

Análise de Custo e Benefício de Mensuração Baseada em GQM - Um Estudo de Caso Replicado -

Christiane Gresse von Wangenheim¹, Günther Ruhe²

¹Programa de Pós-Graduação em Engenharia de Produção
Universidade Federal de Santa Catarina,
Centro Tecnológico - Campus Universitário
Trindade - Cx. Postal 476 - CEP 88040-900
Florianópolis, Brasil
gresse@eps.ufsc.br

²Fraunhofer-Institute for Experimental Software Engineering (IESE),
Sauerwiesen 6,
D-67661 Kaiserslautern, Alemanha
ruhe@iese.fhg.de

Resume

For continuous improvement of software quality and productivity, software measurement taking into account the environment and focusing on the company-specific goals is essential. A goal-oriented approach for measurement, is the Goal/Question/Metric (GQM) paradigm, which has been successfully applied in industrial strength projects on a large spectrum of quality goals in software development and maintenance. However, the applicability of new technologies in practice requires to be evaluated by experimental investigations. In this paper, the return of investment of GQM-based measurement is studied concerning cost and benefits. Results of a replicated case study in different industrial environments are reported from the ESPRIT-ESSI project CEMP (including software organizations of Robert Bosch GmbH, Digital SPA, Schlumberger RPS). Influencing factors are identified through the careful analysis and comparison of commonalities and differences between different organizations.

Palavras Chaves: Software quality, Goal-oriented measurement, GQM, Cost/benefit, Case study, Industrial application

1 Introdução

Quase qualquer negócio hoje em dia envolve o desenvolvimento ou utilização de software, e, em função disso, o significado da qualidade e produtividade do negócio de software aumentou e tornou-se de importância essencial para a competitividade de uma organização. Porém, o estado-da-prática é que sistemas de software freqüentemente carecem de qualidade e muitos projetos de software são muito demorados e terminam fora do prazo [6]. Muitas organizações estabeleceram processos de desenvolvimento e manutenção em um maneira *ad hoc*, baseadas em experiências de outras empresas, o que freqüentemente conduziu a soluções incompatíveis e conseqüências negativas.

Para continuamente melhorar qualidade e produtividade de software, é necessário entender os produtos e processos de software específicos a uma organização e o impacto de fatores de processo, produto e recursos nestes produtos e processos. Isto requer que organizações capturem explicitamente *know-how* de desenvolvimento de software (por exemplo, *know-how* relacionado aos modelos de

custo ou distribuições de defeitos) e reutilizem esse conhecimento durante o planejamento, controle e melhoramento de projetos. Uma abordagem para a melhoria contínua e execução sistemática de projetos de software é o *Quality Improvement Paradigm* (QIP) [1]. O QIP é um processo iterativo que inclui o planejamento, execução, e avaliação de melhorias para ambientes de desenvolvimento de software, como também a incorporação de experiências ganhas através de esforços de melhoria em projetos. Isto é sustentado pela *Experience Factory* (EF) [1], uma organização dentro da empresa que apóia o projeto de software analisando e sintetizando todos os tipos de experiências, agindo como um repositório para o conhecimento, e prove experiência para projetos em demanda. Para a aplicação do QIP, o estabelecimento de programas de mensuração orientados em metas, é essencial derivar dados quantitativos e qualitativos pertinentes aos processos e produtos de software como uma base para melhoria. Muitos métodos, modelos, e medidas de software, foram desenvolvidos, cada um se adequa a analisar e melhorar alguma característica de software, seu desenvolvimento ou manutenção. Durante os últimos anos, várias empresas introduziram mensuração de software para aumentar a produtividade e melhorar a qualidade dos seus produtos. Mas, muitos problemas ocorreram, resultando da aplicação de uma tecnologia imprópria na organização específica. Para assegurar que a introdução de uma tecnologia conduza a melhorias, tecnologias novas e os seus efeitos nos processos e produtos organizacionais precisam ser avaliados [4,13]. Para possibilitar um uso eficiente e efetivo de tecnologias de software na prática, nós precisamos responder a perguntas como:

- Que restrições e condições existem em relação à aplicação da tecnologia?
- Como seu uso é medido quantitativamente e qualitativamente?
- Quais custos originam?
- Quais são os benefícios esperados?

Só experimentos e outros tipos de estudos empíricos permitem avaliar a efetividade do uso de tecnologias, métodos e ferramentas em ambientes industriais.

O princípio fundamental da ciência, quase a sua definição, é isto: o único teste da validade de qualquer idéia é a experimentação. (R. Feynman)

Infelizmente, hoje especialmente na área de engenharia de software, existe uma grande necessidade de execução de estudos empíricos.

Neste contexto, nós enfocamos no uso e avaliação experimental da relação de custo e benefício de um método para mensuração orientada a metas. O *Goal/Question/Metric* (GQM) [3] é um enfoque de mensuração orientada em metas que ajuda na definição e implementação de um espectro amplo de metas de melhoria de software operacionais e mensuráveis, isto em todo o processo de desenvolvimento de software [8,11]. Como o método GQM é uma tecnologia de software relativamente nova, sua introdução na prática industrial precisa ser avaliada. Neste contexto, nós apresentamos os resultados de um estudo de caso replicado que foi executado em três ambientes industriais diferentes no projeto ESPRIT/ESSI CEMP (incluindo as organizações de software de Robert Bosch GmbH, Digital SPA, Schlumberger RPS).

