

## 2

### Revisão da literatura

Esse capítulo tem por objetivo apresentar o estado da arte no que diz respeito à utilização de questionários para a elicitación de requisitos, sobretudo requisitos de software, no contexto da Engenharia de Software. Posteriormente, será apresentada a técnica GQM (*Goal Question Metric*), que será utilizada no processo de elaboração de perguntas.

#### 2.1.

#### A utilização de questionários na Engenharia de Requisitos

O trabalho estuda a técnica de questionários para a elicitación de requisitos e sua priorização. Questionários são utilizados em diversas áreas, sobretudo nas Ciências Sociais e *Marketing*, sendo aplicados, por exemplo, em pesquisas de opinião, censo e pesquisas de boca de urna em época de eleição. Questionários também são bastante utilizados como uma técnica auxiliar às técnicas de entrevistas e reuniões, técnicas essas amplamente utilizadas no processo de elicitación de requisitos. Um dos objetivos aqui propostos é estudar e explorar a sua utilização na área de Engenharia de Requisitos, como uma técnica a ser utilizada para a elicitación de requisitos, sobretudo requisitos de software. O objetivo é prover ao Engenheiro de Software uma forma de ajudá-lo no processo de elaboração de questionários, sobretudo no processo de elaboração de perguntas, criação e formatação de questionários.

Para que isso fosse possível, foram realizadas várias pesquisas com objetivo de identificar iniciativas que utilizassem questionários para a elicitación de requisitos de software. A idéia era tentar utilizar ao mesmo tempo técnicas para a formulação de bons questionários (o que não envolve muito o conhecimento técnico, mas sim humano) e as técnicas de Engenharia de Software destinadas a esse fim. Esse processo deu-se através da realização de pesquisas em duas áreas diferentes: pesquisas na área de Engenharia de Software e mais especificamente

na área de Engenharia de Requisitos; pesquisas em outras áreas, tais como: Ciências Sociais e *Marketing* [1-4, 7-8]. As pesquisas realizadas na segunda área resultaram positivamente, tendo o resultado dessas pesquisas gerado uma lista de requisitos para a construção de questionários de qualidade, com boas práticas para a criação e formulação de bons questionários.

Já as pesquisas na área de Engenharia de Software e mais especificamente na área de Engenharia de Requisitos, também foram realizadas pesquisas em diversas fontes, tais como:

- No *site* de buscas do Google;
- Em periódicos, portais, *workshops* e conferências da área, tais como: Workshop on Requirements Engineering [33, 34]; The 16th International Conference on Advanced Information Systems Engineering [25]; Requirements Engineering Journal [38]; REFSQ'04 [39]; IEEE Computer Society [26]; Schloss Dagstuhl International Conference and Research Center for Computer Science [27]; SpringerLink [28]; M.E.Sharpe, Inc. [29]; Volere Requirements Resources [40]; RE 2003 - 11<sup>th</sup> IEEE International Requirements Engineering Conference [41]; RE 2004 - 12<sup>th</sup> IEEE International Requirements Engineering Conference [42]; The Requirements Engineering Specialist Group of the British Computer Society [43]; Journal of Management Information Systems [44]; Journal of Information, Law e Technology [30]; Journal of Information e Knowledge Management [31];
- Nas bibliotecas central e setorial de informática da PUC-Rio;
- Foram trocados e-mails com alguns professores de universidades estrangeiras da área de Engenharia de Software, tais como: Luisa Mich (Department of Information and Communication Technology - University of Trento); Mark Bergman (Information and Computer Science Department - University of California, Irvine); Al Davis (Information Systems College of Business - University of Colorado); Glenn Browne (Texas Tech University); Katrina Hands (Applied Computing – University of Dundee); Sue Lewis (School of Mathematics - University of Southampton); Alistair Sutcliffe (Systems Engineering – University of Manchester); Bashar Nuseibeh (Computing Department – The Open University);

Essas buscas não tiveram um resultado tão positivo quanto o obtido na área de Ciências Sociais, porém, foram identificadas algumas fontes importantes para a pesquisa que, mesmo não estando diretamente relacionadas a elicitación de requisitos através da utilização de questionários (a maioria dos resultados obtidos eram diretamente relacionados às técnicas de entrevista ou de reunião), ajudaram a servir de base para o processo de pesquisa e estudo do estado da arte do assunto.

