UNIVERSIDADE FEDERAL DE PERNAMBUCO

CENTRO DE INFORMÁTICA

**Aplicando a Abordagem GQM para Avaliar o Impacto da Adoção da Metodologia Ágil Scrum**

Autor: Philippe Pereira das Neves

**Trabalho de Graduação do Curso de Ciência da Computação**

Orientador: Judith Kelner

Recife, 07 de julho de 2012

**Philippe Pereira das Neves**

**Aplicando a Abordagem GQM para Avaliar o Impacto da Adoção da Metodologia Ágil Scrum**

Trabalho de conclusão de curso apresentado

como parte das atividades para obtenção do

título de Bacharel em Ciência da

Computação do Centro de Informática da

Universidade Federal de Pernambuco

Professor Orientador: Judith Kelner

Recife, 2012

**Agradecimentos**

À minha orientadora Judith Kelner, pela paciência e assistência durante este trabalho.

À minha namorada Adnna Fernades, pelo incentivo, carinho e compreensão.

Ao amigo Wesley Davison, pelo apoio e motivação nos estudos e, principalmente, pela amizade.

Ao amigo Tarcício Amorim Farias, gerente da unidade TST Garantia Veicular Recife-PE, pelo total apoio e confiança na minha capacidade, permitindo a realização do estudo de caso no projeto de desenvolvimento do sistema de gerenciamento da empresa.

Muito Obrigado!

Vocês foram essenciais para a conclusão deste trabalho.

Conteúdo

[1. Introdução 6](#_Toc331020786)

[1.1. Contexto 6](#_Toc331020787)

[1.2. Objetivos 6](#_Toc331020788)

[1.3. Organização do documento 6](#_Toc331020789)

[2. Scrum 8](#_Toc331020790)

[2.1. Visão Geral 8](#_Toc331020791)

[2.2. Papéis e Responsabilidades 9](#_Toc331020792)

[2.2.1. Scrum Master 9](#_Toc331020793)

[2.2.2. Scrum Team 9](#_Toc331020794)

[2.2.3. Product Owner 9](#_Toc331020795)

[2.2.4. Etapas do Sprint 9](#_Toc331020796)

[2.2.5. Sprint Planning 9](#_Toc331020797)

[2.2.6. Daily Scrum 10](#_Toc331020798)

[2.2.7. Sprint Review 10](#_Toc331020799)

[2.2.8. Sprint Retrospective 10](#_Toc331020800)

[2.3. Artefatos 10](#_Toc331020801)

[2.3.1. Product Backlog 10](#_Toc331020802)

[2.3.3. Sprint Backlog 11](#_Toc331020803)

[2.3.4. Impediment Backlog 11](#_Toc331020804)

[2.3.5. Burndown Chart 11](#_Toc331020805)

[3. Mensuração em Projetos de Software 13](#_Toc331020806)

[3.1. Mensuração para Compreender Produtos e Processos de Software 13](#_Toc331020807)

[3.2. Mensuração para Gerenciar Projetos de Software 13](#_Toc331020808)

[3.3. Mensuração para Guiar o Melhoramento 13](#_Toc331020809)

[4. Abordagem Goal Question Metric 14](#_Toc331020810)

[4.1. Etapas do GQM 15](#_Toc331020811)

[4.1.1. Estudo Prévio 16](#_Toc331020812)

[4.1.2. Identificação de Metas GQM 16](#_Toc331020813)

[4.1.3. Desenvolvimento do Plano GQM 17](#_Toc331020814)

[4.1.4. Desenvolvimento do Plano de Coleta 18](#_Toc331020815)

[4.1.5. Coleta de Dados 19](#_Toc331020816)

[4.1.6. Análise dos Dados 20](#_Toc331020817)

[4.1.7. Interpretação dos Dados 20](#_Toc331020818)

[4.1.8. Captura de Experiências 21](#_Toc331020819)

[5. Estudo de caso 22](#_Toc331020820)

[5.1. A Empresa 22](#_Toc331020821)

[5.2. O Projeto Selecionado 22](#_Toc331020822)

[5.3. Adoção do SCRUM 22](#_Toc331020823)

[5.3.1. Acúmulo de Responsabilidades 22](#_Toc331020824)

[5.3.2. Duração do Sprint 23](#_Toc331020825)

[5.3.3. Sprint Review 23](#_Toc331020826)

[5.3.4. Sprint Retrospective 23](#_Toc331020827)

[5.3.5. Ferramenta de Apoio 23](#_Toc331020828)

[5.4. Aplicação da Abordagem GQM no Projeto 23](#_Toc331020829)

[5.4.1. Estudo prévio 24](#_Toc331020830)

[5.4.2. Identificação de Metas GQM 25](#_Toc331020831)

[5.4.3. Desenvolvimento do Plano GQM 26](#_Toc331020832)

[5.4.4. Desenvolvimento do plano de Mensuração 32](#_Toc331020833)

[5.4.5. Coleta dos dados 32](#_Toc331020834)

[5.4.6. Análise e Interpretação dos dados 33](#_Toc331020835)

[5.4.7. Captura de Experiência 36](#_Toc331020836)

[6. Conclusão 38](#_Toc331020837)

[6.1. Contribuições 38](#_Toc331020838)

[6.2. Trabalhos Futuros 38](#_Toc331020839)

[Referências 40](#_Toc331020840)

# Introdução

Software mais barato, desenvolvido mais rápido, com menos documentação, mais conformidade com requisitos, melhoria de resultados em curto prazo e implantação barata é um atrativo das metodologias ágeis para pequenas organizações. No entanto, muitas organizações desenvolvem software sem usar nenhum processo ou método. Geralmente isso ocorre porque os processos tradicionais não são adequados às realidades das organizações. Em particular, as organizações pequenas e médias não possuem recursos suficientes para adotar o uso de processos pesados. Por esta razão, elas não utilizam nenhum processo. O resultado desta falta de sistematização na produção de software é a baixa qualidade do produto final, além de dificultar a entrega do software nos prazos e custos predefinidos e inviabilizar a futura evolução do software [Soares 2010].

Humphrey (1989), em seu livro sobre melhoria de processos, argumenta que o conjunto de medidas de processo é essencial para o aperfeiçoamento do processo [Sommerville 2003]. Elas podem ser utilizadas para avaliar se a eficiência de um processo foi atingida ou não. No entanto, a dificuldade fundamental nas medições de processo é saber o que medir. Basili e Rombach (1988) propuseram o que chamam de paradigma de GQM (Goal-Question-Metric / meta-questão-métrica) [Sommerville 2003]. Essa abordagem auxilia na decisão de quais medições devem ser feitas e como elas devem ser utilizadas na avaliação do processo.

## Contexto

A G8 Consultoria Desenvolvimento em Informática atua no mercado do Recife desde 2003, e um dos serviços oferecidos é o desenvolvimento de software personalizado, com funcionalidades e customizações intrínsecas ao negócio do cliente. Atualmente na empresa, os projetos desenvolvidos, sob medida, não utilizam qualquer metodologia de desenvolvimento em seu processo, ficando a cargo do sócio responsável a forma de conduzir os projetos.

Com o objetivo de melhorar o processo de desenvolvimento de software, a G8 decidiu adotar em um de seus projetos, a metodologia de desenvolvimento ágil SCRUM. Mas terá essa adoção um impacto positivo como o esperado?

## Objetivos

Este trabalho tem como objetivo principal analisar o impacto da adoção da metodologia ágil SCRUM, em um projeto sob medida desenvolvido pela G8 consultoria. Para isso será utilizada a abordagem GQM para definir quais medidas serão utilizadas e como devem ser analisadas para identificar características de melhoria ou não no processo de desenvolvimento de software.

## Organização do documento

O capítulo 2 aborda as características da metodologia ágil Scrum, descrevendo os papéis de responsabilidade, etapas e artefatos produzidos durante o processo.

O capítulo 3 trata do uso de medições de software para melhoria de processo de software,.

O capítulo 4 descreve a abordagem GQM para medições de Software, detalhando suas etapas, artefatos produzidos, e ferramentas de apoio.

O capítulo 5 apresenta o estudo de caso aplicado, o qual teve por objetivo identificar o impacto da adoção da metodologia ágil SCRUM, utilizando a abordagem GQM.

Por fim considerações são feitas no capítulo 6 e propostas para trabalhos futuros são expostas.

# Scrum

Neste capítulo, as etapas, papéis de responsabilidade, ferramentas utilizadas e os artefatos produzidos são descritos com o objetivo de proporcionar um conhecimento básico sobre a metodologia, a qual será adotada no estudo de caso apresentado no capítulo 5 deste trabalho.

## Visão Geral

O Scrum [[Schwaber, Beedle 2002] tem o progresso de desenvolvimento baseado em interações com duração fixa. Tais interações são chamadas de *Sprint*. No início de cada *Sprint* uma reunião de planejamento chamada de *Sprint Planning* é realizada, onde o time de desenvolvimento, em conjunto com o cliente define o que será implementado na iteração.

Na etapa seguinte, o time detalha as tarefas necessárias para implementar cada solicitação do cliente, e posteriormente inicia a execução das mesmas. Durante o *Sprint* o time realiza rápidas reuniões diárias para apurar o progresso de desenvolvimento.

Ao final do *Sprint* é realizada uma reunião para a validação da entrega chamada de *Sprint Review*, na qual o cliente e o time podem verificar se o objetivo do *Sprint* foi atingido. Por fim, uma reunião é realizada, mas agora apenas pelo time de desenvolvimento. Nessa ultima reunião, que chamamos de *Sprint Retrospective*, o *Sprint* é avaliado sob o ponto de vista de processo, time ou produto, e os acertos e os erros são identificados.

A figura 1 resume o ciclo de desenvolvimento utilizando Scrum.