A avaliação empírica do método GQM aponta para a comparação de dados quantitativos de custo relacionados à introdução de um programa de mensuração baseado em GQM na prática com dados principalmente qualitativos dos benefícios alcançados. A repetição dos estudos de casos habilita-nos a tecer uma análise cuidadosa de diferenças e similaridades entre organizações diferentes. Foram identificados fatores de influência e *cost drivers* na comparação e em relação aos

contextos das organizações. Os resultados das avaliações empíricas são essenciais para a aceitação e aplicação ampla da tecnologia GQM. Além disso, os estudos também demonstram como avaliações empíricas em forma de estudos de caso replicados podem prover declarações validadas em relação à avaliação de tecnologias, métodos e ferramentas.

2 A Abordagem GQM

O Paradigma *Goal/Question/Metric* (GQM) [2,3] é uma abordagem de mensuração orientada a metas que apóia a definição e implementação *top-down* de metas operacionais e mensuráveis para melhoria de software e a interpretação *bottom-up* dos dados coletados. O processo GQM [3,7] cobre o planejamento, execução de um programa de mensuração e a captura de experiências para reutilização futura, por exemplo em forma de modelos ou diretrizes. Os passos principais do processo GQM são:

GQM1 – Estudo prévio. O ambiente onde o programa de mensuração está estabelecido é caracterizado e um projeto de piloto para o estabelecimento de um programa de mensuração é selecionado. O pessoal envolvido é motivado e treinado.

GQM2 - Identificação de metas GQM e desenvolvimento de plano GQM.

GQM2A - Identificação de metas GQM. Baseado nas metas do negócio, metas da melhoria e problemas existentes no processo de software, metas de mensuração potenciais são identificadas por todas as pessoas envolvidas no projeto-piloto. A definição das metas GQM inclui as seguintes dimensões [3]:

Dimensão	Definição	Exemplo
Objeto de Estudo	O que será analisado?	Processo de desenvolvimento
Objetivo	Porque o objeto será analisado?	Caracterização
Enfoque de Qualidade	Qual atributo do objeto será analisado?	confiabilidade
Ponto de Vista	Quem será usar os dados coletados?	Equipe
Contexto	Em qual ambiente está localizado?	Empresa x, Projeto A

GQM 2B - Desenvolvimento de plano GQM. Baseado na meta, um plano GQM [3,7] é desenvolvido, o qual consiste de uma meta (como definida em GQM2A), um conjunto de perguntas, um conjunto de modelos, e um conjunto de medidas. Para refinar a meta transformando-a em medidas operacionais, o enfoque de qualidade precisa ser definido adequadamente e ajustado à necessidade do ponto de vista.

Objeto	Objetivo	Enfoque de Qualidade	Ponto de Vista	Contexto
processo de software	caracterizar	confiabilidade	equipe	Empresa x, Projeto A
Fatores de Qualidade		Fatores de variação		
- número total de defeitos - número dos defeitos por criticidade (não-crítico, crítico) - número de defeitos por fase de ciclo da vida introduzido (REQ, HLD, LLD/IMP) - esforço total de <i>rework</i> (em horas)		- tipo de inspeção de código - experiência de pessoas da equipe de desenvolvimento		
Hipótese de Linha-Base		Impacto na Hipótese de Linha-Base		
- número total de falhas: 120 - 75% não-crítico, 25% crítico - REQ 10, HLD 20, LLD/IMP 40, TESTE 30 - esforço total de <i>rework</i> : 1000 h		- com ad-hoc inspeções de código menos defeitos serão detectados do que com outros tipos de inspeções. - pessoas da equipe de desenvolvimento mais experientes introduzem menos defeitos.		

Figura 1. Exemplo de *Abstraction Sheet*

Conhecimento pertinente é adquirido entrevistando-se as pessoas declaradas no ponto de vista da meta GQM. Esta informação é usada para derivar perguntas e modelos de qualidade válidos e corretos, identificar fatores de contexto relevantes e medidas pertinentes. As entrevistas cobrem os seguintes tópicos [8]:

- Enfoque da qualidade: Especifica o que o enfoque de qualidade significa para o entrevistado em dimensões de qualidade.
 - Hipótese de linha-base: Para cada dimensão de qualidade que pertence ao enfoque de qualidade, uma distribuição esperada de valores pode ser declarada.
 - Fatores de variação: Fatores que têm um impacto nas dimensões de qualidade são declarados.
 - Impacto na hipótese de linha-base: Para cada fator de variação, deveria ser especificado o impacto esperado do fator de variação na dimensão de qualidade.
- Um instrumento comumente usado para a aquisição e que suporta a estruturação de conhecimento durante as entrevistas é o *Abstraction Sheet* [8] (veja Figura 1). O plano GQM é desenvolvido baseado no conhecimento adquirido durante as entrevistas. Para cada dimensão de qualidade e fator de variação documentado no *Abstraction Sheet*, uma pergunta é refinada no plano GQM, expressando a necessidade por informação [3,7]. As perguntas representam um refinamento intuitivo da meta de mensuração. Uma vez que planos GQM normalmente consistem

