LISTA DE VERIFICAÇÃO GERAL DOS ARTEFATOS DE IHC

Inspeção de artefatos pelo método de Fagan

Universidade: Universidade de Brasília (UnB) Faculdade: Faculdade UnB Gama (FGA) Disciplina: Interação Humano-Computador (IHC) Professor: Prof. Dr. André Barros de Sales Grupo Nro: 2 Data: 21/11/2026	**Integrantes e matrículas:** Felipe Lopes Pedroza (231026330) Matheus do Vale Lameira (202070064) Victor Andreozzi De La Rocque Couto (221029285)

Esta lista de verificação serve para inspecionar os artefatos de qualquer projeto de Interação Humano-Computador: perfil de usuário e coleta de dados, personas, cenários, análise hierárquica de tarefas (HTA), árvore de tarefas concorrentes (CTT), storyboard, protótipo de papel, planejamento da avaliação e relato dos resultados. Os itens foram reunidos das listas de verificação que cada integrante elaborou ao longo da disciplina, com a mesma quantidade de itens por integrante: 35 (seção 3). Cada item traz a fonte, com página e capítulo, e a foto do trecho do livro, reaproveitadas da lista de origem, e um critério do método de inspeção de Fagan: o tipo de defeito que a não conformidade representa, com a fonte e a foto do trecho do artigo. As perguntas não citam nenhum projeto em particular.

A inspeção segue o processo de Fagan (seção 1), e cada não conformidade é classificada pelo tipo de defeito e pela severidade (seção 2) e registrada na ficha da seção 5.

1. Como usar a lista: o processo de inspeção de Fagan

A inspeção de software é um tipo de revisão que pode ser aplicado a todos os artefatos e que tem um processo de detecção de defeitos rigoroso e bem definido (Kalinowski e Spínola, p. 68). O processo tradicional, criado por Fagan, tem seis atividades (p. 71). A Tabela 1 mostra cada atividade e como esta lista é usada nela.

Atividade	Quem	O que acontece	Uso desta lista
1. Planejamento	Moderador	Define o contexto da inspeção (artefato, autor, técnica de detecção), seleciona os inspetores e distribui o material (Kalinowski e Spínola, p. 71). Confere se o artefato atende aos critérios de entrada (Doolan, p. 175).	Escolhe as seções da lista que correspondem ao artefato inspecionado.
2. Apresentação	Autor	Apresenta o artefato; pode ser omitida se os inspetores já o conhecem (Kalinowski e Spínola, p. 71). Em Doolan, é uma reunião curta de abertura, de 20 a 30 minutos, que define o objetivo da inspeção (p. 175).	Mostra aos inspetores os itens que serão usados.
3. Preparação	Inspetores	Cada inspetor estuda o artefato sozinho e anota as discrepâncias (Kalinowski e Spínola, p. 71), usando os padrões e as listas de verificação indicados pelo moderador (Doolan, p. 176).	Cada inspetor responde aos itens e anota as não conformidades.
4. Reunião	Moderador, autor e inspetores	As discrepâncias são discutidas e classificadas como defeito ou falso positivo; a decisão final é do moderador. A solução não é discutida, e a reunião não passa de duas horas (Kalinowski e Spínola, p. 71).	Cada defeito é registrado na ficha da seção 5, com tipo e severidade (seção 2).
5. Retrabalho	Autor	O autor corrige os defeitos confirmados pelo moderador (Kalinowski e Spínola, p. 71).	A coluna “Corrigido?” da ficha é preenchida.
6. Continuação	Moderador	Analisa o material corrigido, reavalia a qualidade do artefato e decide se haverá nova inspeção (Kalinowski e Spínola, p. 71); confere que cada defeito teve uma ação do autor (Doolan, p. 176).	O moderador marca a decisão final da ficha.

Tabela 1: atividades do processo de inspeção de Fagan e o uso desta lista.

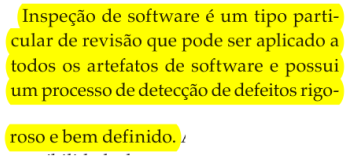
Papel	Responsabilidade
Moderador	Organiza a inspeção, monta a equipe de no máximo cinco pessoas, contando o autor (Doolan, p. 175), conduz a reunião e decide se cada discrepância é defeito ou falso positivo (Kalinowski e Spínola, p. 71).
Autor	Apresenta o artefato, participa da reunião e corrige os defeitos (Kalinowski e Spínola, p. 71).
Inspetor	Estuda o artefato sozinho, responde aos itens da lista e registra as discrepâncias (Kalinowski e Spínola, p. 71).

Tabela 2: papéis da inspeção.

1.1 Fotos dos trechos citados sobre o processo de Fagan

Foto 1: definição de inspeção de software.

Fonte: Página: 68, Introdução. Artigo: MARCOS KALINOWSKI, RODRIGO OLIVEIRA SPÍNOLA, Introdução à Inspeção de Software, Engenharia de Software Magazine, edição 01, p. 68 a 74.