Foram identificados dois principais artigos e uma dissertação de mestrado, os quais descreviam *cases* que utilizaram indiretamente questionários com esse fim. O primeiro [23], discute o desenvolvimento de uma ferramenta de entrevistas realizadas via computador, com objetivo de facilitar a elicitación de requisitos e conduzir avaliações de usuários. Esse artigo utilizou a técnica de questionários como uma técnica auxiliar, com objetivo de apoiar a técnica de entrevistas. Pesquisas realizadas analisaram as técnicas e processos adotados por entrevistadores especialistas humanos e simularam essas técnicas e processos em softwares. Um projeto de pesquisa foi então desenvolvido como uma investigação preliminar da potencial incorporación de uma ferramenta de entrevistas via computador nas áreas de usabilidade e de Engenharia de Software. O objetivo do projeto, denominado UsE-IT, foi verificar se uma ferramenta de entrevista via computador poderia ajudar a facilitar os processos de obtenção de requisitos e teste com usuário, que poderiam conseqüentemente permitir de forma mais eficiente a criação de um sistema que satisfizesse os usuários. Foi planejado desenvolver uma ferramenta de entrevistas composta de duas partes: uma interface hipermídia para entrevistas e uma interface hipermídia para a análise das entrevistas.

Abaixo é apresentado um resumo dos principais requisitos definidos para o sistema que foi construído, em [23]:

- Ordenação dinâmica das questões: cada questão deveria ter associada a ela um valor de prioridade, que poderia ter o seu valor alterado durante o curso da entrevista dependendo das respostas dadas pelos entrevistados durante a realização da entrevista. Devem ser exibidas aos entrevistados as questões com prioridade mais alta. Logo, as questões devem ser perguntadas em uma ordem apropriada e apenas questões relevantes devem ser perguntadas.

Na figura abaixo é possível visualizar o processo de ordenação dinâmica das questões:

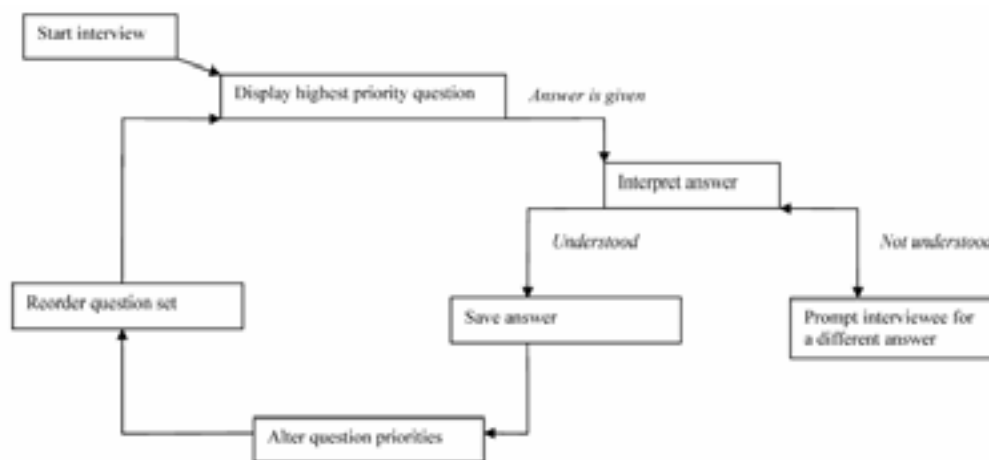


Figura 1 – Processo de ordenação dinâmica das questões de [23]

- Tipos de questão: as questões podem estar numa variedade de formatos. As de múltipla escolha devem ser utilizadas para questões fechadas e as de texto livre devem ser utilizadas para questões abertas. Também poderia existir a facilidade de utilizar *check-lists* e listas *drop-down* conforme apropriado, de modo a tornar o processo de entrevistas o mais simples possível para o entrevistado.

- Facilidade para rever as respostas: os entrevistados deveriam ser capazes de rever todas as respostas que eles tivessem dado durante qualquer estágio da entrevista.

- Confirmação das respostas: deveria ser solicitado que os entrevistados confirmassem suas respostas às questões de texto livre. Isso poderia permitir que os entrevistados checassem se eles marcaram as respostas da forma que eles realmente pretendiam.