<p>Q1. Qual é o número total de defeitos antes de entrega? <i>Modelo: Número total de defeitos pode ser calculado com base no número total dos relatórios de defeitos.</i> M1.1 contador de relatórios de defeitos informados antes de entrega [taxa: inteira]</p> <p>Q2. Qual é a distribuição de defeitos informados antes da entrega por criticidade? <i>Modelo: Distribuição = (# defeitos críticos/ total # defeitos, # defeitos não crítico/ total # defeitos)</i> <i>crítico: breakdown completo do sistema; não crítico: impossível de executar uma ou mais de funções F1-F6.</i> M2.1 classificação dos relatórios de defeitos encontrados antes da entrega por criticidade [ordinal:não-crítico; crítico; outros] M2.2 número total dos defeitos detectados antes de entrega [taxa: inteira]</p> <p>Q3. Qual é a distribuição de defeitos por fase de ciclo de vida da introdução detectados antes de entrega? <i>Modelo: Distribuição = (# defeitos em REQ/ total # defeitos, # defeitos em HLD/ total # defeitos, # defeitos em LLD/IMP/ total # defeitos)</i> M3.1 contador de defeitos por fase de ciclo de vida onde o defeito foi introduzido [nominal: REQ, HLD, LLD/IMP]</p> <p>Q4. Qual é o esforço total de rework? <i>Modelo: esforço de rework = (esforço de isolar defeito + esforço de corrigir defeito)</i> M4.1 Cada defeito informado antes de entrega: esforço para isolar as causas [taxa: inteira (pessoa-hora)] M4.2 cada defeito informados antes de entrega: esforço para corrigir o defeito [taxa: inteira (pessoa-hora)]</p>

Figura 2. Exemplo simplificado do plano GQM

de grandes números de perguntas, categorias de perguntas [3] foram definidas, as quais suportam a derivação e estruturação de planos GQM. Continuando o exemplo acima, perguntas e hipóteses são ilustradas na Figura 2. Cada pergunta no plano GQM é formalizada e modelos de qualidade ou recursos são definidos [3]. Esses provêm meios para responder as perguntas, como elas quantificam os vários atributos abstratos dos artefatos a serem estudados e definem como as comparações, avaliações, e predições serão executadas [3] (veja Figura 2). As perguntas são quantitativamente refinadas em um conjunto de medidas pelos modelos [3]. Para um determinado atributo a ser medido, a definição de uma medida inclui a definição dos dados a serem coletados, além da seleção de um nível de medida, unidade, e faixa (veja Figura 2). O plano GQM é refinado iterativamente enquanto inconsistências e aspectos não claros existem. Quando o plano GQM é completado, ele é revisado com respeito à correção e completude.

GQM3 - Desenvolvimento de plano de mensuração. O enfoque principal do plano de mensuração é a integração apropriada de medidas no processo de desenvolvimento de software do projeto piloto. Procedimentos de coleta de dados são definidos determinando para cada medida identificada nos planos GQM, quando, como, e por quem os dados serão coletados [3,7].

Para implementar os procedimentos de mensuração, os instrumentos de coleta de dados exigidos precisam ser desenvolvidos. Dependendo do tipo de instrumento de

coleta de dados, uma ferramenta tem que ser desenvolvida, questionários têm que ser projetados ou estruturas de entrevistas têm que ser planejadas.

Nome	Evento	Instante de tempo	Recurso/ coletor de dados	Instrumento
Esforço do projeto	Periódica	semana	Humano/desenvolvedor	Questionario_esforço
Criticalidade de defeitos	Fim do processo	processamento falhas	Humano/Testador	Questionario_defeito
Introdução de defeito	Fim do processo	processamento falhas	Humano/desenvolvedor	Questionario_defeito

Tabela 1. Exemplo simplificado de plano de mensuração

GQM4 - Coleta de dados. Durante a fase de execução do programa de mensuração o dados são coletados de acordo com os procedimentos especificados no plano de mensuração.

GQM5 - Análise e interpretação. Seguindo o plano GQM *bottom-up*, o dados coletados são analisados quantitativamente por análises estatísticas ou qualitativamente (por exemplo usando a *Rough Set Theory* [12]). O dados analisados são interpretados em *Feedback sessions* [9] que envolvem todas as pessoas envolvidas no programa de mensuração. Potencial de melhoramento é identificado e ações de melhoria são determinadas.

GQM6 - Captura de experiências. Os resultados da análise e interpretação são capturados sob a forma de modelos e experiências, armazenados na base de experiências para serem disponíveis para suportar projetos futuros.

3. Validação de Tecnologias em Estudos de Caso Replicados

Para validar tecnologias, nós temos que executar estudos empíricos [4]. Estudos empíricos variam de experimentos formais, estudos de caso a pesquisas. Um estudo de caso é compreendido como uma avaliação de um problema por um grupo de pessoas em um projeto específico. No domínio de engenharia de software, estudos de caso provaram ser poderosos e informativos. Embora eles não alcancem o rigor científico de experimentos formais, estudos de caso podem prover informação suficiente para ajudar julgar se tecnologias específicas serão benéficas para uma organização ou um projeto particular.