Inspeção de software é um tipo particular de revisão que pode ser aplicado a todos os artefatos de software e possui um processo de detecção de defeitos rigoroso e bem definido.



68 **Engenharia de Software Magazine** – Introdução à Inspeção de Software

Foto 2: as seis atividades do processo de Fagan e os papéis de moderador, autor e inspetores.

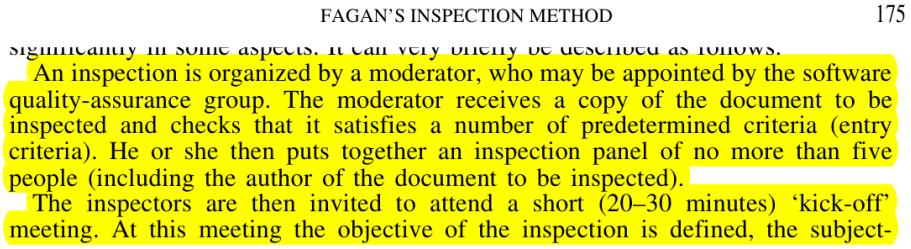
Fonte: Página: 71, O Processo de Inspeção de Software. Artigo: MARCOS KALINOWSKI, RODRIGO OLIVEIRA SPÍNOLA, Introdução à Inspeção de Software, Engenharia de Software Magazine, edição 01, p. 68 a 74.

FAGAN (1976) desenvolveu o processo tradicional de inspeção de software, uma forma detalhada de se realizar uma revisão. Neste processo, existem seis atividades principais:

- Planejamento. Um usuário, desempenhando o papel de moderador da inspeção, define o contexto da inspeção (descrição da inspeção, técnica a ser utilizada na detecção de defeitos, documento a ser inspecionado, autor do documento, entre outros), seleciona os inspetores e distribui o material a ser inspecionado.
- Apresentação. Os autores dos artefatos a serem inspecionados apresentam as características destes. Esta fase pode ser omitida se os inspetores possuem conhecimento sobre o projeto e os artefatos que devem ser inspecionados.
- Preparação. Os inspetores estudam os artefatos individualmente, e eventualmente fazem anotações sobre estes produzindo uma lista de discrepâncias. O fornecimento de técnicas de leitura pode facilitar a execução desta tarefa.
- Reunião. Uma reunião em equipe ocorre, envolvendo o moderador, os inspetores e os autores do documento. Discrepâncias são discutidas, e classificadas como defeito ou falso positivos. A decisão final sobre a classificação de uma discrepância sendo discutida é do moderador. A solução dos defeitos não é discutida durante a reunião, que não deve exceder duas horas, uma vez que após este tempo a concentração e a capacidade de análise dos inspetores costuma reduzir drasticamente. No caso em que uma reunião precisar de mais de duas horas, é sugerido que o trabalho de inspeção continue no próximo dia.
- Retrabalho. O autor corrige os defeitos encontrados pelos inspetores e confirmados pelo moderador.
- Continuação. O material corrigido pelos autores é repassado para o moderador, que faz uma análise da inspeção como um todo e re-avalia a qualidade do artefato inspecionado. Ele tem a liberdade de decidir se uma nova inspeção deve ocorrer ou não.

Edição 01 – **Engenharia de Software Magazine** **71**

Foto 3: moderador, critérios de entrada, equipe de até cinco pessoas e reunião de abertura de 20 a 30 minutos.
Fonte: Página: 175, Fagan's Inspection: an Overview of the Method. Artigo: E. P. DOOLAN, Experience with Fagan's Inspection Method, Software: Practice and Experience, vol. 22, n. 2, p. 173 a 182, 1992.