- Sem facilidade para alterar as respostas: os entrevistados não deveriam ser capazes de alterar suas respostas uma vez que elas tivessem sido salvas na base de dados. Isso é para imitar as entrevistas face-a-face onde todas as respostas dadas serão lembradas pelo entrevistador. Uma questão pode ser apresentada mais de uma vez durante o processo de entrevista, podendo ser dadas respostas diferentes para a mesma pergunta, mas todas as respostas devem ser gravadas na base de dados.

- Facilidade para adicionar comentários: os entrevistados deveriam ser capazes de adicionar comentários a qualquer resposta dada, no momento em que eles estiverem respondendo ou quando estiverem revendo as respostas dadas anteriormente. Isso permite que expandam, expliquem ou justifiquem suas respostas sempre que sentirem necessidade.

Foi decidido também que o processo de entrevista deveria ser extremamente fácil de ser realizado por usuários de computador experientes e inexperientes. A ferramenta deveria estar altamente acessível, então usabilidade e acessibilidade foram consideradas importantes durante todo o processo de desenho. Foi considerado importante implementar um sistema que se permitia um processo interativo com o entrevistado. Essa interação entre os entrevistados e os entrevistadores é realizada perguntando aos entrevistados certas questões e depois imediatamente perguntando questões diretamente relacionadas às respostas que foram dadas anteriormente, conforme apropriado. Outra técnica de entrevista utilizada foi a de perguntar apenas questões relevantes aos entrevistados, já que os mesmos tendem a responder mais favoravelmente se sentirem que cada questão é importante e significativa.

As questões incluídas na entrevista e suas ordens correspondentes são determinadas pelos autores da entrevista. Conforme as questões iam sendo cadastradas na base de dados, foram sendo associadas prioridades a elas, o que determinava a ordem em que elas iriam ser apresentadas aos entrevistados. Certas regras foram sendo utilizadas para alterar as prioridades das questões conforme necessário, durante o processo atual de entrevistas. Foi desenvolvido um sistema *web*, de forma a atender o requisito de que o sistema deveria estar disponível a usuários em uma variedade de localizações geográficas.

O segundo artigo [32], discute as causas raízes da dificuldade de se capturar requisitos arquiteturais significantes, sugerindo então uma abordagem sistemática para a captura desses tipos de requisito, que são extremamente importantes, de forma a garantir que eles não sejam negligenciados. Um requisito arquitetural é qualquer requisito arquiteturalmente significativo, onde o nível de significância pode estar implícito ou explícito. Requisitos arquiteturais implícitos são aqueles que possuem atributos particulares. Por exemplo, qualquer requisito de alto risco, alta prioridade ou baixa estabilidade pode ser considerado arquiteturalmente

significante. Entretanto o artigo foca primeiramente nos requisitos explícitos, os quais são freqüentemente técnicos por natureza.

Alguns exemplos de requisitos arquiteturais explícitos:

- Suporte a múltiplas linguagens;
- O banco utilizado será o Oracle 8i;
- O sistema rodará sete dias por semana, vinte e quatro horas por dia;
- Será requerido um *help on-line*;
- Toda a lógica de apresentação será escrita em Visual Basic.

O método propõe uma abordagem sistemática que proveja um *framework* para classificar requisitos arquiteturais, de forma a garantir também que eles não sejam negligenciados.

Nesse artigo [32] um sistema para a classificação de requisitos, o FURPS+ [32], foi desenvolvido por Robert Grady da empresa Hewlett-Packard. O acrônimo FURPS+ representa:

- Funcionalidade;
- Usabilidade;
- Confiabilidade;
- Performance;
- Suportabilidade.

O “+” em FURPS+ nos ajuda a lembrar interesses como:

- Requisitos de desenho;
- Requisitos de implementação;
- Requisitos de interface;
- Requisitos físicos.

A sigla “URPS” restante descreve requisitos não funcionais que são geralmente arquiteturalmente relevantes:

- Usabilidade: interesse nas características como estética e consistência na interface do usuário;

- **Confiabilidade:** interesse nas características como disponibilidade, acurácia nos cálculos realizados pelo sistema, e habilidade do sistema se recuperar no caso de falhas;
- **Desempenho:** interesse nas características como transferência de dados, tempo de resposta, tempo para recuperação, tempo para inicialização.
- **Suportabilidade:** interesse nas características como testabilidade, adaptabilidade, gerenciabilidade, compatibilidade, configurabilidade, instalabilidade, escalabilidade e locabilidade.