Porém, com relação à derivação de resultados geralmente aplicáveis, nem experimentos formais nem estudos de caso provaram ser suficientemente significantes para a validação de tecnologias inovadoras na prática. A dificuldade é derivar declarações generalizadas na área de pesquisa empírica no domínio de engenharia de software. Isto ocorre em função de repetições mal feitas de experimentos, da dinâmica dos sistemas estudados, e da variedade de fatores de influência nos sistemas. Assim, uma possível solução é a repetição de um estudo de caso em vários contextos para aumentar a validade dos resultados dos estudos. Isto significa, que um estudo de caso é concluído em paralelo em organizações diferentes, mas enfocando no mesmo objetivo. Isto supõe um procedimento coordenado temporal e metódico, com uma compreensão clara do tipo e âmbito das diferenças nas restrições e fatores de influência essenciais. Resultados atingidos nos ambientes individuais podem ser comparados com relação as diferenças e aspectos em comum. Divergências entre os resultados procura-se explicar pelas características respectivas de contexto.

A abordagem principal nos estudos de caso replicados é orientada no QIP que apóia o desenvolvimento de um corpo empiricamente justificado de conhecimento para o desenvolvimento e melhoramento sistemático de software. Para a avaliação do método GQM, um estudo de caso replicado foi executado no contexto do projeto

CEMP (*Customized Establishment of Measurement Programs*) [5], um projeto do *European System and Software Program* (ESSI) da União européia. Nesse projeto a introdução de mensuração baseada em GQM relacionado aos aspectos de confiabilidade e reusabilidade de software foi avaliada em três empresas em paralelo: Bosch GmbH, Digital SPA, e Schlumberger RPS.

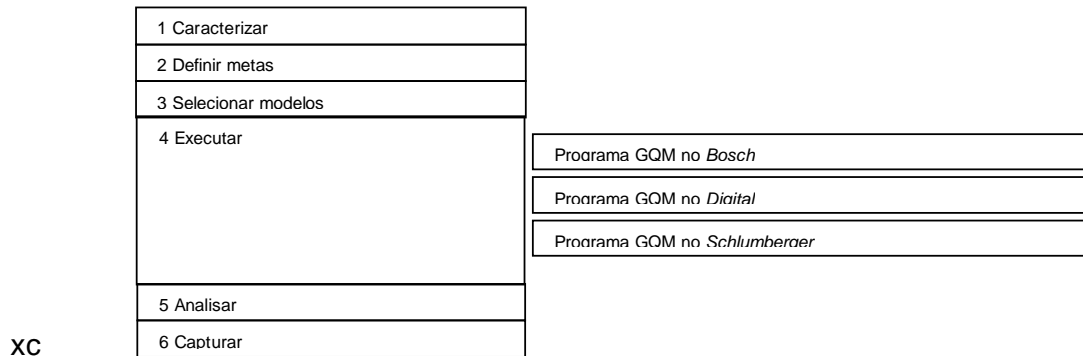


Figura 3. Projeto experimental do estudo de caso replicado

Um estudo geral foi planejado para todos os três empresas em cooperação. Posteriormente o estudo foi executado em paralelo nos três ambientes e os dados exigidos foram coletados em cada local. Os dados coletados foram analisados, interpretados e comparados entre os locais. Os aspectos em comum e diferenças entre os três ambientes experimentais foram analisados com base em uma metodologia comum, o paralelismo dos estudos, o objeto comum de investigação, e a descrição cuidadosa dos ambientes experimentais. Generalizações estão baseadas em experiências comuns em ambientes diferentes que oferecem assim um potencial de reutilização de experiências mais amplas dentro e fora das empresas participantes. Uma visão geral da estrutura do estudo de caso replicado é mostrada em Figura 3.

4. Análise de Custo e Benefícios de Mensuração Baseado em GQM

4.1 Análise de Custo

Para a análise de custo relacionada ao estabelecimento de um programa de mensuração baseado em GQM um modelo detalhado de custo foi desenvolvido. Derivado das entrevistas feitas nas organizações, o custo da introdução de GQM foi definido como a soma do esforço necessário para as fases de um programa de mensuração baseado em GQM e para todo o pessoal envolvido. Assim, custo é compreendido como o esforço requerido (pessoas-hora) para o programa de mensuração.

Os resultados seguintes da análise do custo relacionado à introdução de mensuração baseado em GQM são generalizados a partir da análise paralela pelas empresas [5]. Todos os valores são médias dos resultados dos locais e projetos individuais e incluem só atividades diretamente relacionadas ao processo:

- O esforço total necessário para introduzir mensuração baseado em GQM é aproximadamente 1 pessoa-ano. Isto inclui o esforço total do grupo de garantia de qualidade e do pessoal de projeto piloto no planejamento e na execução de um programa de mensuração.