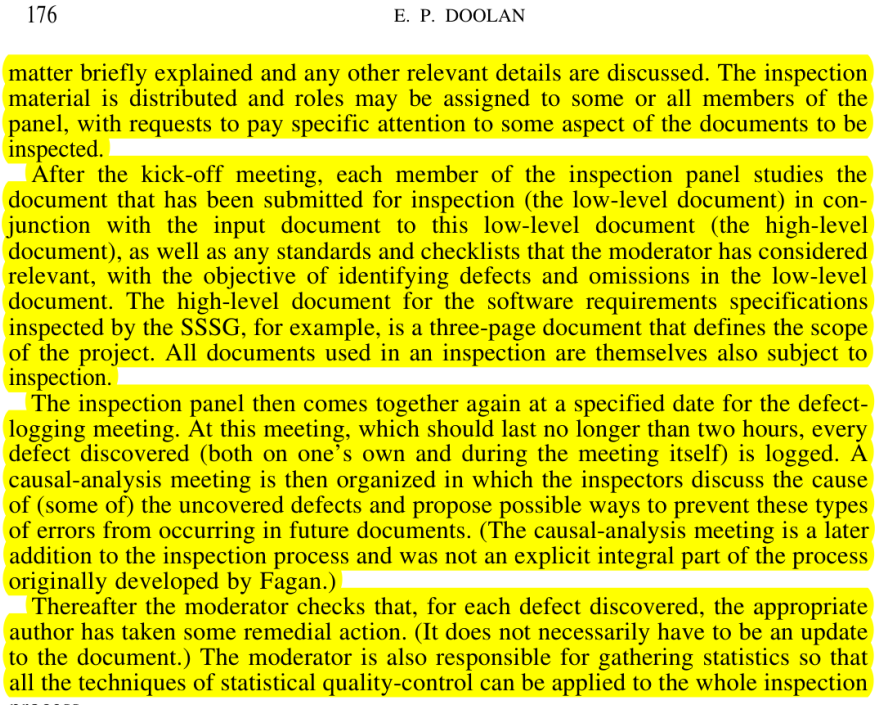
FAGAN'S INSPECTION METHOD 175

significantly in some aspects. It can very briefly be described as follows.

An inspection is organized by a moderator, who may be appointed by the software quality-assurance group. The moderator receives a copy of the document to be inspected and checks that it satisfies a number of predetermined criteria (entry criteria). He or she then puts together an inspection panel of no more than five people (including the author of the document to be inspected).

The inspectors are then invited to attend a short (20–30 minutes) 'kick-off' meeting. At this meeting the objective of the inspection is defined, the subject-

Foto 4: preparação com padrões e listas de verificação, reunião de registro de até duas horas e acompanhamento das correções pelo moderador.
Fonte: Página: 176, Fagan's Inspection: an Overview of the Method. Artigo: E. P. DOOLAN, Experience with Fagan's Inspection Method, Software: Practice and Experience, vol. 22, n. 2, p. 173 a 182, 1992.



176 E. P. DOOLAN

matter briefly explained and any other relevant details are discussed. The inspection material is distributed and roles may be assigned to some or all members of the panel, with requests to pay specific attention to some aspect of the documents to be inspected.

After the kick-off meeting, each member of the inspection panel studies the document that has been submitted for inspection (the low-level document) in conjunction with the input document to this low-level document (the high-level document), as well as any standards and checklists that the moderator has considered relevant, with the objective of identifying defects and omissions in the low-level document. The high-level document for the software requirements specifications inspected by the SSSG, for example, is a three-page document that defines the scope of the project. All documents used in an inspection are themselves also subject to inspection.

The inspection panel then comes together again at a specified date for the defect-logging meeting. At this meeting, which should last no longer than two hours, every defect discovered (both on one's own and during the meeting itself) is logged. A causal-analysis meeting is then organized in which the inspectors discuss the cause of (some of) the uncovered defects and propose possible ways to prevent these types of errors from occurring in future documents. (The causal-analysis meeting is a later addition to the inspection process and was not an explicit integral part of the process originally developed by Fagan.)

Thereafter the moderator checks that, for each defect discovered, the appropriate author has taken some remedial action. (It does not necessarily have to be an update to the document.) The moderator is also responsible for gathering statistics so that all the techniques of statistical quality-control can be applied to the whole inspection

2. Critérios de avaliação de cada item

Cada item é respondido com Conforme, Não conforme ou Não se aplica. Uma resposta Não conforme é uma discrepância, isto é, um suposto defeito, que na reunião pode ser confirmado como defeito ou descartado como falso positivo (Kalinowski e Spínola, p. 69). Cada defeito recebe um tipo (Tabela 3) e uma severidade (Tabela 4).

Tipo de defeito	Descrição (adaptada aos artefatos de IHC)
Omissão	Falta no artefato um elemento importante, uma seção, uma referência a figura ou tabela, ou a definição de um termo.

Tipo de defeito	Descrição (adaptada aos artefatos de IHC)
Ambiguidade	Um trecho admite várias interpretações, por usar termos diferentes para a mesma coisa ou um termo com vários significados.
Inconsistência	Duas ou mais partes do artefato, ou o artefato e outro artefato do projeto, são conflitantes.
Fato incorreto	O artefato descreve algo que não é verdadeiro nas condições do projeto.
Informação estranha	O artefato traz informação desnecessária, que não é usada.
Outros	Outros defeitos, como um conteúdo colocado na seção errada.

Tabela 3: tipos de defeito, a partir da taxonomia de Kalinowski e Spínola (p. 69).

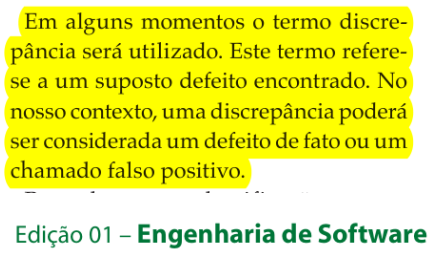
Severidade	Descrição
Menor	Vai de um erro de digitação ou de construção de frase, que pode deixar o sentido aberto a interpretação, a uma dúvida que pede só uma frase de esclarecimento.
Maior	Se o produto fosse construído a partir do artefato sem correção, o defeito geraria um problema relatado depois, como uma funcionalidade errada ou faltando ou uma interface pouco compreensível.
Super-maior	Defeito tão grave que tornaria o produto praticamente inútil, ou exigiria refazer grande parte dele.