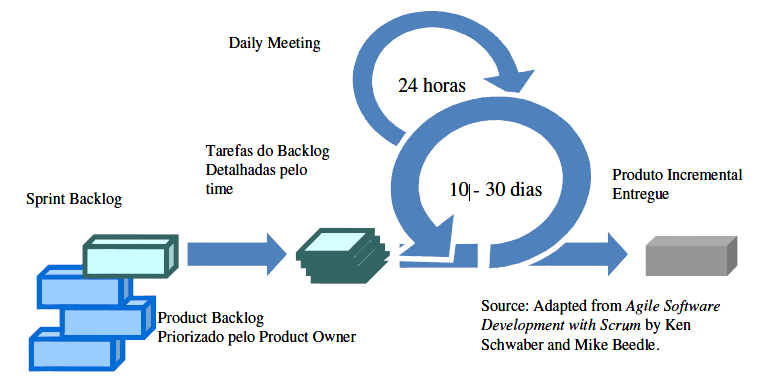


Figura 1 - Ciclo de Desenvolvimento do Scrum [Varaschi 2009].

Nas sessões seguintes uma descrição mais detalhada do processo do Scrum será realizada.

## Papéis e Responsabilidades

### Scrum Master

O Scrum Master é o responsável pela remoção de impedimentos que venham afetar à capacidade da equipe de entregar o objetivo do *Sprint*. Ele deve garantir que o processo Scrum seja usado como pretendido, protegendo a equipe e mantendo focada nas tarefas do *Sprint*.

### Scrum Team

O time ou equipe Scrum é responsável pelo desenvolvimento do produto. É tipicamente composta por um número pequeno de pessoas que possuem habilidades multifuncionais. Fazem o trabalho de analisar, projetar, desenvolver, testar, documentar, entre outros. É recomendado que a equipe seja auto-organizada e auto-conduzida, mas que muitas vezes trabalhem com alguma forma de projeto ou gestão de equipe [Varaschi 2009].

### Product Owner

O Product Owner representa a voz do cliente e é responsável por garantir que a equipe agregue valor ao negócio, é de responsabilidade do Product Owner [Varaschi 2009]:

* Definir todas as funcionalidades que o produto deve possuir, também é importante salientar que sugestões de histórias podem ser feitas por qualquer pessoa, mas a inclusão destas no Product Backlog é de gerência do Product Owner;
* Priorizar as funcionalidades que devem ser incluídas no Product Backlog a cada *Sprint*.
* Aceitar ou rejeitar as entregas do time ao fim do *Sprint*.

## Etapas do Sprint

O *Sprint* é a etapa básica de desenvolvimento em Scrum. Em um *Sprint* o time de desenvolvimento terá um período de tempo definido para implementar um conjunto de funcionalidades propostas pelo Product Owner, e com as quais se comprometeram a desenvolver.

### Sprint Planning

A primeira atividade dentro de um *Sprint* é uma reunião denominada Sprint Planning. No Sprint Planning é realizada a seleção das histórias que serão implementadas durante aquele *Sprint*.

O *Sprint* Planning é dividido em duas partes: na primeira delas, o Product Owner junto com o Scrum Master e a equipe de desenvolvimento, selecionam as histórias que serão implementadas no *Sprint* que se segue, de forma que o Product Owner deve priorizar os itens existentes no Product Backlog em ordem de importância, para que o time possa selecionar aqueles que considera possíveis de desenvolvimento dentro do Sprint. Ao final desta etapa, as histórias que vão ser implementadas no *Sprint* que se segue, devem ter sido selecionadas e se encaixar na velocidade do time.

Na segunda parte do Sprint Planning, o time deve desmembrar as histórias em tarefas que serão cumpridas durante o *Sprint*. Para cada história, o time detalha as atividades e estima o tempo necessário para o desenvolvimento. Ao finalizar as estimativas, se o tempo do *Sprint* não for suficiente para a realização de todas as tarefas detalhadas, é necessário dialogar com o Product Owner para rever as histórias que compõem o *Sprint*. Caso o tempo estimado seja inferior ao tempo do *Sprint*, deve-se prever a possibilidade de adicionar mais histórias do Product Backlog.

### Daily Scrum

O Daily Scrum é uma reunião rápida realizada diariamente e em pé, durante a qual cada membro do time explica o que fez desde a última reunião, o que fará até a próxima e quais são os impedimentos que surgiram. Esta reunião tem por objetivo acompanhar o desenvolvimento do projeto e distribuir as tarefas.

### Sprint Review

O Sprint Review é uma reunião que deve acontecer ao final de cada *Sprint*. Nela o time mostra o trabalho realizado para o Product Owner e possíveis convidados. Os itens que foram definidos como requisitos durante o Sprint Planning através de histórias de aceitação serão avaliados levando em consideração a completude do trabalho realizado.

Nesta etapa, o time irá demonstrar cada história desenvolvida, e cada uma delas será avaliada. O Product Owner realizará os testes de cada uma das histórias para verificar se as condições estabelecidas foram satisfeitas. Caso exista alguma história não finalizada, a equipe deve explicar os motivos ao Product Owner, para que o mesmo tome conhecimento de possíveis problemas, erros ou mudanças que ocorreram.

### Sprint Retrospective

O Sprint Retrospective é uma reunião que tem como principal foco o time. Realizada ao final de cada *Sprint*, nela são discutidos os problemas enfrentados pelo time no processo de desenvolvimento e sugestões de melhorias. Participam desta reunião o time, o Scrum Master e possíveis convidados por parte do time.

## Artefatos

### Product Backlog

O Product Backlog é a lista que contém as funcionalidades de negócio, os requisitos técnicos e os erros encontrados no sistema que precisam ser desenvolvidos. Esta lista está em constante evolução, seja pela adição de novos requisitos de negócio que surgiram com o tempo, ou pela necessidade técnica de manutenção no software.

O Product Backlog deve ser mantido em ordem de prioridade pelo Product Owner, que tem a liberdade de alterar a qualquer momento as histórias listadas. Um dos maiores desafios do Scrum é manter um Project Backlog organizado, seja com relação às prioridades ou com relação à complexidade referente ao desenvolvimento das historias [Varaschi 2009]. É de responsabilidade do Product Owner informar um valor de negócio para cada historia existente. O time é responsável por atribuir uma complexidade a cada história, e deve ser sempre solicitado a reavaliar as estimativas no Product Backlog.

A questão da priorização é resolvida pelas necessidades do negócio e a complexidade geralmente utiliza uma ferramenta chamada Planning Poker [Cohn 2005].

#### Planning Poker

O Planning Poker é uma técnica que o time de desenvolvimento utiliza para estimar a complexidade das histórias contidas no Project Backlog. A escala de complexidade é baseada na sequência de Fibonacci (1, 2, 3, 5, 8, 13, 21...) como base para a avaliação de complexidade.

É importante que a equipe sempre mantenha o Product Backlog com estimativas coerentes com a sua velocidade de desenvolvimento. Para tanto, cartas contendo os valores da sequencia de Fibonacci são entregues para cada integrante. Cada história presente no Product Backlog é estimada com uma carta mostrada por cada um dos integrantes do time ao mesmo tempo. Após a apresentação das cartas conferem-se os números das mesmas para ver se todos tem o mesmo entendimento sobre a complexidade da historia estimada. Em caso de divergência, cada membro pode argumentar o que o levou a pensar diferente dos demais e nesse momento pode usar os argumentos que justifique aquele item ser mais complexo ou mais simples. Após isto é realizada nova rodada e assim segue até se chegar a um consenso. Caso o time não consiga chegar ao mesmo entendimento sobre a complexidade da historia, é responsabilidade do Scrum Master intervir.

### Sprint Backlog

O Sprint Backlog é um conjunto de tarefas que foram derivadas das histórias propostas pelo o Product Owner e aceitas pela equipe. O desmembramento das histórias em tarefas menores é responsabilidade da equipe, e realizado na segunda parte do Sprint Planning.

### Impediment Backlog

O Impediment Backlog é uma lista de impedimentos que podem gerar problemas na entrega do *Sprint*, ou até do Projeto como um todo. Os impedimentos são tarefas que não puderam ser realizadas por alguma pendência externa. Essa lista deve ser tratada pelo Scrum Master, que deve agir como um facilitador na resolução destes impedimentos.

### Burndown Chart

O Sprint Burndown Chart é uma representação gráfica do andamento do trabalho no Sprint (ver Figura 2). Estes gráficos mostram ao longo do tempo a quantidade de trabalho realizado e a quantidade que ainda resta ser feito, sendo um excelente mecanismo para visualizar a relação entre a quantidade de trabalho realizada e em progresso.

O eixo vertical apresenta as horas de esforço restante no *Sprint*. O eixo horizontal representa os dias do *Sprint*. O Burndown deve ser representado por uma linha saindo do início do *Sprint* com as horas iniciais, descendo até o final do *Sprint*, sem deixar horas sobrando.

Figura 2 - Burndown Chart

# Mensuração em Projetos de Software

Com a aplicação de um programa de mensuração em uma organização que produz software, alguns objetivos podem ser alcançados. São eles [Nasa 94]:

* Compreensão dos produtos e processos de software
* Gerência de projetos de software
* Aperfeiçoamento contínuo

## Mensuração para Compreender Produtos e Processos de Software

Para que haja um controle e aperfeiçoamento do processo e produto de software, a compreensão dos mesmos é fundamental. Com base nas informações conseguidas através de um programa de medição, a organização pode estabelecer seus modelos de qualidade específicos e analisar fatores de contexto que têm impacto nesses modelos.

## Mensuração para Gerenciar Projetos de Software

Após a definição dos modelos de qualidade e os fatores que influenciam estes, a organização terá uma base melhor para a tomada de decisões. Esse conhecimento adquirido pode ser usado para [Nasa 94]:

* Planejamento de novos projetos de software: através dos modelos formais definidos através da medição, conhecimento sobre estimativa de esforço requerido, cronograma, alocação da equipe, entre outros, podem ser reutilizados para um planejamento mais seguro.
* Controle do projeto: através de comparações dos valores estimados com os valores coletados pelo plano de medição, ajustes e ações corretivas podem ser iniciadas, quando desvios do plano de projeto ou problemas são identificados.
* Validação dos modelos organizacionais: Por causa das mudanças permanentes na área de software, os modelos precisam ser ajustados ou outros modelos devem surgir para se adaptar às novas tendências.