- O esforço total necessário para introduzir GQM de aproximadamente 1 pessoa-ano vem de 1/3 do pessoal de projeto piloto e 2/3 do grupo de garantia de qualidade.
- As contribuições relativas ao esforço às fases do processo GQM do grupo de garantia de qualidade são mostradas na tabela 2 para cada um dos projetos piloto no projeto CEMP. É óbvio, que durante o planejamento do programa GQM, a execução das entrevistas e o desenvolvimento do plano GQM, junto com a análise e interpretação dos dados de medida em a maioria dos casos requerem 2/3 do esforço total.

Projeto	Treinamento	Estudo Prévio	Identificação de metas GQM	Des. de plano GQM	Des. de plano de mensuração	Coleta, análise e Interpretação
A1	11.1	1.3	2.9	40.0	22.5	78.6
A2	6.3	0.5	3.8	45.6	34.9	--
B1	33.5	26.1	23.9	102.7	32.0	105.1
B2	--	9.5	9.5	35.2	7.3	92.5
C1	7.2	7.8	3.8	55.5	41.6	78.3
C2	3.9	3.8	5.0	25.9	7.6	24.2

Tabela 2. Esforço relacionado à mensuração por fase (em pessoas-dia)¹

- Como o esforço total da fase planejamento e execução é aproximadamente 1 pessoa-ano, existe um alcance relativamente amplo de aproximadamente 4 - 11 pessoas-mese relacionado ao esforço do planejamento entre os projetos-piloto, que é explicado pelos fatores de variação seguintes:
 - Grau de envolvimento do pessoal de projeto em atividades do processo GQM
 - Características da equipe do projeto como tamanho e distribuição em locais
 - Escopo do programa de mensuração
 - Experiência do grupo de garantia de qualidade
- O esforço total para o planejamento e a execução de um programa de mensuração diminui nos programas seguintes. No projeto CEMP, as empresas gastaram aproximadamente 6 pessoas-mês nos segundos projetos comparado com aproximadamente 12 pessoas-mês nos primeiros projetos. A razão principal para a diminuição é a disponibilidade de apoio de ferramenta para a tecnologia GQM, experiência aumentada com o a tecnologia GQM e reutilização de artefatos relacionados ao programa de mensuração.
- O esforço adicional do projeto devido à mensuração baseada em GQM para só o pessoal do projeto (coleta de dados, etc.) foi menos do que 3% para os primeiros projetos em todos os locais experimentais. Mas, no outro lado, a produtividade da equipe aumentou em um local, porque os desenvolvedores estavam muito contentes de ver que o dados coletados era a causa direta de melhoramentos no ambiente de desenvolvimento deles.
- O esforço total para mensuração baseado em GQM (para a fase do planejamento e execução, inclusive o pessoal de projeto e o grupo de garantia de qualidade) em porcentagem do esforço de projeto de software total é aproximadamente 17%. O valor pode variar muito para um grau alto e pode depender da situação inicial da organização, o escopo do programa de mensuração e do grau das atividades adicionais relacionadas à mensuração.
- No projeto CEMP, o esforço total para mensuração por mês de execução é aproximadamente 0.4 pessoa-mês. O esforço é quase igual nos primeiro e segundo projetos. Isto é importante, porque o esforço total gasto no programa de

¹ Para garantir a confidencialidade dos dados, nós denotamos as organizações pelo A,B, e C com o primeiro projeto piloto 1 (marcado cinzento) e o seguinte projeto piloto 2 (branco).

mensuração depende da duração da fase de execução (período de coleta, análise e interpretação dos dados).

As diferenças parcialmente grandes no esforço são explicadas através de vários fatores influentes. Os fatores seguintes foram identificados, e embora a influência deles não tenha sido provada, eles também foram identificados em estudos diferentes [5, 11]:

Escopo do programa de mensuração:

- Tamanho da organização e número de projetos incluídos no programa de mensuração
- Tamanho do programa de mensuração (número de metas de mensuração, escopo da meta, número de perguntas, número de medidas, etc.).
- Duração do programa de mensuração (p.ex., duração da fase de coleta de dados, frequência de *feedback sessions*)

Características da organização/projeto:

- Disponibilidade de uma infra-estrutura de mensuração (p.ex. grupo de garantia de qualidade)
- Disponibilidade de condições prévias para um programa de mensuração (p.ex. modelo de processo de software)
- Complexidade e estabilidade dos processos de software subjacentes
- Conformidade entre modelo de processo e o processo real
- Tamanho da equipe do projeto e distribuição

Experiência com programas de mensuração:

- Grau de maturidade de mensuração da organização
- Disponibilidade de apoio de ferramenta para mensuração (p.ex. coleta de dados)
- Reutilização de programas de mensuração passados
- Grau de atividades de apoio adicionais (p.ex. implementação de ferramentas)

4.2 Análise de Benefícios

Um problema principal em avaliações de custo-benefício em tecnologias orientadas a melhoramento, como por exemplo mensuração, é que, embora o custo possa ser descrito quantitativamente, a maioria dos benefícios porém, pode ser declarada só qualitativamente, já que a avaliação quantitativa dos efeitos dos benefícios requer um período de observação mais longo.