Tabela 4: severidade dos defeitos, a partir da classificação de Doolan (p. 178).

2.1 Fotos dos trechos citados sobre os critérios

Foto 5: discrepância: suposto defeito, que pode ser um defeito de fato ou um falso positivo.

Fonte: Página: 69, Definição dos Conceitos. Artigo: MARCOS KALINOWSKI, RODRIGO OLIVEIRA SPÍNOLA, Introdução à Inspeção de Software, Engenharia de Software Magazine, edição 01, p. 68 a 74.

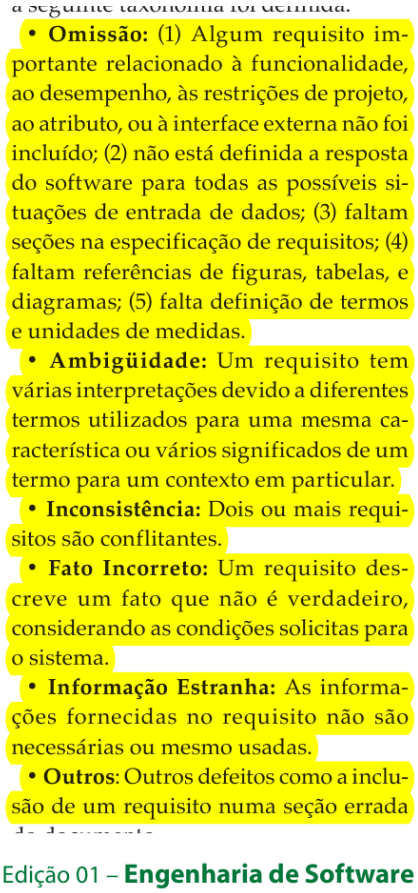


Em alguns momentos o termo discrepância será utilizado. Este termo refere-se a um suposto defeito encontrado. No nosso contexto, uma discrepância poderá ser considerada um defeito de fato ou um chamado falso positivo.

Edição 01 – **Engenharia de Software Magazine** **69**

Foto 6: taxonomia dos tipos de defeito (omissão, ambiguidade, inconsistência, fato incorreto, informação estranha e outros).

Fonte: Página: 69, Definição dos Conceitos. Artigo: MARCOS KALINOWSKI, RODRIGO OLIVEIRA SPÍNOLA, Introdução à Inspeção de Software, Engenharia de Software Magazine, edição 01, p. 68 a 74.



a seguinte taxonomia foi definida:

- **Omissão:** (1) Algum requisito importante relacionado à funcionalidade, ao desempenho, às restrições de projeto, ao atributo, ou à interface externa não foi incluído; (2) não está definida a resposta do software para todas as possíveis situações de entrada de dados; (3) faltam seções na especificação de requisitos; (4) faltam referências de figuras, tabelas, e diagramas; (5) falta definição de termos e unidades de medidas.
- **Ambigüidade:** Um requisito tem várias interpretações devido a diferentes termos utilizados para uma mesma característica ou vários significados de um termo para um contexto em particular.
- **Inconsistência:** Dois ou mais requisitos são conflitantes.
- **Fato Incorreto:** Um requisito descreve um fato que não é verdadeiro, considerando as condições solicitas para o sistema.
- **Informação Estranha:** As informações fornecidas no requisito não são necessárias ou mesmo usadas.
- **Outros**: Outros defeitos como a inclusão de um requisito numa seção errada

do documento

Edição 01 – **Engenharia de Software Magazine** **69**

Foto 7: classificação dos defeitos em menores, maiores e super-maiores.

Fonte: Página: 178, The Classification of Defects. Artigo: E. P. DOOLAN, Experience with Fagan's Inspection Method, Software: Practice and Experience, vol. 22, n. 2, p. 173 a 182, 1992.

THE CLASSIFICATION OF DEFECTS

Currently, we use three classes of defects. These are minor, major and super-major. The categories can be described as follows:

1. *Minor defects.* These range from something like a spelling mistake (which may be truly trivial) through sentence-construction errors (thereby possibly leaving the meaning open to misinterpretation) to clarification issues. These last defects arise when the specification raised some question in an inspector's mind that was left unanswered in the text. Such issues can be major omissions on the author's part (in which case they would be classified as major errors), or they may simply need the addition of a clarifying sentence indicating the 'lie of the land'. In the latter case these errors would be classified as 'minor'.
2. *Major defects.* These can be identified very simply: had the software been produced according to the uncorrected specification, the 'major defect' would have resulted in a fault report. Many of the major defects uncovered in requirements specifications for the software produced by the SSSG have to do with the user interface and the choice of default values. Filling one or two places on the inspection panel with an experienced processor, whose prime function is to check that the proposed user interface is comprehensible, accessible, logical and as user-friendly as possible, guarantees that a number of major errors are always detected in this area. A second source of major errors is incorrect or missing functionality or some vagueness in the specification that would have led to serious misinterpretations.
3. *Super-major errors.* These are errors so serious that they would have made the software virtually useless or, at any rate, would have caused it to deviate from existing standards and/or processing techniques to such an extent that chaos would have ensued. If left undiscovered in the original specification, to repair such an error in the final product would require a complete or, (at the very least) substantial redesign and rewrite. One example of such a problem is a specification requiring an input file that was almost impossible to create in a standard seismic data-processing sequence.

