, A., Improving Software Process Improvement. IEEE Software July/August 2002.
- [4] CMMI Product Team. CMMI for Systems Engineering/Software Engineering, Version 1.1 Staged Representation (CMU/SEI-2002-TR-029, ESC-TR-2002-029). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, August, 2002.
- [5] CMMI Product Team. CMMI for Systems Engineering/Software Engineering, Version 1.1 Continuous Representation (CMU/SEI-2002-TR-028, ESC-TR-2002-028). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, August, 2002.
- [6] Dória, Emerson, Replicação de estudos empíricos em engenharia de software. 1v. 154p. Dissertação de Mestrado. Universidade de São Paulo/São Carlos, 2001.
- [7] Duarte, Cristina & Falbo, R.A., Uma ontologia de qualidade de software. Workshop de Qualidade de Software, João Pessoa, p. 275-285, Outubro de 2000.
- [8] Dunaway, D. & Masters, S., CMM-Based Appraisal for Internal Process Improvement (CBA IPI): Method Description (CMU/SEI-96-TR-007). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, April 1996.
- [9] Grady, Robert, Practical Software Metrics for Project Management and Process Improvement. Prentice-Hall, 1992.
- [10] IEEE Std 1028–1997, IEEE Standard for Software Revisions.
- [11] IEEE Std 1061–1998, IEEE Standard for a Software Quality Metrics Methodology.
- [12] IEEE Std 1074-1997, IEEE Standard for Developing Software Life Cycle Process.
- [13] ISO 9001, Sistemas de Gestão da Qualidade – Requisitos. 2000.
- [14] ISO/IEC TR 15504, Software Process Assessment, 2003.
- [15] Kroll, Per & Kruchten, Philippe, The Rational Unified Process made easy: a practitioner’s guide to the RUP. Pearson Education, 2003.
- [16] Manzoni, Lisandra V. & Price, Roberto T., Identifying Extensions required by RUP (Rational Unified Process) to comply with CMM (Capability Maturity Model) Levels 2 and 3. IEEE Transactions on Software Engineering, Vol. 29. No. 2, Fevereiro de 2003.
- [17] Marczak, S., Sá, L. Ceccato, I., Audy, J. & Antunes, D., Uma proposta de organização e funcionamento da função de garantia de qualidade de software em um contexto de implantação do SW-CMM. II Simpósio Brasileiro de Qualidade de Software. Fortaleza, Brasil. 2003.
- [18] Mohaghegi, Parastoo, Software Engineering Processes RUP and XP. Lecture Notes to IKT-2340 “Open Systems Seminar” at Hogskolen Agder, 2002. Disponível em <http://fag.grm.hia.no/ikt2340/year2002/themes/process/notes/2-RUP-XP.pdf>. Acessado em janeiro de 2004.
- [19] Paulk, M. C.; Curtis, B.; Chrissis, M. B.; & Weber, C. V., Capability Maturity Model for Software. Version 1.1 (CMU/SEI-93-TR-024, ADA 263403). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 1993.
- [20] Paulk, M. C. et. Al., Key Practices of the Capability Maturity Model, Version 1.1. 1993.
- [21] Rational Software Corporation, Rational Unified Process, Version 2003.06.00.65, CD-ROM, Rational Software, Cupertino, California, 2003.

- [22] Rocha, A. R. C. et al., Qualidade de software: Teoria e prática. Prentice Hall, 2001.
- [23] Schulmeyer, Gordon & McManus, James, Handbook of Software Quality Assurance. New Jersey: Prentice Hall, 1999.
- [24] Smith, John., Inconsistencies found in RUP Test Drive with IEEE 1074. IBM by Chuck Walrad of Davenport Consulting, Inc., 2003. Disponível em: [http://www.p1074-workgroup.org/3M\\_Compatibility\\_Team/references/Annex%20A%20table-RUP.doc](http://www.p1074-workgroup.org/3M_Compatibility_Team/references/Annex%20A%20table-RUP.doc). Acessado em dezembro de 2003.
- [25] Sommerville, Ian, Software Engineering. 6ª ed. Addison-Wesley Pub Co, 2000.
- [26] The Chaos Study, The Standish Group International, Inc., Dennis, MA. 1994.
- [27] Unhelkar, Bhuvan, Process Quality Assurance for UML-based projects. In: The Addison-Wesley.