## Mensuração para Guiar o Melhoramento

Para se destacar ou até mesmo sobreviver no mercado, uma empresa deve ter por princípio básico desenvolver um produto com alta qualidade e dentro do cronograma e orçamento acordado com o cliente. Para tanto, melhorias contínuas no processo e no produto devem fazer parte do objetivo da empresa.

O aperfeiçoamento do produto pode ser alcançado através da melhoria do processo usado para produzir o produto. O aperfeiçoamento do processo pode ser atingido pela modificação dos processos gerenciais ou técnicos, que fazem parte do processo como um todo. A partir do plano de medição, áreas que podem ser melhoradas e que têm um alto impacto na qualidade do produto são identificadas e planos de mudanças podem ser executados e avaliados. O impacto dessas mudanças precisa ser acompanhado, para que seja possível determinar se uma mudança foi positiva ou não, uma linha-base de comparação tem que estar definida e disponível [Differding, Hoisl 95].

# Abordagem Goal Question Metric

Proposto por Basili e Rombach em 1988, o paradigma Goal Question Metric (GQM) [Basili, Rombach 1988] é uma abordagem orientada a metas para a medição de produtos e processos de software. É executada em duas partes: a primeira contempla a definição *top-down* de um programa de mensuração e a segunda parte a análise e interpretação *bottom-up* dos dados coletados [Gresse 98] conforme apresentado na Figura 3. Essa abordagem foi utilizada com sucesso em diversas empresas, como NASA-SEL (EUA), Allianz Lebensversicherungs-AG (Alemanha) [Günther, Rombach, Ruhe 94], Motorola [Daskalantonakis 92].

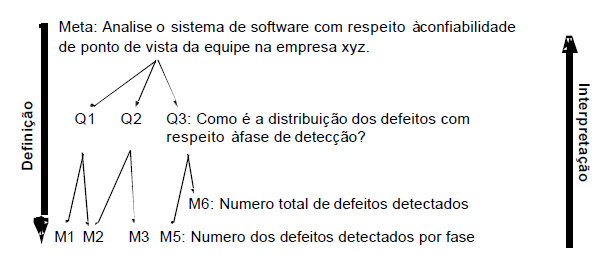


Figura 3 - Abordagem GQM [Gresse 98]

O paradigma GQM se baseia no princípio de que a medição deve ser orientada a metas, de forma que a coleta dos dados deve ser baseada nos fatores de qualidade que influenciam direta ou indiretamente as metas definidas.

Segundo Gesse von Wangenheim [Gresse 98] programas de mensuração baseados em GQM devem ser planejados e executados de acordo com os seguintes princípios:

* A tarefa de análise a ser executada precisa ser especificada precisamente e explicitamente através de uma meta.
* As medidas precisam ser derivadas de forma *top-down* baseadas nas metas definidas e perguntas. Uma estrutura de metas e perguntas não pode ser adaptada de forma retroativa a um conjunto de medidas existente.
* Cada medida precisa ter um fundamento lógico subjacente que é documentado explicitamente. O fundamento lógico é usado para justificar a coleta dos dados e para guiar a análise e interpretação.
* Os dados que são coletados com respeito às medidas definidas precisam ser interpretados de uma forma *bottom*-up no contexto das metas e das perguntas.
* As pessoas que vão utilizar os resultados do programa de mensuração precisam ser envolvidas profundamente na definição e interpretação do programa de mensuração. Elas são os reais peritos com respeito ao objeto e ao enfoque de qualidade investigado no programa de mensuração, portanto proveem interpretações válidas no ambiente especifico.

## Etapas do GQM

Com base nas experiências de aplicações do paradigma GQM em várias empresas, o processo GQM foi modelado em detalhe [Gresse, Hoisl, Wüst 95]. A seguir, uma visão geral da abordagem GQM é dada, a qual será descrita em etapas definidas com base no modelo de processo GQM abordado em [Gresse, Hoisl, Wüst 95]. As fases do processo GQM são orientadas ao Paradigma de Melhoria de Qualidade (QIP).

No início do plano de mensuração, um estudo prévio é realizado com o objetivo de identificar as características da organização, as metas e os projetos que ela desenvolve. Um projeto piloto deve ser selecionado e caracterizado para aplicar o programa de mensuração. Na etapa seguinte, uma meta a ser atingida pelo programa de mensuração deve ser especificada, bem como o objeto de estudo, o objetivo, o enfoque de qualidade, o ponto de vista e o contexto da análise.

Um conjunto das medidas relevantes é derivado via perguntas e modelos, resultando em um plano GQM consistindo de meta, perguntas, modelos e medidas. Um plano de mensuração é então formulado pela integração das medidas definidas no plano GQM e do plano de projeto de software, pela qual deve se determinar quando, como, e por quem as medidas requeridas devem ser coletadas.

Durante a etapa de execução do programa de mensuração, os dados são coletados de acordo com o plano de mensuração. Esses dados devem ser analisados e interpretados com respeito às metas GQM de acordo com o plano GQM em seções de *feedback*. Os resultados de análise e interpretação são acondicionados em modelos reutilizáveis [Gresse 98].

Uma visão geral das etapas do processo de um programa de mensuração é dada na figura 4.

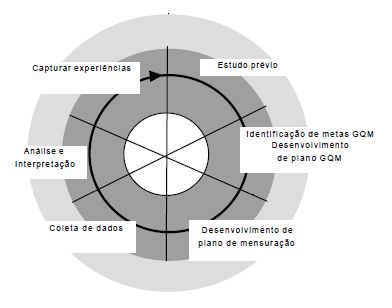


Figura 4 - Visão geral do processo GQM [Gresse 98]

Nas próximas seções segue-se uma descrição detalhada de cada passo de um programa de mensuração baseado em GQM.

### Estudo Prévio

O objetivo do estudo prévio é reunir informações pertinentes à introdução de um programa de mensuração baseado em GQM na organização. Essas informações são usadas para apoiar a escolha de um projeto piloto para a introdução de um programa de mensuração e definir as metas de mensuração.

Nesta etapa, precondições e restrições devem ser identificadas. A organização e os projetos potenciais para aplicação do programa devem ser caracterizados, e as metas de melhoria organizacionais identificadas. Com base nessas informações, um projeto piloto é selecionado para a aplicação do programa de mensuração.

### Identificação de Metas GQM

Com base nas metas de melhorias organizacionais e do projeto piloto selecionado, as metas que devem ser atingidas pelo programa de mensuração são determinadas, o que servirá de base para o desenvolvimento de um programa de mensuração efetivo. As metas GQM podem ser derivadas das metas de negócio, das metas estratégicas da organização, ou mais diretamente das metas organizacionais de melhoramento. Um questionário, que implicitamente consistem numa caracterização de projeto e organização e identificação de meta de melhoramento, pode estimular a discussão para a definição de meta [Solingen 95]:

* **Objeto de Estudo:** o que será analisado?
* **Objetivo:** porque o objeto será analisado?
* **Enfoque de qualidade:** qual atributo do objeto será analisado?
* **Ponto de Vista:** quem usará os dados coletados?
* **Contexto:** em qual contexto será analisado?

### Desenvolvimento do Plano GQM

De acordo com as metas GQM, um plano de mensuração é formulado para cada meta GQM selecionada. Um plano GQM é constituído por uma meta GQM e um conjunto de perguntas, modelos e medidas. Nele, encontram-se o motivo da escolha de cada medida e como elas serão utilizadas. As perguntas serão utilizadas para identificar a informação necessária para alcançar a meta, e as medidas para definir operacionalmente os dados que serão coletados para responder as perguntas.

Para determinar quais perguntas são relevantes com respeito à meta, entrevistas são feitas. O objetivo das entrevistas é adquirir o conhecimento implícito das pessoas envolvidas. Essas entrevistas são feitas com as pessoas declaradas no ponto de vista da meta GQM [Gresse 98].

#### Entrevistas GQM

As entrevistas devem ser realizadas com as pessoas que fazem parte da dimensão “ponto de vista” da meta GQM, para assegurar que as informações coletadas são realmente pertinentes à perspectiva da meta. Alguns tópicos devem ser enfocados na entrevista, são eles [Gresse, Hoisl, Wüst 95][Günther, Rombach, Ruhe 94]:

* **Fatores de Qualidade:** para o entrevistado, o que significa o enfoque de qualidade especificado na meta GQM? Por exemplo, se o enfoque de qualidade definido foi Qualidade do Produto, então o que significa qualidade do produto para o entrevistado.
* **Hipótese de linha-base:** Para cada fator de qualidade que pertence ao enfoque de qualidade é declarado: qual é a estimativa do entrevistado no estado atual ao fator de qualidade? Por exemplo, qual a estimativa da quantidade de defeitos descobertos em produção atualmente? Neste caso, temos como fator de qualidade os defeitos descobertos em produção, que está relacionado à qualidade do produto.
* **Fatores de Variação:** Quais fatores ambientais variáveis têm possivelmente um impacto nos fatores de qualidade? Seguindo a linha dos exemplos anteriores, a experiência dos desenvolvedores pode ter impacto na quantidade de defeitos gerados. Outro fator de variação pode ser a qualidade dos testes realizados antes da entrega do produto.
* **Impacto na Hipótese de Linha-Base:** Para cada fator de variação é declarado: para o entrevistado, qual é a estimativa do impacto do fator de variação no fator de qualidade? Nesse item, é descrito o impacto que os fatores de variação têm que sobre as hipóteses de linha base.