3. Distribuição dos itens por integrante

Integrante	Nº de itens	Conteúdo (artefato e nº de itens)	Itens nesta lista	Lista de origem
Felipe Lopes Pedroza	35	Árvore de tarefas concorrentes (CTT) (10); Storyboard (14); Protótipo de papel e prototipação (3); Planejamento da avaliação (framework DECIDE) (3); Relato dos resultados da avaliação (5)	40 a 63, 66 a 68, 74, 81, 96, 100 a 102, 104 a 105	Lista de Verificação: CTT; Lista de Verificação: Entrega 6 (protótipo de papel, relato e planejamento da avaliação); Lista de Verificação: Storyboard
Matheus do Vale Lameira	35	Perfil de usuário e coleta de dados (12); Storyboard (2); Protótipo de papel e prototipação (1); Planejamento da avaliação (framework DECIDE) (19); Relato dos resultados da avaliação (1)	1 a 12, 64 a 65, 69, 75 a 77, 82 a 94, 97 a 99, 103	Lista de Verificação: Perfil do Usuário; Lista de Verificação: Storyboard
Victor Andreozzi De La Rocque Couto	35	Personas (9); Cenários (10); Análise hierárquica de tarefas (HTA) (8); Protótipo de papel e prototipação (4); Planejamento da avaliação (framework DECIDE) (4)	13 a 39, 70 a 73, 78 a 80, 95	Lista de Verificação: Cenários; Lista de Verificação: HTA; Lista de Verificação: Personas; Lista de Verificação: Planejamento da Avaliação (DECIDE); Lista de Verificação: Protótipo de Papel
Total	105	9 seções de artefatos	1 a 105	10 listas de origem

Tabela 5: quantidade de itens e conteúdo por integrante.

Seção da lista (artefato)	Itens	Nº	Integrante
Perfil de usuário e coleta de dados	1 a 12	12	Matheus
Personas	13 a 21	9	Victor
Cenários	22 a 31	10	Victor
Análise hierárquica de tarefas (HTA)	32 a 39	8	Victor
Árvore de tarefas concorrentes (CTT)	40 a 49	10	Felipe
Storyboard	50 a 65	16	Felipe (14), Matheus (2)
Protótipo de papel e prototipação	66 a 73	8	Felipe (3), Matheus (1), Victor (4)
Planejamento da avaliação (framework DECIDE)	74 a 99	26	Felipe (3), Matheus (19), Victor (4)
Relato dos resultados da avaliação	100 a 105	6	Felipe (5), Matheus (1)

Tabela 6: seções da lista, por artefato.

4. Lista de verificação

Para cada item: marque a resposta; se for Não conforme, o critério de Fagan do item indica o tipo de defeito (Tabela 3); marque a severidade (Tabela 4) e registre o defeito na ficha da seção 5.

4.1 Perfil de usuário e coleta de dados

Item 1: O perfil de usuário especifica o cargo ou a ocupação dos usuários e o papel que desempenham na organização, com a experiência no cargo e as responsabilidades?

Fonte: Página: 134 e 135, Capítulo 5. Identificação de Necessidades dos Usuários e Requisitos de IHC (Seção 5.2). Livro: SIMONE DINIZ JUNQUEIRA BARBOSA, BRUNO SANTANA DA SILVA, Interação Humano-Computador, 1ª ed., Editora Campus/Elsevier, 2010.

Foto:

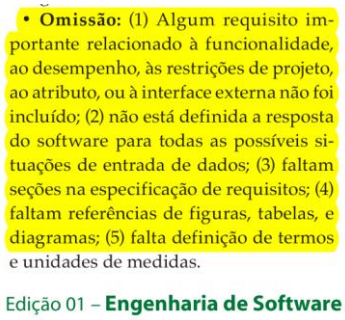
- *experiência no cargo que ocupa*: cargo atual, experiência nesse cargo, tempo na empresa, responsabilidades, trabalhos e cargos anteriores, plano de carreira;

Resposta: ( ) Conforme ( ) Não conforme ( ) Não se aplica

Critério de Fagan: se a resposta for Não conforme, o defeito é do tipo Omissão: falta no artefato um elemento importante que o item exige.