O “+” no acrônimo FURPS+ é utilizado para identificar categorias que geralmente representam restrições. De modo simplificado, mecanismos arquiteturais representam uma solução comum para um problema encontrado com frequência. Mecanismos arquiteturais são frequentemente utilizados para realizar requisitos arquiteturais. A tabela abaixo, obtida em FURPS [32], mostra três categorias de mecanismos arquiteturais e como mecanismos arquiteturais são expressos em cada uma das três categorias:

<b>Mecanismo de análise</b>	<b>Mecanismo de desenho</b>	<b>Mecanismo de implementação</b>
Persistência	RDBMS OODBMS	Oracle Ingres ObjectStore
Comunicação	Object Request Broker Message queue	Orbix, VisiBroker MSMQ, MQSeries

Um mecanismo de análise representa uma implementação independente de solução. Um mecanismo de desenho é um refinamento do mecanismo de análise. Ele assume alguns detalhes do ambiente de implementação, mas não faz parte de uma implementação específica. Finalmente um mecanismo de implementação é um refinamento do mecanismo de desenho e especifica a implementação exata do mecanismo.

A figura abaixo resume os relacionamentos entre requisitos e mecanismos mostrando refinamentos dos requisitos do FURPS em diferentes níveis de refinamento:

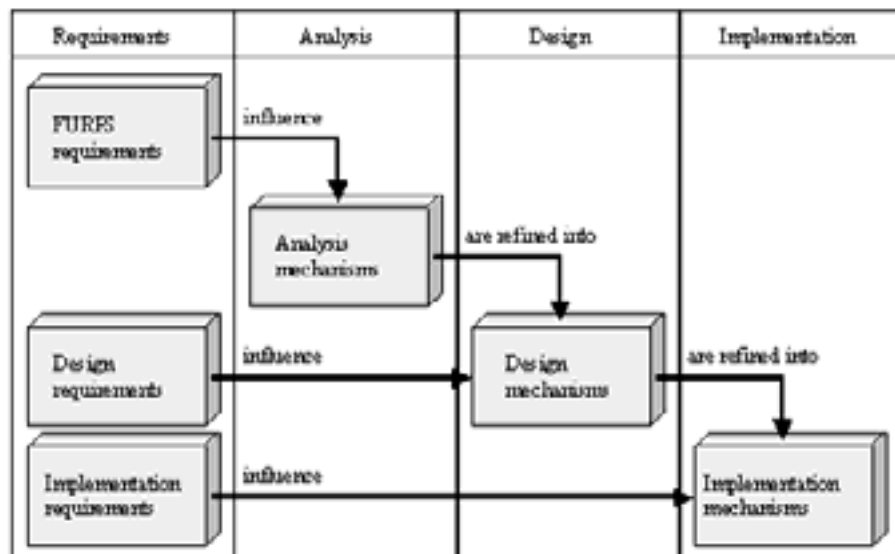


Figura 2 – Relacionamentos entre requisitos e mecanismos [32]

A abordagem proposta para obter requisitos arquiteturais com FURPS foi a seguinte, de acordo com [32]:

1. “Mantenha uma lista completa de requisitos arquiteturais, sem levar em consideração se os itens listados são relevantes ou não para um projeto particular”;
2. “Para cada requisito arquitetural, formule uma ou mais questões que possam ajudar no processo de especificação. Tenha certeza de que todos os envolvidos no projeto possam entender essas questões”;
3. “Ajude os envolvidos no projeto mostrando a eles o impacto potencial de responder uma questão de uma forma ou de outra”;
4. “Capture as respostas dadas a cada uma das questões”;
5. “Ajude o arquiteto assegurando que os envolvidos no projeto (em adição às respostas das questões) atribuam uma prioridade ou um peso a cada requisito arquitetural. Esse peso atribuído ajudará o arquiteto a fazer escolhas entre os requisitos”.