Para auxiliar a aquisição e estruturação de conhecimento durante as entrevistas, uma ferramenta chamada Abstraction Sheet pode ser utilizada [Günther, Rombach, Ruhe 94]. O Abstraction Sheet é um documento de uma página com quatro quadrantes, um para cada tópico que deve ser enfocado, representando uma visão simplificada do plano GQM. As informações reunidas durante uma ou mais entrevistas serão documentadas e resumidas nesta planilha. Um exemplo de Abstraction Sheet pode ser observado na Tabela 1.

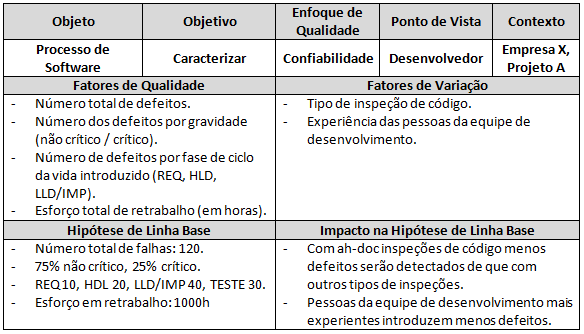


Tabela 1 - Exemplo de um Abstraction Sheet [Gresse 98]

#### Refinamento de Plano GQM

O objetivo dessa etapa é derivar um conjunto de medidas a partir de perguntas e modelos, com base nos fatores de qualidade e fatores de variação obtidos nas entrevistas.

Neste passo o plano GQM é desenvolvido. As perguntas são derivadas com base na informação documentada no Abstraction Sheet. Para cada fator de qualidade no Abstraction Sheet, uma ou mais perguntas são formuladas expressando a necessidade de informação relacionada ao fator de qualidade. A informação contida na seção “fatores de variação” do Abstraction Sheet é usada para derivar perguntas com respeito ao contexto pertinente [Gresse 98].

Em seguida, modelos preditivos são construídos. Os atributos a serem medidos para responder as perguntas são operacionalizados através de modelos preditivos. A definição das medidas pode ser composta pelo nível de mensuração (exemplo: nominal, ordinal, intervalo, racional), a faixa de valores aceitáveis (valores esperados), o contexto de aplicação e a descrição de como a medida é calculada [Gresse 98].

### Desenvolvimento do Plano de Coleta

Nessa etapa, é determinado quando, como e por quem as medidas identificadas no plano GQM serão coletadas. Dependo do objetivo de mensuração e dos dados coletados, dois tipos principais de estratégias podem ser adotados para a coleta dos dados [Briand, Differding, Rombach 97]:

* Periodicamente (exemplo: coleta de dados de esforço por dia ou semana).
* Início, fases ou fim de atividades (exemplo: defeitos graves encontrados ao fim da *Sprint* do scrum).

Depois que um cronograma para a coleta for definido, é determinado quem irá coletar os dados. Para algumas medidas, existe a possibilidade de utilizar uma ferramenta para automatizar a coleta de dados.

Definir corretamente os instrumentos que serão utilizados para coletar os dados é crucial para receber dados confiáveis. A decisão sobre qual instrumento é utilizado depende dos dados que serão coletados. Três principais categorias de instrumentos de coleta de dados podem ser identificadas [Gresse 98]:

* Ferramentas: podem ser utilizadas para coletar medidas objetivas de artefatos. Por exemplo, analisadores de código estáticos que calculam o número total de linhas de código.
* Questionários: podem ser utilizados para coletar medidas se processo. Por exemplo, esforço por sprint.
* Entrevistas estruturadas: assim como os questionários, as entrevistas também podem ser utilizadas para coletar medidas de processo. Por exemplo, a usabilidade de uma interface.

A Tabela 2 apresenta um plano de mensuração montado. Analisando a primeira linha, temos que o esforço do projeto será medido, considerando a unidade homens/mês, que será coletada periodicamente a cada semana pelo próprio desenvolvedor. Para tanto, será utilizado um questionário de esforço como instrumento de coleta, e o gerente terá a responsabilidade de conferir a completude e corretude dos dados registrados.

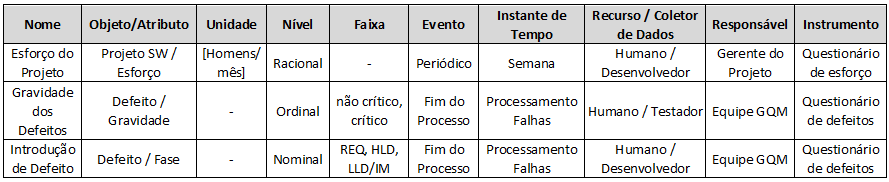


Tabela 2 - Exemplo de Plano de Mensuração [Gresse 98]

### Coleta de Dados

Durante essa etapa, os dados são coletados de acordo com o plano de mensuração. Após a coleta das medidas, um processo de garantia de qualidade deve ser realizado para garantir os seguintes pontos [Gresse 98]:

* Completude: garantir que todos os questionários foram submetidos e estão completos.
* Plausibilidade: garantir que os dados coletados são do tipo especificado e se o valor informado para uma determinada medida é possível.

Se durante processo de garantia de qualidade algum dado faltante ou falso for descoberto, este deve ser discutido com o coletor de dados responsável, e se possível efetuar a correção. O tempo entre a coleta e garantia de qualidade de dados não deve ser muito longo, para permitir uma possível correção [Gresse 98].

Os dados válidos são armazenados no banco de dados de mensuração disponíveis para a etapa de análise.

### Análise dos Dados

Nessa etapa, os dados obtidos são comparados com as hipóteses de linha base, permitindo avaliar e quantificar a relação entre os fatores de variação e os fatores de qualidade.

#### Comparação com linhas bases

Os dados coletados podem ser usados para construir linhas-base quantitativas para os projetos de software da organização. Em geral, é interessante comparar as linhas-base medidas atualmente com as hipóteses esperadas que estejam definidas nos Abstraction Sheets. Dessa forma, divergências são discutidas entre os desenvolvedores e gerentes para a interpretação dos dados [Gresse 98].

#### Validação de hipóteses sobre o impacto de fatores de variação

Dependendo do propósito da meta GQM, estratégias diferentes são aplicadas [Briand, Differding, Rombach 97]. Para o objetivo de predição, as hipóteses sobre o impacto de fatores de variação são testadas respondendo à seguinte pergunta: o fator de variação teve o impacto esperado no fator de qualidade? Se o impacto esperado não foi verificado, então o fator de variação é descartado. Se o impacto esperado for verificado, a relação identificada pode ser usada para construir modelos novos ou mais confiáveis para a administração de projeto e garantia de qualidade [Gresse 98].

Os resultados da análise de dados são uma entrada essencial para a interpretação de dados.

### Interpretação dos Dados

A interpretação dos dados coletados é feita em reuniões chamadas de **Feedback Sessions**. Nessas reuniões devem estar presentes os representantes do ponto de vista especificado na meta e os coletores de dados. O objetivo do **Feedback Sessions** é interpretar os dados coletados com pessoal diretamente relacionados com os dados coletados e que tem o conhecimento necessário. Assim os participantes poderão sugerir interpretações e modificações para aperfeiçoamento, com base nos dados analisados que forem apresentados [Gresse 98], [Briand, Differding, Rombach 97],[ Hoisl, Oivo, Ruhe 96],[ Gresse, Hoisl, Wüst 95].

#### Apresentação dos Dados Analisados

O plano GQM apoia a análise e a interpretação dos dados no sentido bottom-up. Os dados coletados são combinados às medidas no plano GQM e processados como descrito no modelo. Esses resultados são colocados junto com as perguntas correspondentes e comparados às hipóteses iniciais [Gresse 98].

#### Feeback Sessions

**Feeback Sessions** são reuniões de grupo, cujos participantes são os representantes dos pontos de vista especificados nas metas GQM e os coletores de dados. O objetivo principal do **Feedback Sessions** é a interpretação de dados de mensuração, a verificação das hipóteses declaradas nos planos GQM, e a identificação das possibilidades para melhoramento do processo de software [Gresse 98].

As pessoas que representam os pontos de vista das metas sabem usar os dados para os propósitos delas, e só elas podem tirar conclusões. Os coletores de dados sabem quão bem os dados providos foram coletados, e podem alertar uma falha que ocorreu durante a coleta de dados [Gresse 98].

A interpretação dos dados pode conduzir à identificação de pontos de fraquezas no processo, o que levará à discussão de possíveis estratégias de aperfeiçoamento. A iniciação de mudanças para alcançar benfeitoria do processo de software tem que ser preparada especificando “pontos de ação” para cada mudança [Gresse 98]:

* Qual modificação foi sugerida?
* Quem é responsável pela implantação de modificações?
* Quando a modificação deve ser implantada?

O feedback session também pode ser utilizado para avaliar o próprio programa de mensuração [Gresse 98]:

* Se alguns dados de mensuração apresentam pouco ou nenhum uso ou se referem a uma pergunta que já foi respondida, esses dados (e as perguntas correspondentes) são excluídos do plano GQM.
* Se os procedimentos de coleta são muito intrusivos, muito demorados, eles devem ser simplificados.
* O foco de interesse da organização pode mudar durante o programa de mensuração.

### Captura de Experiências

O objetivo dessa etapa é definir claramente as experiências ganhas durante o programa de mensuração para reutilizar esse conhecimento em projetos de software futuros [Gresse 98]. Os dados coletados, analisados e interpretados no programa de mensuração podem ser usados para construir modelos de processos e produtos básicos desenvolvidos pela organização.

Algumas formas de se capturar experiência de acordo com [Nasa 94]:

* Políticas e guias de gerência de software, incluindo vários modelos, por exemplo, modelos preditivos de esforço.
* Padrões de desenvolvimento e manutenção de software, definindo produtos, métodos, técnicas e ferramentas utilizadas durante a aplicação GQM, que se mostraram benéficas para a organização.
* Relatórios de estudos de processo enfocando as questões específicas, os métodos aplicados, os resultados medidos e as conclusões obtidas.