Fonte: Página: 69, Definição dos Conceitos (taxonomia de defeitos). Artigo: MARCOS KALINOWSKI, RODRIGO OLIVEIRA SPÍNOLA, Introdução à Inspeção de Software, Engenharia de Software Magazine, edição 01, p. 68 a 74.

Foto:



• **Omissão:** (1) Algum requisito importante relacionado à funcionalidade, ao desempenho, às restrições de projeto, ao atributo, ou à interface externa não foi incluído; (2) não está definida a resposta do software para todas as possíveis situações de entrada de dados; (3) faltam seções na especificação de requisitos; (4) faltam referências de figuras, tabelas, e diagramas; (5) falta definição de termos e unidades de medidas.

Edição 01 – **Engenharia de Software Magazine** **69**

Severidade (Doolan, p. 178): ( ) Menor ( ) Maior ( ) Super-maior

Origem: Matheus do Vale Lameira, Lista de Verificação: Perfil do Usuário, item 10.

Item 2: O perfil de usuário registra os idiomas que os usuários conhecem e os jargões ou o vocabulário próprio da sua profissão, atividade ou grupo social?

Fonte: Página: 135, Capítulo 5. Identificação de Necessidades dos Usuários e Requisitos de IHC (Seção 5.2). Livro: SIMONE DINIZ JUNQUEIRA BARBOSA, BRUNO SANTANA DA SILVA, Interação Humano-Computador, 1ª ed., Editora Campus/Elsevier, 2010.

Foto:

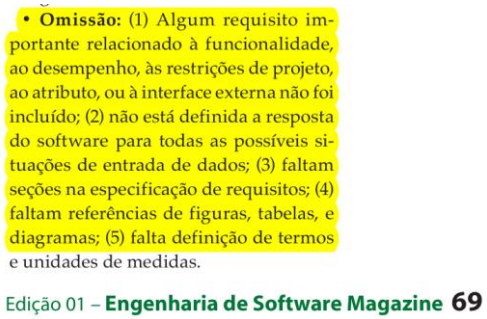
> *idiomas e jargões*: que idiomas o usuário conhece e utiliza fluentemente? Ele possui um jargão profissional particular, um vocabulário próprio da empresa, da sua atividade ou de algum grupo social relevante para o seu projeto?

Resposta: ( ) Conforme ( ) Não conforme ( ) Não se aplica

Critério de Fagan: se a resposta for Não conforme, o defeito é do tipo Omissão: falta no artefato um elemento importante que o item exige.

Fonte: Página: 69, Definição dos Conceitos (taxonomia de defeitos). Artigo: MARCOS KALINOWSKI, RODRIGO OLIVEIRA SPÍNOLA, Introdução à Inspeção de Software, Engenharia de Software Magazine, edição 01, p. 68 a 74.

Foto:



• **Omissão:** (1) Algum requisito importante relacionado à funcionalidade, ao desempenho, às restrições de projeto, ao atributo, ou à interface externa não foi incluído; (2) não está definida a resposta do software para todas as possíveis situações de entrada de dados; (3) faltam seções na especificação de requisitos; (4) faltam referências de figuras, tabelas, e diagramas; (5) falta definição de termos e unidades de medidas.

Edição 01 – **Engenharia de Software Magazine** **69**

Severidade (Doolan, p. 178): ( ) Menor ( ) Maior ( ) Super-maior

Origem: Matheus do Vale Lameira, Lista de Verificação: Perfil do Usuário, item 14.

Item 3: O perfil de usuário indica o quanto o usuário valoriza treinamento, o seu estilo de aprendizado preferido e o tempo que pode investir para aprender a usar o produto?

Fonte: Página: 135, Capítulo 5. Identificação de Necessidades dos Usuários e Requisitos de IHC (Seção 5.2). Livro: SIMONE DINIZ JUNQUEIRA BARBOSA, BRUNO SANTANA DA SILVA, Interação Humano-Computador, 1ª ed., Editora Campus/Elsevier, 2010.

Foto:

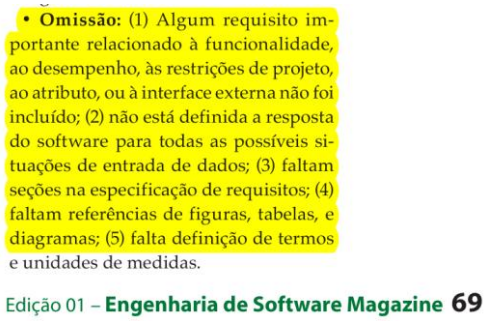
> *treinamento*: o quanto o usuário valoriza treinamento? Prefere um estilo de aprendizado visual, auditivo ou outro? Pode investir tempo aprendendo a utilizar o produto em questão?;

Resposta: ( ) Conforme ( ) Não conforme ( ) Não se aplica

Critério de Fagan: se a resposta for Não conforme, o defeito é do tipo Omissão: falta no artefato um elemento importante que o item exige.