A tabela abaixo exemplifica um questionário desse tipo, questionário esse proposto em FURPS [32], que inclui também exemplos de respostas:

<b>Requisito</b>	<b>Questões</b>	<b>Impacto</b>	<b>Respostas</b>	<b>Prioridade</b>
<b>Licenças</b>	O sistema, ou parte dele será licenciado?  Existem muitas restrições no mecanismo utilizado para prover capacidade de licenciamento?	Quanto maior a sofisticação do mecanismo de licenciamento, maior o tempo para realização da comercialização e maior o custo de manutenção.	O módulo de controle de estoque será comercializado como um componente separado do sistema e irá requerer licença. A ferramenta FlexLM é utilizada por toda nossa organização para prover capacidade de licenciamento.	Média
<b>Disponibilidade</b>	Existe algum requisito a respeito do tempo médio entre falhas do sistema (MTBF)?	Quanto maior a disponibilidade, maior o grau de comercialização do sistema.	Disponibilidade é uma característica chave do produto. O produto deve ter um MTBF de 60 dias	Alta
	Que plataformas	O desenvolvimento	O produto deve	Alta

<b>Suporte a plataformas</b>	o sistema deve suportar?	para uma única plataforma pode reduzir o potencial de comercialização do produto. Também pode permitir uma integração com as características das plataformas.	ser implementado para rodar nas seguintes plataformas UNIX: Sun Solaris IBM AIX HPUX	
------------------------------	--------------------------	---	--	--

Já a dissertação da Ana Paula Gilvaz [5] propõe uma estratégia na qual o Engenheiro de Software conta com a ajuda de um assistente automatizado para a condução de entrevistas. Essa estratégia instancia um modelo conceitual durante o processo de entrevistas, modelo conceitual esse que representa os elementos do sistema objeto, segundo o enfoque dado por três métodos para a determinação de requisitos de informação: BSP (Business System Planning), CSF (Critical Success Factors) e E/M (End-Means Analysis). Esses métodos utilizam uma estratégia para a determinação dos requisitos do sistema de informação, tomando como base a análise das características do sistema envolvente ao sistema de informações, que é a Organização.

Após a entrevista, o Engenheiro de Software dispõe de uma base de conhecimento organizada segundo o modelo conceitual proposto. O Engenheiro de Software interage com o assistente e o cliente, conduzindo o processo de entrevista. No decorrer da entrevista uma ferramenta (FAES), que consiste num assistente para apoiar a entrevista, sugere a agenda de perguntas, fazendo críticas com base em heurísticas e informações armazenadas na base de conhecimento. O cliente vai fornecendo as respostas que vão sendo alimentadas na ferramenta pelo Engenheiro de Software.

A ferramenta é composta de quatro componentes:

- Controle: trata da interação com o Engenheiro de Software e controla os mecanismos para perguntar e aplicar as heurísticas;
- Agenda de perguntas: baseia-se no modelo conceitual e contém as perguntas responsáveis pela instanciação desse modelo;

- Base de conhecimento: armazena as respostas fornecidas pelo cliente, as críticas feitas pela ferramenta, e as entradas feitas pelo Engenheiro de Software;
- Heurísticas: foram derivadas do modelo conceitual e do seu uso em experimentos pilotos.

Nessa dissertação [5], foi adotado um padrão de questionário utilizado para instanciar o modelo que utiliza o conceito de unificação de variáveis, isto é, cada pergunta é composta de uma parte fixa mais uma parte variável que corresponde à resposta de uma outra pergunta já respondida. Cada pergunta tem o objetivo de instanciar um nodo no meta-modelo. A relação entre duas perguntas é definida por cada arco que liga um par de nodos do modelo. No decorrer da entrevista, os nodos vão sendo instanciados com cada uma das respostas para cada pergunta feita e os arcos vão ligando cada par de instâncias de respostas que se relacionam.



Figura 3 – Modelo conceitual de entrevista [5]

A estratégia apresentada na dissertação da Ana Paula Gilvaz [5] propõe cinco tipos de perguntas:

1. Perguntas de instanciação: se referem às perguntas responsáveis pelo preenchimento do modelo;

2. Perguntas de relação: foram derivadas a partir de um conjunto de heurísticas de relação. Tais heurísticas têm o objetivo de descobrir novas relações não previamente definidas entre os elementos do modelo;

3. Perguntas de complementação: foram derivadas a partir de algumas heurísticas de complementação. Seu objetivo é verificar se uma informação associada a algum elemento do modelo é importante para algum outro elemento do modelo que possua uma ligação direta com o elemento informação;

4. Perguntas de inconsistência: tem o propósito de alertar sobre inconsistências ocorridas a respeito de uma resposta;

5. Perguntas de investigação: questionam sobre a existência de informações que podem ter sido esquecidas de serem mencionadas, permitindo que sejam acrescentadas no modelo novas instâncias.