GQM foi demonstrado como sendo uma abordagem de mensuração poderosa para melhoramento sistemático no domínio de software. Principalmente em todos os estudos a diferença qualitativa entre a tecnologia GQM e outras abordagens usadas anteriormente nas empresas (p.ex. coleta de dados sem orientação em metas, reutilização de modelos sem consideração de contexto e sem envolvimento do pessoal de projeto durante o planejamento de mensuração e interpretação dos dados) ficou clara. Nós descrevemos os benefícios qualitativos mais importantes para as empresas ganhos pela mensuração baseado em GQM ilustrados através de alguns exemplos:

- Identificação de problemas ocultos, os quais ficaram claros através da modelagem do processo e a busca sistemática de relações e dependências. Em um ambiente industrial, o período entre dois marcos internos que correspondem à consolidação de um produto intermediário foi analisado. Este período foi identificado como um gargalo causando demoras desnecessárias no desenvolvimento de produto. Através da mensuração foi demonstrado que este período é 5 vezes maior do que o tempo necessário para localizar e consertar

defeitos. Pela redução desse período entre marcos, uma melhoria do processo de desenvolvimento foi atingida.

- Melhor compreensão dos aspectos e relações relacionados às metas de qualidade confiabilidade e reutilização incluindo uma descrição quantitativa destes. No contexto dos programa de mensuração relacionado à confiabilidade, em um local a distribuição dos defeitos descobertos foi examinada com respeito à criticidade deles. A análise dos dados de mensuração mostrou que a parte dos defeitos fatal se estabiliza no curso do projeto aos 25% (veja Figura 4). Antes do programa de mensuração o pessoal do projeto supunha que só uma parte de 10% de defeitos eram fatais. Pelos dados de mensuração quantitativos disponíveis, foi refutada esta suposição errônea e um modelo correto foi produzido relativo à distribuição de defeitos.

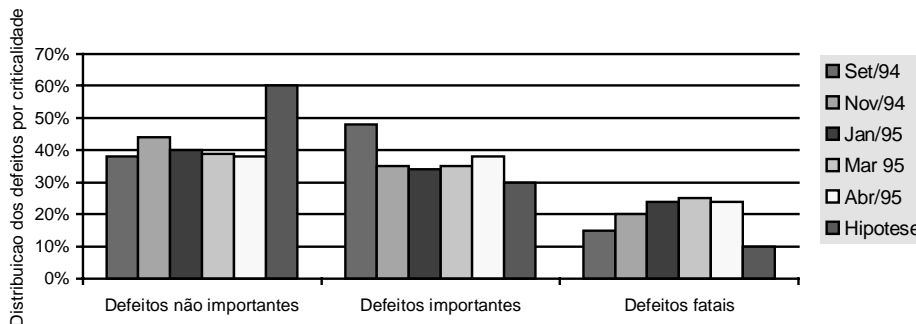


Figura 4. Distribuição de defeitos por criticidade [5]

- Identificação de potencial de melhoramento com impacto médio/alto pela compreensão e exame de dependências básicas. Figura 5 mostra a distribuição de defeitos por função de sistema em um ambiente industrial. Neste ambiente, foi demonstrado que as funções de administração mostram a densidade de defeitos mais alta. Isto é devido à taxa alta de mudança dos requisitos para essas funções. Como resultado desta descoberta, a preparação das especificações foi considerada muito mais importante.

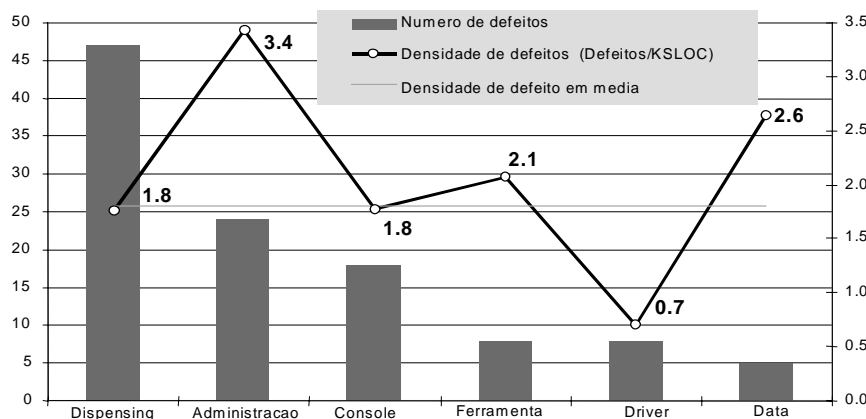


Figura 5. Distribuição de defeitos por função de sistema [4]

- Execução controlada de ações de melhoramento baseadas nos resultados e conclusões ganhas através do programa de mensuração. Com base nos pontos fracos identificados em relação ao processo de software, produtos ou recursos,

melhoramentos eram iniciados e executados. Por exemplo, em uma organização, através da reorganização do processo de teste resultaram uma redução do esforço de teste de 30% [5].

- Uma maior vontade geral para pensar em mudanças em função de um processo compreendido e controlado. Em um ambiente, a efetividade de procedimentos de teste foi examinada no contexto de um programa de mensuração (veja Figura 6). Isto mostrou, que pelo «teste de usuário», num procedimento até lá considerado insignificante, uma parte essencial de defeitos foi detectada. Uma continuação da análise mostrou que uma parte alta dos defeitos descobertos por este procedimento foi classificada como fatal. Como consequência, o processo de teste foi alterado e esse procedimento foi introduzido como uma parte regular do processo de teste na organização.