Fonte: Página: 69, Definição dos Conceitos (taxonomia de defeitos). Artigo: MARCOS KALINOWSKI, RODRIGO OLIVEIRA SPÍNOLA, Introdução à Inspeção de Software, Engenharia de Software Magazine, edição 01, p. 68 a 74.

Foto:



• **Omissão:** (1) Algum requisito importante relacionado à funcionalidade, ao desempenho, às restrições de projeto, ao atributo, ou à interface externa não foi incluído; (2) não está definida a resposta do software para todas as possíveis situações de entrada de dados; (3) faltam seções na especificação de requisitos; (4) faltam referências de figuras, tabelas, e diagramas; (5) falta definição de termos e unidades de medidas.

Edição 01 – **Engenharia de Software Magazine** **69**

Severidade (Doolan, p. 178): ( ) Menor ( ) Maior ( ) Super-maior

Origem: Matheus do Vale Lameira, Lista de Verificação: Perfil do Usuário, item 15.

Item 4: O plano de coleta de dados distingue os usuários primários (que usam o produto regularmente), os secundários (que o usam ocasionalmente) e os demais interessados afetados pelo sistema?

Fonte: Página: 136, Capítulo 5. Identificação de Necessidades dos Usuários e Requisitos de IHC (Seção 5.3). Livro: SIMONE DINIZ JUNQUEIRA BARBOSA, BRUNO SANTANA DA SILVA, Interação Humano-Computador, 1ª ed., Editora Campus/Elsevier, 2010.

Foto:

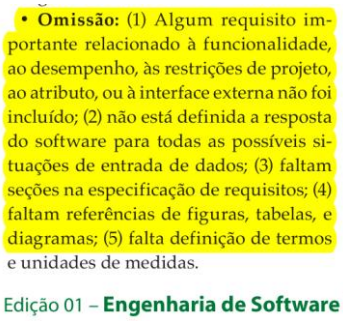
O termo "usuário" geralmente diz respeito aos usuários finais, aqueles que são ou serão usuários diretos do seu produto, sejam **primários**, que utilizam o produto regularmente, ou **secundários**, que o utilizam ocasionalmente, por exemplo, em atividades de configuração eventuais. Além deles, pode ser importante traçar o perfil de outras partes interessadas (***stakeholders***), que não utilizam o produto diretamente mas são afetados pelo seu uso, como, por exemplo, pessoas que devem receber informações ou artefatos resultantes do uso do produto. Note que o termo *stakeholders* costuma se aplicar a todas as partes interessadas, incluindo os próprios usuários. Além disso, um único *stakeholder* pode exercer diversos papéis num sistema ou organização, os quais podem ter necessidades contraditórias.

Resposta: ( ) Conforme ( ) Não conforme ( ) Não se aplica

Critério de Fagan: se a resposta for Não conforme, o defeito é do tipo Omissão: falta no artefato um elemento importante que o item exige.

Fonte: Página: 69, Definição dos Conceitos (taxonomia de defeitos). Artigo: MARCOS KALINOWSKI, RODRIGO OLIVEIRA SPÍNOLA, Introdução à Inspeção de Software, Engenharia de Software Magazine, edição 01, p. 68 a 74.

Foto:



• **Omissão:** (1) Algum requisito importante relacionado à funcionalidade, ao desempenho, às restrições de projeto, ao atributo, ou à interface externa não foi incluído; (2) não está definida a resposta do software para todas as possíveis situações de entrada de dados; (3) faltam seções na especificação de requisitos; (4) faltam referências de figuras, tabelas, e diagramas; (5) falta definição de termos e unidades de medidas.

Edição 01 – **Engenharia de Software Magazine** **69**

Severidade (Doolan, p. 178): ( ) Menor ( ) Maior ( ) Super-maior

Origem: Matheus do Vale Lameira, Lista de Verificação: Perfil do Usuário, item 16.

Item 5: Os participantes escolhidos para a coleta de dados são fontes confiáveis, relevantes e representativas dos usuários e do seu trabalho?

Fonte: Página: 135 e 136, Capítulo 5. Identificação de Necessidades dos Usuários e Requisitos de IHC (Seção 5.3). Livro: SIMONE DINIZ JUNQUEIRA BARBOSA, BRUNO SANTANA DA SILVA, Interação Humano-Computador, 1ª ed., Editora Campus/Elsevier, 2010.

Foto:

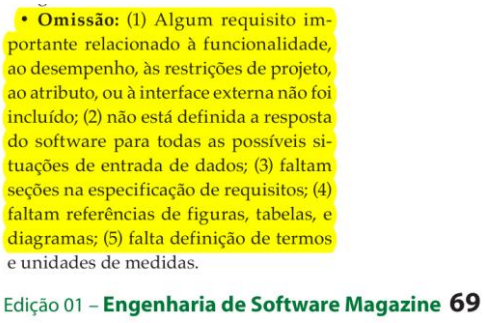
> Um aspecto importante da coleta de dados é definir quem fornecerá qual tipo de informação. Ao coletar dados sobre os usuários do sistema, é essencial encontrar fontes confiáveis, relevantes e representativas dos usuários e do seu trabalho. Caso contrário, não apenas os dados coletados serão de pouca utilidade, como também podem prejudicar seu produto, sua credibilidade e a credibilidade das atividades de IHC em geral.