Após a realização da entrevista, temos uma base de conhecimento onde as informações armazenadas estão organizadas segundo o modelo conceitual. O objetivo é que esse modelo forneça os subsídios para que o Engenheiro de Software possa iniciar o processo de definição de requisitos. O modelo documenta a entrevista de uma forma estruturada, disponibilizando ao Engenheiro de Software as respostas da entrevista organizadas segundo o papel de cada uma no sistema objeto, permitindo que sejam feitas vários tipos de consultas estruturadas para extrair do modelo a mais variada gama de informações.

O uso do modelo permitiu a representação de componentes do sistema objeto e da relação entre eles, introduzindo uma clareza maior ao problema, além de possibilitar a definição de um conjunto de heurísticas que criticam e validam as informações armazenadas no modelo. Essas heurísticas foram derivadas do modelo conceitual e procuram por relações que possam fazer algum sentido. As heurísticas utilizadas se classificam em:

- Heurísticas de relação: estão baseadas na existência de termos-comuns entre as instâncias do modelo pertencentes a classes conceituais distintas;
- Heurísticas de complementação: são aplicadas quando as heurísticas de relação são válidas;
- Heurísticas de validação: ocorrem quando as heurísticas de relação não são válidas e caracterizam regras de condição de validade de alguns componentes do modelo;

- Heurísticas de consistência: verificam a consistência de um componente do modelo.

O assistente atua em dois momentos: em tempo da entrevista, fornecendo uma estratégia para o preenchimento do modelo conceitual e identificando relações não pré-definidas; e num segundo momento (pós-entrevista) disponibilizando o modelo preenchido, que pode ser manipulado para a extração de relatórios e estatísticas.

Como pôde ser verificado, os artigos [23] e [32] não falam especificamente da utilização da técnica de questionários para a eliciação de requisitos, sobretudo requisitos de software. Tanto em [23] quanto em [32], fica a cargo do próprio Engenheiro de Software formular as perguntas que serão utilizadas nos questionários, assim como elaborá-los, não existindo um método que o apoie nessa tarefa.

Já a dissertação da Ana Paula Gilvaz [5] utiliza a técnica de questionários para a eliciação de requisitos, porém, ela é aplicada como um apoio à técnica de entrevistas. As perguntas do questionário são geradas pelo assistente automatizado proposto, com objetivo de conduzir as entrevistas realizadas pelo Engenheiro de Software.

Esses dois resumos dos artigos [23] e [32] anteriormente detalhados, assim como a dissertação da Ana Paula Gilvaz [5], foram os que acreditamos terem sido os mais relevantes, após as pesquisas que foram realizadas na área de Engenharia de Software e mais especificamente na área de Engenharia de Requisitos. Eles ajudaram a servir de base para o processo de pesquisa e estudo do estado da arte do assunto.

## 2.2.

### A técnica GQM

Diversos atributos dos produtos, projetos e processos de *software* podem ser medidos. No entanto, deve-se selecionar um conjunto de métricas pequeno e equilibrado, que irá ajudar a organização a acompanhar o progresso na direção de seus objetivos. Segundo Solingen & Berghout [14], a técnica GQM – *Goal Question Metric* (Objetivo/Pergunta/Métrica) é excelente para selecionar as métricas apropriadas aos seus objetivos. A técnica é iniciada selecionando-se alguns objetivos do processo ou da organização. Declaram-se os objetivos de modo que sejam quantificáveis e mensuráveis. Acreditamos que [14] fornece uma boa explicação sobre GQM e resumimos abaixo seus pontos principais.

Para cada objetivo, identificam-se as perguntas que teriam de ser respondidas, para saber se o objetivo está sendo alcançado. Se o seu objetivo era “reduzir os custos de manutenção em 50% no prazo de um ano”, algumas perguntas apropriadas poderiam ser: Quanto gasta em manutenção a cada mês? Qual a parcela dos custos de manutenção que gastamos em cada aplicativo suportado?