# Estudo de caso

## A Empresa

O estudo de caso foi realizado em uma microempresa de software, atuante no mercado da região metropolitana do Recife – PE, chamada G8 Consultoria Desenvolvimento em Informática. A G8 é especializada em desenvolver soluções específicas, a partir das necessidades de cada cliente. Composta basicamente pelos seus sócios, cada um é responsável por desenvolver seus projetos. Quase sempre cada projeto é desenvolvido por um único analista desenvolvedor, responsável por todo o ciclo de desenvolvimento do software, desde o levantamento de requisitos até a sua manutenção. Não existe processo de desenvolvimento de software padrão definido na empresa. A preocupação está voltada ao produto final: o software.

Por ter pouco recurso humano e o curto prazo de entrega do projeto, micro e pequenas empresas muitas vezes não utilizam nenhum processo de desenvolvimento, o que pode levar a um descontrole do projeto, impactando negativamente na qualidade do produto e na satisfação do cliente. As metodologias de desenvolvimento ágil propõem uma abordagem mais focada nos indivíduos e interações, com o software estar executável, com a colaboração do cliente e as respostas rápidas a mudanças e alterações. Esses conceitos aproximam-se melhor com a forma que pequenas e médias organizações trabalham e respondem a mudanças. Entre as metodologias ágeis a mais conhecida é o Scrum [Schwaber, Beedle 2002].

## O Projeto Selecionado

O projeto escolhido trata-se de um gerenciador de contratos de garantia veicular. Esse projeto foi escolhido por ser um tipo de projeto típico da empresa, com necessidade de benfeitoria e não possuir muitos riscos nem impedimentos.

O projeto foi iniciado em agosto de 2011, solicitado pela empresa TST Garantia Veicular em Recife - PE. Inicialmente, o sistema foi desenvolvido para o controle dos pagamentos das mensalidades dos clientes. Aos poucos, o sistema foi crescendo e hoje, consegue gerenciar além dos contratos, clientes, pagamento, fornecedores, e os sinistros ocorridos. O projeto inteiro está sendo desenvolvido por apenas um analista desenvolvedor, de forma que a cada semana, um número de horas é reservado para aquele projeto, e ao fim da semana ou conclusão de uma nova funcionalidade, uma nova versão do sistema é entregue e posta em produção.

## Adoção do SCRUM

O Scrum tem o processo de desenvolvimento baseado em iterações com duração fixa, chamadas de Sprints. Este ciclo se adapta bem a forma com que a G8 desenvolve o projeto selecionado. Alguns pontos foram moldados para que o Scrum pudesse ser implementado.

### Acúmulo de Responsabilidades

Como o projeto é desenvolvido por apenas um analista, os papéis de Scrum Master e o Scrum Team são atribuídos à mesma pessoa, no caso, o próprio analista.

### Duração do Sprint

O sócio responsável destina ao projeto vinte horas por semana, distribuídas em três dias na semana, no próprio escritório do cliente. Como cada nova funcionalidade desenvolvida deve ser colocada em produção rapidamente, o tempo de um Sprint não pode ser muito longo. Por essa razão, foi definido que a duração do Sprint seria o total de horas disponíveis para execução do projeto em uma semana, no caso, vinte horas.

### Sprint Review

No Sprint Review, além do desenvolvedor, que nesse caso representa o time de desenvolvimento, e do Product Owner, representado pelo cliente, são convidados a participar da reunião todos os usuários do sistema que usarão diretamente as funcionalidades implementadas, no Sprint que se encerra. Usuários que serão afetados pela inclusão ou modificação de algum requisito também podem ser convidados a participar. A presença dos convidados tem o objetivo de treinar os usuários para o uso das historias implementadas.

### Sprint Retrospective

Nessa etapa, além de procurar melhorias no processo, o desenvolvedor deverá também coletar algumas métricas de controle utilizadas na abordagem GQM.

### Ferramenta de Apoio

Para facilitar o controle do projeto, uma ferramenta baseada no framework Scrum foi adotada, o Scrum Half, uma ferramenta web para apoio a projetos ágeis. O Scrum Half foi utilizada para:

* Criação e gerenciamento do Product Backlog por parte do Product Owner (cliente).
* Criação e gerenciamento do Sprint Backlog pelo desenvolvedor.
* Utilização do quadro de tarefas para controle visual das tarefas do Sprint.
* Apoiar o Sprint Review e o Sprint Retrospective.
* Obter relatórios sobre o andamento do Sprint e coleta de medidas.

## Aplicação da Abordagem GQM no Projeto

Para avaliar o impacto da adoção da metodologia de desenvolvimento Scrum, foi construído um plano de mensuração baseado na abordagem GQM. O plano tomou como base as etapas básicas do GQM, descritas no item 3 deste trabalho.

Primeiro foi realizado o estudo prévio da organização e do projeto selecionado. Em seguida, com base nas metas organizacionais, as metas GQM foram identificadas e caracterizadas. O estudo continuou com o desenvolvimento do plano GQM, realizado a partir de entrevistas padrões e posterior refinamento para se chegar as medidas que deveriam ser coletadas. Com isso, foi desenvolvido um plano para coletar os dados definidos, no qual foi descrito como, quando e por quem esses dados seriam coletados. Após a coleta, os dados foram analisados e interpretados.

Nos itens seguintes deste trabalho, são descritas cada etapa da Abordagem GQM realizada no estudo de caso.

### Estudo prévio

#### Caracterização da Organização

* A G8 consultoria é uma empresa de desenvolvimento de software, especializada em desenvolver soluções específicas, a partir das necessidades de cada cliente.
* Cada projeto é desenvolvido por um único analista desenvolvedor, responsável por todo o ciclo de desenvolvimento do software, desde o levantamento de requisitos até a sua manutenção.
* Não existe processo de desenvolvimento de software padrão definido. A preocupação está voltada ao produto final: o software.

#### Projeto Potencial para a aplicação da medição GQM

Para aplicar a mensuração GQM, o projeto selecionado atende os seguintes critérios:

* É um projeto típico para a organização.
* Projeto com necessidade de benfeitoria.
* Projeto com ciclos de curta duração.
* O projeto não possui muitos riscos.

Caracterização do projeto selecionado:

* Projeto: Gerenciador de Contratos de Garantia Veicular.
* Duração: iniciada e em andamento. Total de 10 meses até 01/06/2012
* Recurso Destinado: Um analista desenvolvedor
* Plataforma: Sistema Desktop
* Linguagem de programação: Delphi 2010
* Banco de dados: SQL Server 2008 R2

É muito importante que as pessoas envolvidas no projeto e que irão participar ou sofrerão impacto do programa de mensuração, sejam motivadas e concordem com a introdução de um programa de mensuração. Uma reunião com o cliente foi realizada, a fim de informar os benefícios do programa de mensuração baseado em GQM.

#### Modelo de processo de software utilizado no Projeto

A existência de um modelo de processo de software é essencial para planejar e executar um programa de mensuração [Bröckers, Differding, Threin 96]. Com o objetivo de melhorar o controle do processo de desenvolvimento, a G8 Consultoria adotou a metodologia de desenvolvimento SCRUM no projeto selecionado.

#### Meta Organizacional

As metas organizacionais devem ter por objetivo o melhoramento com respeito a problemas conhecidos. Visando aprimorar o processo de desenvolvimento de software, a metodologia Scrum foi adotada. Mas terá essa adoção um impacto positivo no processo de desenvolvimento do projeto? A partir desta pergunta, e do problema conhecido, chegou-se a meta organizacional.

Meta Organizacional: **“Avaliar se a adoção de SCRUM teve um impacto positivo ou não no andamento do projeto”.**

Para se chegar as metas de progressos GQM, é necessário refinar a meta organizacional. A pergunta abaixo foi dirigida ao analista desenvolvedor responsável pelo projeto, com o objetivo de refinar a meta organizacional.

**O que faria a adoção do Scrum ter um impacto que fosse positivo?**

1. Se o processo de desenvolvimento for melhor definido.
2. Se mais informação sobre o andamento estiver disponível.
3. Se o processo de desenvolvimento gerar produtos com melhor qualidade.
4. Se o processo de desenvolvimento produzir mais funcionalidades em um mesmo período de tempo.

Com base nas respostas obtidas, os itens de controle selecionados para avaliar o processo de desenvolvimento de software foram:

* **Qualidade do Produto Produzido**, de acordo com o item **c**.
* **Produtividade**,de acordo com o item **d**.
* **Controle**, de acordo com os itens **a** e **b.**

Dessa forma, a partir dos itens de controle listados chega-se às metas de melhoria que têm impacto na meta organizacional. São elas:

* **Avaliar a qualidade do produto produzido**
* **Avaliar a produtividade**
* **Avaliar o controle do processo**

### Identificação de Metas GQM

O modelo de meta GQM é escrito para cada meta de progresso seja identificada. Dessa forma é possível indicar a informação necessária a mensuração e apoiar a definição da meta. Os modelos são descritos abaixo:

**Objeto de Estudo:** O Processo de Desenvolvimento.

**Objetivo:** Avaliação.

**Enfoque de Qualidade:** Produtividade

**Ponto de Vista:** Gerente de Projeto e o desenvolvedor.

**Contexto: Organização:** G8 consultoria; Projeto: Gerenciador de Contratos de Garantia Veicular.

**Objeto de Estudo:** O Processo de Desenvolvimento.

**Objetivo:** Avaliação.

**Enfoque de Qualidade:** Qualidade do Produto Produzido

**Ponto de Vista:** Gerente de Projeto e o desenvolvedor.

**Contexto:** Organização: G8 consultoria Projeto: Gerenciador de Contratos de Garantia Veicular.

**Objeto de Estudo:** O Processo de Desenvolvimento.

**Objetivo:** Avaliação.

**Enfoque de Qualidade:** Controle

**Ponto de Vista:** Gerente de Projeto e o desenvolvedor.

**Contexto:** Organização: G8 consultoria Projeto: Gerenciador de Contratos de Garantia Veicular.

#### Justificativa da escolha da meta

* O Objeto de Estudo foi escolhido pela necessidade de compreender melhor o processo de desenvolvimento.
* Avaliação visa comparar e aferir o processo de software após a adoção da metodologia de desenvolvimento SCRUM.
* O enfoque de qualidade é consistente com a prioridade de metas de melhoramento organizacionais.