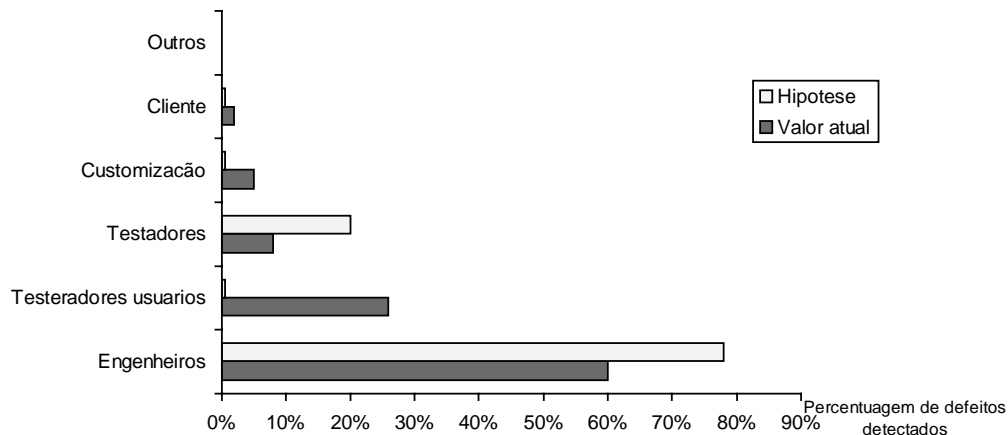


Figura 6. Distribuição de defeitos descobertos por procedimento de teste [5]

- Os resultados e interpretações, resultando do programa de mensuração, representam uma base fundamental para o processo decisório e planejamento de projetos de software. Figura 7 apresenta o esforço em média para a localização e conserto de defeitos em um ambiente industrial. Esses dados mostram, que em média uma semana por mês é usada para a remoção de defeitos. Nesta organização, as regras práticas seguintes foram derivadas com base nestes dados [9]:
 - A densidade de defeitos em média é 1.9 defeitos por KSLOC.
 - A densidade de defeitos das funções de administração é cinco vezes mais alta do que a densidade de defeitos de funções de automatização.
 - O tempo em média para a remoção de defeito é aproximadamente cinco horas para funções de automato e um hora para funções de administração.
- A mensuração baseada em GQM habilitou a possibilidade para examinar projetos que já foram terminados com o novo *know-how* ganho. Na análise retrospectiva de projetos, onde nenhuma interpretação sensível era possível antes, a aplicação de mensuração baseada em GQM permitiu o melhor agrupamento, análise e interpretação de dados.

Os benefícios alcançados pela introdução de mensuração baseada em GQM dependem de certos fatores de contexto. Uma condição prévia importante para o sucesso de um programa de mensuração é o modelo descritivo do processo de desenvolvimento de software. Outros fatores importantes são:

- Envolvimento de todas as pessoas envolvidas no projeto durante o planejamento do programa de mensuração e a interpretação dos dados coletados.
- Quando da introdução da mensuração, deveriam ser enfocadas só poucas metas de medida. Depois de ter terminado com sucesso o projeto piloto, as experiências ganhas podem ser usadas por um estabelecimento passo-a-passo da mensuração na organização inteira.
- Condições prévias da organização, como o comprometimento da gerencia para a introdução e estabelecimento de mensuração.