Finalmente, identificam-se métricas que ajudarão a responder cada pergunta. De acordo com Solingen & Berghout [14], algumas delas serão simples itens de dados, contadas diretamente, tais como valor total gasto em manutenção. Outras métricas serão calculadas a partir de um ou mais itens de dados. Para responder a última pergunta anterior, deve-se saber o número de horas gasto em cada um dos três tipos de manutenção citados, bem como o custo total de manutenção em um dado período. As métricas e sua interpretação refletem os valores e os pontos de vista dos diferentes grupos afetados: colaboradores, usuários e operadores. Conforme descrito acima, podemos concluir então que a técnica GQM define certos objetivos, refina esses objetivos em questões e define métricas que devem prover informações que respondam às questões. Respondendo às questões, os dados medidos definem os objetivos de modo operacional podendo então ser analisados para identificar se os mesmos foram atingidos ou não.

O primeiro passo de um programa de métricas é estabelecer a *baseline*<sup>1</sup> corrente, de forma que o progresso possa ser avaliado através de comparação com ela e em termos dos objetivos.

O GQM define o modelo de medição em três níveis, segundo Solingen & Berghout [14]:

- **Nível conceitual (Objetivo):** um objetivo é definido para um objeto, para uma variedade de razões, com respeito aos vários modelos da qualidade, dos vários pontos de vista e relativo a um ambiente particular. O conjunto de objetivos é baseado nas necessidades da organização e seus projetos. O objetivo determina o que deve ser melhorado ou aprendido. Segundo [14], o processo de definição de objetivos é suportado por *templates* como o descrito abaixo. Utilizando esses *templates* é possível definir objetivos em termos de finalidade, perspectiva e ambiente. A identificação de sub-objetivos, entidades e atributos relacionados aos sub-objetivos é feita nesse passo.

**Finalidade:** analisar alguns objetos (processos, produtos) com a finalidade de melhorar, avaliar, motivar, caracterizar, prever.

**Perspectiva:** com enfoque em custo, corretude, remoção de defeitos, mudanças, segurança, do ponto de vista do usuário, cliente, gerente, desenvolvedor, corporação.

**Ambiente:** nos contextos de fatores de problemas, pessoais, recursos, processos.

- **Nível operacional (Questão):** um conjunto de questões quantificáveis são utilizadas para definir modelos do objeto de estudo e depois focar nesse objeto para caracterizar a avaliação ou a realização de um objetivo específico. Objetivos de negócio são traduzidos em questões com foco na medição. As mesmas questões podem ser definidas para suportar interpretações de dados de múltiplos objetivos;

- **Nível quantitativo (Métrica):** um conjunto de métricas, baseadas em modelos, são associadas com cada pergunta com objetivo de responder-lhe de uma maneira mensurável. Nesse passo, as métricas identificadas devem ser adequadas para prover informações que respondam às questões. Geralmente cada métrica pode fornecer informações que respondam a várias questões e algumas

<sup>1</sup> Baseline: "Conjunto de artefatos aceitos e controlados que serão utilizados em atividades

vezes são necessárias combinações de métricas para que uma questão seja respondida.

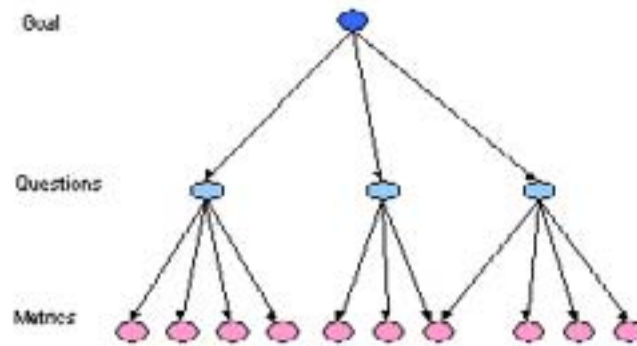


Figura 4 – A técnica GQM [14]

Segundo Solingen & Berghout [14], uma vez que esses passos foram identificados, dados são coletados e interpretados para produzir respostas às questões quantificáveis que foram definidas para completar os objetivos.