### Desenvolvimento do Plano GQM

Nesta etapa vamos definir as perguntas utilizadas para descobrir como alcançar as metas, e as medidas que respondem as perguntas. Tal objetivo foi alcançado com o uso de entrevistas.

#### Entrevista

Para determinar as perguntas relevantes com respeito à meta, entrevistas são feitas com as pessoas definidas no ponto de vista da meta. Foram destinadas ao Gerente as seguintes perguntas:

#### Perguntas para definir os Fatores de Qualidade de Cada Meta

Ponto de vista: Analista Desenvolvedor

* **P1.** O que significa qualidade?

**R1.** Produzir menos distorções e conseguir entregar o que foi prometido no tempo e no custo estimado.

* **P2.** O que significa produtividade?

**R2.** Produzir mais funcionalidades no mesmo espaço de tempo.

* **P3.** O que significa controle?

**R3.** Possuir dados de apoio.

A partir das repostas podemos identificar os fatores de qualidade. São eles:

* + Defeitos em produção – Qualidade.
  + Funcionalidades previstas, mas não concluídas - Qualidade.
  + Quantidade de funcionalidades entregue – Produtividade.
  + Completude dos dados de apoio – Controle.

#### Perguntas para definir as hipóteses de linha-base

Ponto de vista: Analista Desenvolvedor

* **P4.** Qual é a estimativa no estado atual para os defeitos descobertos em produção?

**R4.** Em média dois defeitos são descobertos em produção a cada entrega.

* **P5.** Qual é a estimativa no estado atual da quantidade de funcionalidades que NÃO são entregues no prazo?

**R5.** Em média uma funcionalidade não é totalmente concluída a cada ciclo.

* **P6.** Qual é a estimativa no estado atual da quantidade de funcionalidades entregues por ciclo?

**R6.** Cinco funcionalidades por ciclo.

* **P7.** Qual é a estimativa no estado atual da completude dos dados de apoio que são informados durante o ciclo?

**R7.** Apenas um documento é mantido, com a quantidade de horas trabalhadas no projeto, atualizado com 100% das informações.

* **P8.** Qual a estimativa da quantidade de informação de apoio que está disponível diante do que realmente é necessário?

**R8.** 70%

#### Perguntas para definir os fatores que podem afetar a qualidade

Ponto de vista: Analista Desenvolvedor

* **P9.** Quais são os fatores que podem influenciar na quantidade de defeitos descobertos em produção?

**R9.** Experiência do desenvolvedor com a linguagem de programação utilizada e realização de testes.

* **P10.** Quais são os fatores que podem influenciar no erro de estimativa?

**R10.** Complexidade de algumas funcionalidades estimada de forma inadequada.

* **P11.** Quais são os fatores que podem influenciar na quantidade de funcionalidades entregues?

**R11.** Experiência do desenvolvedor e impedimentos.

* **P12.** Quais são os fatores que podem influenciar na completude dos dados de apoio?

**R12.** Não registro da informação pelo responsável.

#### Perguntas para definir impacto do fator de variação no fator de qualidade

Ponto de vista: Analista Desenvolvedor

* **P13.** Qual é a estimativa do impacto que o não registro da informação tem no nos documentos de apoio?

**R13.** Se os registros necessários não forem informados, os documentos de apoio não conseguem expor um conhecimento completo.

* **P14.** Qual é a estimativa do impacto que a experiência do desenvolvedor tem na quantidade de funcionalidades entregues?

**R14.** Desenvolvedores mais experientes com a linguagem de programação produzem mais rapidamente.

* **P15.** Qual é a estimativa do impacto que a existência de impedimentos tem na quantidade de funcionalidades entregues?

**R15.** Quanto mais impedimentos, mais tempo para a solução dos impedimentos é gasto, o que implica em menos tempo para desenvolver as funcionalidades previstas para o ciclo.

* **P16.** Qual é a estimativa do impacto que o erro na estimativa de complexidade tem nos erros de estimativa de produção do ciclo?

**R16.** Ao se errar a estimativa de complexidade de uma funcionalidade, o tempo de produção destinado pode ser maior ou menor do que o necessário. No caso de se atribuir um tempo menor do que realmente o necessário, ocorre um erro mais grave, pois alguma funcionalidade ao fim do ciclo poderá não ser completamente desenvolvida.

* **P17.** Qual é a estimativa do impacto que a experiência do desenvolvedor tem nos defeitos descobertos em produção?

**R17.** Desenvolvedores mais experientes produzem menos erros.

* **P18.** Qual é a estimativa do impacto que realização de testes tem nos defeitos descobertos em produção?

**R18.** Alguns defeitos podem ser identificados e corrigidos antes da nova versão ser usada em produção.

As tabelas abaixo representam o Abstraction Sheet de cada meta GQM selecionada.

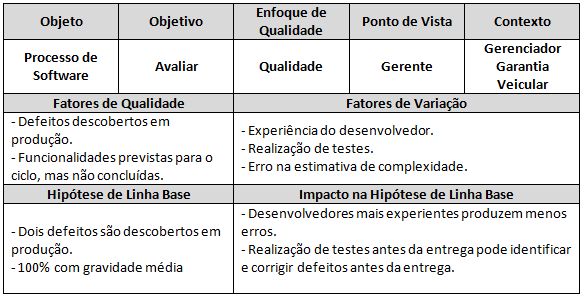


Tabela 3 - Abstraction Sheet Enfoque Qualidade

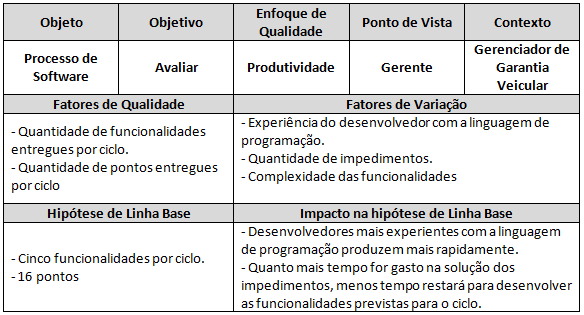


Tabela 4 - Abstraction Sheet Enfoque Produtividade

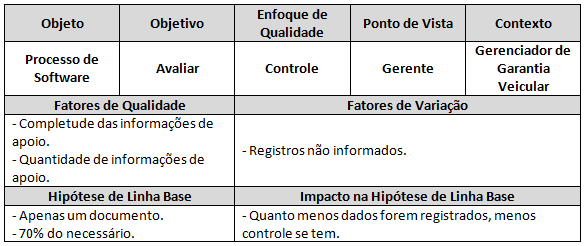


Tabela 5 - Abstraction Sheet Enfoque Controle

#### Entrevista de Refinamento

A partir das respostas obtidas na entrevista de refinamento, pode-se formular um modelo preditivo para cada enfoque de qualidade.

#### Perguntas para refinar o enfoque de qualidade – Controle

Ponto de vista: Analista Desenvolvedor

* **P1.** Quais são os documentos e sua representatividade que trariam 100% de controle?

**R1.**

* + **Controle de horas**

70%.

* + **Controle de funcionalidades previstas, em andamento e finalizadas**

30%.

* **P2.** Quais informações que compõem cada documento, e dependem do registro do responsável?

**R2.**

* + **Controle de horas**

Data hora de início e término da jornada de trabalho.

* + **Controle das funcionalidades**

Descrição da tarefa.

Momento de início de desenvolvimento da tarefa.

Momento de término do desenvolvimento tarefa.

Baseado nas respostas das perguntas de refinamento acima informadas, o modelo formado para predizer o enfoque de qualidade Controle é descrito da seguinte forma:

**Modelo 1:** completude do documento de controle de horas. Calculado de seguinte forma:

Completude controle de horas = nº de registros informados / nº total de registros que deveriam ser informados

**Atributo 1: nº de registros informados.** Obtido através do documento de controle de horas no final do ciclo.

**Modelo 2**: completude dos registros referentes ao andamento das funcionalidades finalizadas. Calculado da seguinte forma:

Completude dos registros das tarefas em andamento = nº de tarefas com registro de andamento / nº total de tarefas finalizadas.

**Atributo 2: nº de total de tarefas alocadas para o ciclo.** Obtido através da ferramenta Scrum Half no final do ciclo.

**Atributo 3: nº de tarefas com registro de andamento.** Obtido através da ferramenta Scrum Half no final do ciclo.

#### Perguntas para refinar o enfoque de qualidade – Produtividade

Ponto de vista: Analista Desenvolvedor

* **P3.** Qual a distribuição de experiência entre as pessoas da equipe?

**R3.** Há apenas um desenvolvedor com três anos ou mais de experiência.

* **P4.** Quanto tempo é gasto em solucionar os impedimentos?

**R4.** Em média 30 minutos são gastos para solucionar impedimentos.

* **P5.** Quais são os impedimentos encontrados?

**R5.** Como o desenvolvimento é realizado no escritório do cliente, dependemos de internet e energia dele. Quedas na internet ou quedas na energia impossibilitam seguir com os trabalhos. Atualizações remotas são realizadas nas filiais, e depende da disponibilidade da filial para que se possa atualizar a versão.

* **P6.** Como estimar a complexidade da funcionalidade?

**R6.** Atividade Planning Poker do Scrum, para estimar a complexidade.

**Modelo 3**: A produtividade deve ser calculada como sendo a soma das complexidades referente a cada história concluída por ciclo.

**Atributo 4: Quantidade de funcionalidades concluídas por ciclo**. Obtido através da ferramenta Scrum Half no final do ciclo.