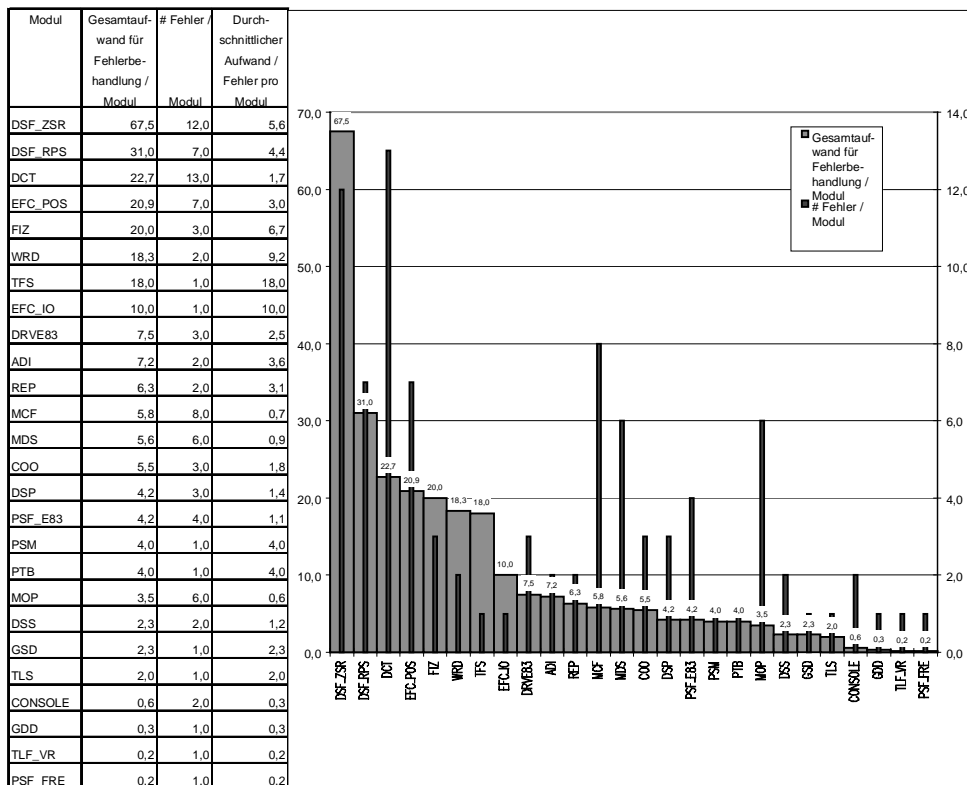


Figura 7: Esforço por remoção de defeitos [9]

5 Conclusões

Com relação à introdução de novas tecnologias com sucesso no desenvolvimento de software comercial, uma análise quantitativa e qualitativa de aspectos relevantes e dos fatores de influência é imperativa. Portanto, estudos experimentais orientados a metas considerando o contexto específico são requeridos. De valor especial são estudos, nos quais com base em experimentos em pequena escala, podem ser feitas conclusões sobre os efeitos da aplicação de uma tecnologia em ambientes industriais. A abordagem de estudos de caso replicados aplicada no projeto CEMP é uma base experimental satisfatória para adquirir resultados generalizados de estudos empíricos com base em modelos adequados. Fatores de sucesso críticos relativos aos estudos de caso replicados são: Base metódica comum do abordagem, paralelismo temporal dos experimentos, objeto de estudo comum enfocando na mesma meta do estudo, como também, análise e interpretação em comum dos resultados ganhos. Enquanto essas restrições de estudos empíricos são consideradas, resultados de relevância mais alta e significado podem ser derivados.

A tecnologia de infra-estrutura GQM contribui de forma importante à transposição do Paradigma QIP para o melhoramento da qualidade e produtividade de processos e produtos de software. Todas as organizações envolvidas no projeto atingiram um grau de maturidade mais alto através do GQM e demonstraram avanços relativos às metas de melhoramento selecionadas.

Agradecimentos

Os autores agradecem todos os participantes do projeto CEMP² (Robert Bosch GmbH, CEFRIEL, Digital SPA e Schlumberger RPS) para a troca contínua de experiências. Nós também agradecemos os colegas do departamento QPE do Fraunhofer Institute IESE e o grupo de Engenharia de Software da Universidade Kaiserslautern para discussões proveitosas. Os autores gostariam de agradecer ao Prof. Dr. Aldo von Wangenheim, do Departamento de Informática e Estatística da UFSC pelas discussões que ajudaram no melhoramento deste artigo.

Referencias

- [1] V. R. Basili, G. Caldiera, H.D. Rombach. *Experience Factory*. In John C. Marciniak (ed.), *Encyclopedia of Software Engineering*, volume 1. John Wiley & Sons, 1994.
- [2] V.R. Basili, G. Caldiera and H. D. Rombach. *Goal Question Metric Paradigm*. In John C. Marciniak (ed.), *Encyclopedia of Software Engineering*, volume 1. John Wiley & Sons, 1994.
- [3] L. C. Briand, C. M. Differding, H. D. Rombach. *Practical Guidelines for Measurement-Based Process Improvement*. *Software Process Improvement and Practice*, vol. 2, 1997.
- [4] V.R. Basili et al. *Experimentation in Software Engineering*. *IEEE Transactions on Software Engineering*, SE-12(7), July 1986.
- [5] CEMP Project. ESSI Project Number 10358. *Customized Establishment of Measurement Programs*, Final Report, July 1996.
- [6] W.W. Gibbs. *Software's Chronic Crisis*. *Scientific American*, November 1994.
- [7] C. Gresse, B. Hoisl, J. Wuest. *A Process Model for GQM-Based Measurement*. Report STTI-95-04-E, Software-Technologie-Transfer Initiative Kaiserslautern, 1995.
- [8] H. Günther et al. *Kontinuierliche Qualitätsverbesserung in der Software-Entwicklung*. *Wirtschaftsinformatik* 38 , 1996.
- [9] B. Hoisl et al. *No Improvement without Feedback: Experiences from Goal-Oriented Measurement at Schlumberger*. Fifth European Workshop on Software Process Technology, France, October 1996.
- [10] F. van Latum et al. *Shifting To Goal-Oriented Measurement In Industrial Environments*. *IEEE Software*, 1998.
- [11] *Software Measurement Guidebook*. Technical Report SEL-94-002, NASA Goddard Space Flight Center, Greenbelt MD 20771, July 1994.
- [12] G. Ruhe. *Qualitative Analysis of Software Engineering Data Using Rough Sets*. The 4th Int. Workshop on Rough sets, Fuzzy Logic, and Machine Discovery, Japan, 1996.
- [13] W.F. Tichy et al. *Experimental Evaluation in Computer Science*. *J. Systems Software*, 28 , 1995.

² promovido em parte pelo projeto de ESPRIT/ESSI #10358 CEMP.