**Atributo 5: Somatório das Complexidades definidas para cada funcionalidade concluída no ciclo.** Obtido através da ferramenta Scrum Half no final do ciclo.

**Atributo 6: Tempo utilizado na solução de impedimentos encontrados durante o ciclo.** Obtido através da ferramenta Scrum Half.

#### Perguntas para refinar o enfoque de qualidade – Qualidade

Ponto de vista: Analista Desenvolvedor

* **P7.** Qual é a distribuição dos defeitos por gravidade? Admita gravidade Alta para defeitos que ocasionem a parada total do sistema, gravidade Média para defeitos que impossibilitam o uso de mais de uma funcionalidade e gravidade Baixa, para defeitos que impossibilitem uma única função.

**R1.** 100% dos defeitos podem ser caracterizados como gravidade média.

* **Q2.** Quantos erros são descobertos e tratados durante os testes antes da entrega?

**R2.** Em média oito falhas são tratadas durante testes, mas não há registros dessa etapa.

**Atributo 7: quantidade de defeitos descobertos em produção**. Obtido através do documento de controle de falhas no final do ciclo.

**Modelo 4:** Distribuição da gravidade dos defeitos descobertos em produção. Deve ser calculado como abaixo:

* Defeitos graves = Nº de defeitos de gravidade alta / nº total de defeitos
* Defeitos graves = Nº de defeitos de gravidade média / nº total de defeitos
* Defeitos graves = Nº de defeitos de gravidade baixa / nº total de defeitos

**Atributo 8: quantidade de defeitos de gravidade alta.** Obtido através do documento de controle de falhas no final do ciclo.

**Atributo 9: quantidade de defeitos de gravidade média.** Obtido através do documento de controle de falhas no final do ciclo.

**Atributo 10: quantidade de defeitos de gravidade baixa.** Obtido através do documento de controle de falhas no final do ciclo.

### Desenvolvimento do plano de Mensuração

Neste passo, para cada medida identificada nos planos GQM do programa de mensuração, é determinada quando, como e por quem os dados são coletados de acordo com o processo de software. A tabela 6 representa o plano de mensuração utilizado neste trabalho.

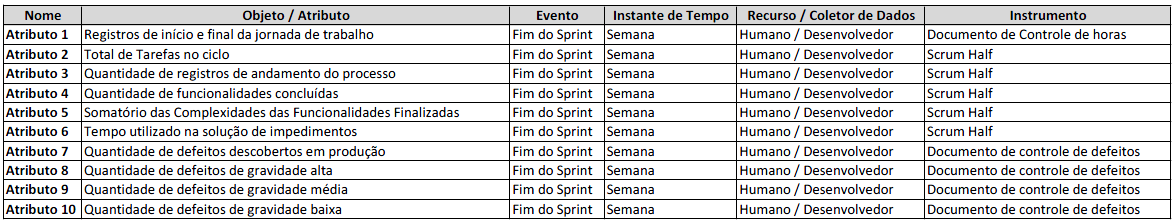


Tabela 6 - Plano de Mensuração

### Coleta dos dados

Os dados foram coletados durante um mês, com o total de 4 Sprints, seguindo o plano de mensuração descrito no item anterior. Os valores coletados estão apresentados na tabela 7:

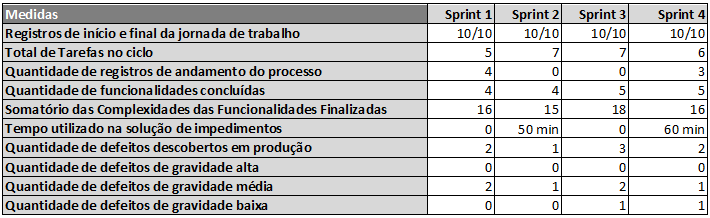


Tabela 7 - Valores Coletados

### Análise e Interpretação dos dados

Nessa etapa, é realizada a análise e interpretação dos dados coletados.

#### Análise e Interpretação do enfoque de qualidade Controle

A métrica “registros de controle de jornada por ciclo” foi coletada com o intuito de verificar completude do documento de controle de horas. Analisando os dados ao final de cada Sprint, verificamos que esse documento estava 100% completo, com todos os registros de início e término da jornada de trabalho. Esse resultado já era esperado, visto que esse documento é a principal fonte de informação da G8, para contabilizar as horas de trabalho que servem como base para a cobrança ao cliente. De acordo com o modelo temos:

Completude controle de horas Sprint 1 = 10 / 10 = 100%

Completude controle de horas Sprint 2 = 10 / 10 = 100%

Completude controle de horas Sprint 3 = 10 / 10 = 100%

Completude controle de horas Sprint 4 = 10 / 10 = 100%

A métrica “Registros de Andamento por ciclo” foi coletada para analisar quantos registros de andamento foram realizados, comparando com o total de funcionalidades que foram finalizadas. Essa comparação visa encontrar a completude dos dados de apoio. Analisando os dados verifica-se que o registro do andamento não foi uma prática regular. Esse resultado era esperado, já que se trata de um documento novo para ser alimentando no processo da empresa, não sendo crítico para ambas as empresas. Acredita-se que esse tempo tenha sido utilizado em outra tarefa mais importante. De acordo com o modelo temos:

Completude dos registros das tarefas em andamento Sprint 1 = 4 / 4 = 100%

Completude dos registros das tarefas em andamento Sprint 2 = 0 / 4 = 0%

Completude dos registros das tarefas em andamento Sprint 3 = 0 / 5 = 0%

Completude dos registros das tarefas em andamento Sprint 4 = 3 / 5 = 60%

Seguindo o modelo proposta para o fator Controle, verificamos:

Completude dos documentos de apoio Sprint 1 = 100% (100% dos 70% dos registros de jornada + 100% dos 30% dos registros de andamento).

Completude dos documentos de apoio Sprint 2 = 70% (100% dos 70% dos registros de jornada + 0% dos 30% dos registros de andamento).

Completude dos documentos de apoio Sprint 3 = 70% (100% dos 70% dos registros de jornada+ 0% dos 30% dos registros de andamento).

Completude dos documentos de apoio Sprint 4 = 88% (100% dos 70% dos registros de jornada + 60% dos 30% dos registros de andamento).

De acordo com a informação contida no campo “Impacto na Hipótese de Linha Base” definido para o Abstraction Sheet do fator de qualidade Controle: “quanto mais completo foram os documentos, mais informação disponível para apoio ao controle se tem”.

Observando os valores de completude dos documentos e com base no Impacto na Hipótese de Linha Base informado, pode-se concluir que houve um aumento de 17 % no fator de qualidade Controle.

**Análise e Interpretação do enfoque de qualidade Produtividade**

A medida “Quantidade de Funcionalidades Concluídas” foi coletada com o objetivo de verificar a produtividade, e serve como base para calcular a quantidade de registros de andamento que deveriam ser registrados. Analisando os dados verifica-se que algumas funcionalidades ficaram pendentes. Na verdade esse valor já era esperado. Para aproveitar o tempo disponível no Sprint por inteiro, se todas as tarefas derivadas das histórias previstas para o *Sprint* foram concluídas, adiciona-se outra história a Sprint para alocar o tempo restante. Quase sempre, acontece não ser possível concluir as tarefas incluídas ao fim do sprint, ficando essas como tarefas não concluídas.

Funcionalidades Concluídas Sprint 1 = 4

Funcionalidades Concluídas Sprint 2 = 4

Funcionalidades Concluídas Sprint 3 = 6

Funcionalidades Concluídas Sprint 4 = 5

Média ~= 5

A medida ”Somatório das complexidades” foi coletada para obter a velocidade de produção baseada nas complexidades definidas para as histórias adicionadas. De acordo com o modelo temos:

Somatório complexidade Sprint 1 = 16 pontos.

Somatório complexidade Sprint 2 = 15 pontos.

Somatório complexidade Sprint 3 = 18 pontos.

Somatório complexidade Sprint 4 = 16 pontos.

Média ~= 16

A medida “Tempo utilizado na solução de impedimentos” visa identificar quanto tempo é perdido por causa de algum problema que surge durante o Sprint. Com base nos dados, observa-se um acúmulo de impedimentos periodicamente. Isso acontece no período de atualização das filiais. Como as filiais são atualizadas remotamente, o processo de atualização depende da permissão de acesso ao servidor e uma parada momentânea nos serviços executados com o sistema.

Tempo de solução de impedimentos Sprint 1 = 0

Tempo de solução de impedimentos Sprint 2 = 50 minutos

Tempo de solução de impedimentos Sprint 3 = 0

Tempo de solução de impedimentos Sprint 4 = 60 minutos

A medida “Total de Funcionalidades no ciclo” foi coletada com o intuito de analisar a quantidade de tarefas que se estima produzir por ciclo, e poder informar quantas não foram implementadas. Analisando os dados, verifica-se um aumento de funcionalidades que são alocadas a cada ciclo.

Funcionalidades Sprint 1 = 5

Funcionalidades Sprint 2 = 7

Funcionalidades Sprint 3 = 7

Funcionalidades Sprint 4 = 6

Média ~= 6

De acordo com a informação contida no campo “Impacto na Hipótese de Linha Base” definido para o Abstraction Sheet do fator de qualidade Produtividade: “quanto mais impedimentos, menos tempo restará para desenvolver as funcionalidades, e menos funcionalidades serão desenvolvidas do que o estimado”. Essa hipótese é confirmada observando que nos ciclos em que houve impedimentos, uma quantidade maior de funcionalidades não foram implementadas.

Observando a média das funcionalidades concluídas e a média do somatório das complexidades, e comparando-as com a Hipótese de Linha Base informada no fator de qualidade Produtividade, pode-se concluir que estão dentro do previsto, ou seja, não sofreram modificações significativas em seus valores.

**Análise e Interpretação do enfoque de qualidade Produtividade**

A medida “Quantidade de defeitos descobertos em produção” auxilia na definição da qualidade do produto entregue. Serve de base para saber a distribuição dos defeitos por gravidade e como informação de apoio. Analisando os dados, verifica-se que não há um grande número de defeitos durante o ciclo de desenvolvimento.

Analisando os dados, verifica-se que nenhum defeito grave foi registrado, a maioria dos defeitos identificados é definido como de gravidade média, e o aparecimento de defeitos definidos como de gravidade baixa.

Ao se comparar a hipótese de linha base do fator de qualidade Qualidade, com os valores obtidos com respeito aos defeitos descobertos em produção, podemos concluir que houve um aumento nos defeitos considerados de baixa gravidade, e uma diminuição dos defeitos de gravidade média, de maneira que:

75% dos defeitos - Gravidade Média

25% dos defeitos - Gravidade Baixa

0% dos defeitos - Gravidade Alta

Ao se comparar a quantidade de defeitos da hipótese de linha base com a média dos defeitos registrados nas coletas, temos:

Hipótese: média de 2 defeitos por ciclo.

Média da coleta: 2 defeitos por ciclo.

Apesar de identificarmos uma leve baixa na gravidade geral dos defeitos, não podemos concluir que esse resultado foi obtido devido ao uso da metodologia SCRUM. Com base no impacto na hipótese de linha base registrada para o fator de qualidade Qualidade, a realização de testes antes da conclusão das novas funcionalidades tem impacto na quantidade de defeitos que são entregues junto com a nova versão do sistema. Apesar de não ser documentado, os testes já eram realizados antes da implantação da metodologia, e é a efetividade desses testes, que tem impacto direto na quantidade de defeitos entregues.

### Captura de Experiência

Durante o desenvolvimento do plano GQM, foi definido os modelos organizacionais, como modelo de controle, produtividade e qualidade. Uma compreensão geral do produto e do processo foi conseguida. As tarefas que compõem o processo de desenvolvimento foram claramente definidas, juntamente com a identificação de valores base para os modelos, o que permite estimativas mais corretas.

Novas ferramentas foram incorporadas ao processo de software. O Scrum Half mostrou ser uma ótima ferramenta de apoio ao projeto, baseada na metodologia Scrum Com esta ferramenta, vários relatórios podem ser gerados com respeito ao andamento do processo. Outra ferramenta incorporada foi o documento de controle de defeitos, que permite registrar e categorizar os defeitos encontrados em produção.

Com a intenção de verificar quanto tempo foi dedicado na aplicação do plano GQM, foi registrada a duração de cada etapa da abordagem. Ao fim, pode-se verificar que foi dedicado um total de 24 horas , distribuídos conforme apresentado na Figura 5.

Figura 5 - Distribuição do Esforço durante a Abordagem GQM

A adoção da metodologia de desenvolvimento Scrum pode ser avaliada no contexto da micro empresa. Através da abordagem GQM aplicada, chegou-se a conclusão que o principal objetivo organizacional, descrito no início do plano GQM como: “ter um melhor controle sobre o processo de desenvolvimento”, pode ser alcançado, sem perdas significativas em outros fatores de qualidade como produtividade e qualidade.

Na análise e interpretação dos dados, sugestões de melhorias puderam ser identificadas com respeito aos modelos de produtividade e qualidade. Dessa forma, a abordagem GQM pode ser continuada visando a melhoria contínua do processo.

# Conclusão

Este trabalho estudou a metodologia de desenvolvimento SCRUM, abordando suas etapas, papeis de responsabilidades e artefatos produzidos. Esse conhecimento serviu como base para usar a metodologia no projeto do estudo de caso proposto.

Foi discutida também, uma abordagem baseada em metas de melhoria chamada de GQM (Goal – Question - Metric). Tal abordagem tem o objetivo de mensurar e guiar o aperfeiçoamento do processo ou produto, de acordo com as metas de melhoria organizacional. Foi utilizada para mensurar o ganho ou perda relacionados aos fatores de qualidade referentes à meta definida.

No estudo de caso relatado neste trabalho, uma micro empresa chamada G8 Consultoria, atuante na região metropolitana de Recife – PE, deu início ao uso da metodologia de desenvolvimento SCRUM em um de seus projetos em andamento. Algumas mudanças foram realizadas na metodologia para se adaptar ao contexto organizacional e também as características do projeto.

Para avaliar se a adoção do SCRUM trouxe melhorias para o processo de desenvolvimento, a abordagem GQM foi adotada. O passo a passo descrito neste trabalho para aplicação da abordagem GQM foi seguido. Metas de melhoria foram identificadas com base em fatores de qualidade. Três fatores da qualidade foram identificados: Controle, Qualidade e Produtividade. Os dois primeiros fatores não tiveram mudanças significativas em seus valores. Observando os valores de completude dos documentos e com base no Impacto na Hipótese de Linha Base, pode-se concluir que houve um aumento de 17 % no fator de qualidade Controle. Com base nessas informações, a adoção da metodologia ágil SCRUM teve um impacto positivo, ou seja, houve uma melhoria no processo de desenvolvimento.

## Contribuições

Dentre as contribuições deste trabalho, destacam-se:

* Definição de um modelo inicial de controle, produtividade e qualidade, indicado para micro e pequenas empresas.
* Adaptações necessárias na metodologia SCRUM e na abordagem GQM para equipes de apenas um único desenvolvedor.

## Trabalhos Futuros

O estudo de caso apresentado neste trabalho avaliou o uso da metodologia SCRUM através da abordagem GQM de apenas um projeto e uma empresa. Seria interessante conduzir mais estudos para validação do fato em outros contextos, que possivelmente nos trariam outras metas a serem alcançadas.

Outra possibilidade para trabalhos futuros é estudar melhor o processo de definição, coleta e adaptações das medidas derivadas durante a abordagem GQM. Ferramentas que agilizem o registro de algumas medidas por parte do time, e que tragam mais transparência ao processo de coleta de dados podem ser criadas.

Foi utilizada durante o estudo de caso, uma ferramenta chamada Scrum Half, que gerencia as etapas da metodologia ágil Scrum. Outra possibilidade de trabalho futuro é o estudo e criação de ferramentas que permitam tal gerenciamento, focando alguns pontos de melhoria. Outra automatização proposta seria a criação de uma ferramenta que gerenciasse as etapas da abordagem GQM, assim como o Scrum Half é para a metodologia SCRUM.

# Referências

[Differding e Hoisl 95] C. Differding, B. Hoisl, C. Lott. Technology Package for the Goal/Question Metric Paradigm. Technical Report 281-96, University of Kaiserslautern, Department of Computer Science, D-67653 Kaiserslautern, Germany, 1995.

[Nasa 94] National Aeronautics and Space Administration. Software Measurement Guidebook. Technical Report SEL-84-101, NASA Goddard Space Flight Center, Greenbelt MD 20771, 1994.

[Varaschi 2009] Varaschi, J. D. Implantando o SCRUM em um Ambiente de Desenvolvimento de Produtos para Internet. PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO DE JANEIRO, Brasil, Rio de Janeiro, fevereiro de 2009.

[Gresse 98] C. Gresse von Wangenheim. Utilização do GQM no Desenvolvimento de Software. Universidade do Vale do Rio dos Sinos, Curitiba, Brasil, 1998.

[Günther, Rombach, Ruhe 94] H. Günther, H. D. Rombach, G. Ruhe. Melhoria contínua da qualidade no desenvolvimento de software - Experiência na Allianz Life Insurance Company, Alemanha, 1994.

[Daskalantonakis 92] M.K. Daskalantonakis. A Practical View of Software Measurement and Implementation Experiences within Motorola. IEEE Transactions on Software Engineering, Vol. 18, No. 11, 1992.

[Gresse, Hoisl, Wüst 95] C. Gresse, B. Hoisl, J. Wüst. A Process Model for GQM- Based Measurement. Technical Report STTI-95-04-E, Software Technology Transfer Initiative, University of Kaiserslautern, Department of Computer Science, Kaiserslautern, Germany, 1995.

[Solingen 95] R. van Solingen. Goal-Oriented Software Measurement in Practice: Introducing Software Measurement in Schlumberger Retail Petroleum Systems. Master Thesis Report, Schlumberger RPS, Netherlands, 1995.

[Briand, Differding, Rombach 97] L. C. Briand, C. M. Differding and H. D. Rombach. Practical Guidelines for Measurement-Based Process Improvement. Software Process Improvement and Practice, vol. 2, 1997.

[Hoisl, Oivo, Ruhe 96] B. Hoisl, M. Oivo, G. Ruhe, and F. van Latum. No Improvement without Feedback: Experiences from goal-oriented measurement at Schlumberger. Fifth European Workshop on Software Process Technology, Nancy, France, October 1996.

[Bröckers, Differding, Threin 96] A. Bröckers, C. Differding and G. Threin. The Role of Software Process Modeling in Planning Industrial Measurement Programs. In Proceedings of the METRICS´ 96. ICSE, 1996.

[Soares 2010] M. S. SOARES. Comparação entre Metodologias Ágeis e Tradicionais para Desenvolvimento de Software. Unipac – Universidade Presidente Antônio Carlos, Faculdade de Tecnologia e Ciências de Conselheiro Lafaiete. Disponível em <http://www.dcc.ufla.br/infocomp/artigos/v3.2/art02.pdf>. Acessado em junho de 2010.

[Sommerville 2003] Sommerville, I. Engenharia de Software. 6ª Ed. São Paulo, 2003.

[Schwaber 2004] K. Schwaber, Agile Project Management with Scrum. Microsoft Press, 2004.

[Schwaber e Beedle (2002)] K. Schwaber, M. Beedle, Agile Software Development with Scrum. Prentice-Hall. 2002.

[Cohn 2005] Cohn, M. Agile Estimating and Planning. Prentice-Hall. 2005.

[Basili, Rombach 1988] V. Basili, H.D. Rombach. The TAME Project: Towards Improvement-Oriented Software Environments, IEEE Transactions on Software Engineering, vol. 14(6), June 1988.