Resposta: ( ) Conforme ( ) Não conforme ( ) Não se aplica

Critério de Fagan: se a resposta for Não conforme, o defeito é do tipo Omissão: falta no artefato um elemento importante que o item exige.

Fonte: Página: 69, Definição dos Conceitos (taxonomia de defeitos). Artigo: MARCOS KALINOWSKI, RODRIGO OLIVEIRA SPÍNOLA, Introdução à Inspeção de Software, Engenharia de Software Magazine, edição 01, p. 68 a 74.

Foto:



• **Omissão:** (1) Algum requisito importante relacionado à funcionalidade, ao desempenho, às restrições de projeto, ao atributo, ou à interface externa não foi incluído; (2) não está definida a resposta do software para todas as possíveis situações de entrada de dados; (3) faltam seções na especificação de requisitos; (4) faltam referências de figuras, tabelas, e diagramas; (5) falta definição de termos e unidades de medidas.

Edição 01 – **Engenharia de Software Magazine** **69**

Severidade (Doolan, p. 178): ( ) Menor ( ) Maior ( ) Super-maior

Origem: Matheus do Vale Lameira, Lista de Verificação: Perfil do Usuário, item 17.

Item 6: A seleção de participantes evita conhecidos, amigos, alunos ou subordinados do avaliador, para que essa relação não influencie nem contamine os dados?

Fonte: Página: 308, Capítulo 9. Planejamento da Avaliação de IHC. Livro: SIMONE DINIZ JUNQUEIRA BARBOSA, BRUNO SANTANA DA SILVA, Interação Humano-Computador, 1ª ed., Editora Campus/Elsevier, 2010.

Foto:

Sempre que possível, o avaliador deve evitar selecionar seus conhecidos, amigos, alunos, subordinados ou pessoas com as quais possua alguma relação próxima, para evitar que a relação entre ele e o participante influencie ou contamine os dados coletados. Por exemplo, um amigo e um aluno podem não relatar problemas que perceberam na interface por se sentirem criticando o trabalho do avaliador amigo ou professor. De qualquer modo, o avaliador pode atenuar a interferência das relações entre eles esclarecendo que o objetivo da avaliação é encontrar problemas na interface e que a opinião dos participantes é importante para melhorá-la.

Resposta: ( ) Conforme ( ) Não conforme ( ) Não se aplica

Critério de Fagan: se a resposta for Não conforme, o defeito é do tipo Outros: defeito que não se encaixa nos demais tipos, como um conteúdo na seção errada.

Fonte: Página: 69, Definição dos Conceitos (taxonomia de defeitos). Artigo: MARCOS KALINOWSKI, RODRIGO OLIVEIRA SPÍNOLA, Introdução à Inspeção de Software, Engenharia de Software Magazine, edição 01, p. 68 a 74.

Foto:

• **Outros**: Outros defeitos como a inclusão de um requisito numa seção errada do documento.

Edição 01 – **Engenharia de Software Magazine** **69**

Severidade (Doolan, p. 178): ( ) Menor ( ) Maior ( ) Super-maior

Origem: Matheus do Vale Lameira, Lista de Verificação: Perfil do Usuário, item 20.

Item 7: O plano de coleta de dados garante a confidencialidade dos dados brutos e o anonimato dos participantes nos relatórios, com o uso de códigos ou nomes fictícios?

Fonte: Página: 140, Capítulo 5. Identificação de Necessidades dos Usuários e Requisitos de IHC (Seção 5.4). Livro: SIMONE DINIZ JUNQUEIRA BARBOSA, BRUNO SANTANA DA SILVA, Interação Humano-Computador, 1ª ed., Editora Campus/Elsevier, 2010.

Foto:

O pesquisador deve garantir aos participantes **a confidencialidade e a privacidade** dos dados brutos coletados. Com o consentimento dos participantes, os dados brutos são compartilhados apenas com os pesquisadores. Ninguém mais deve ter acesso a esses dados brutos.

Resposta: ( ) Conforme ( ) Não conforme ( ) Não se aplica

Critério de Fagan: se a resposta for Não conforme, o defeito é do tipo Omissão: falta no artefato um elemento importante que o item exige.

Fonte: Página: 69, Definição dos Conceitos (taxonomia de defeitos). Artigo: MARCOS KALINOWSKI, RODRIGO OLIVEIRA SPÍNOLA, Introdução à Inspeção de Software, Engenharia de Software Magazine, edição 01, p. 68 a 74.

Foto:

• **Omissão:** (1) Algum requisito importante relacionado à funcionalidade, ao desempenho, às restrições de projeto, ao atributo, ou à interface externa não foi incluído; (2) não está definida a resposta do software para todas as possíveis situações de entrada de dados; (3) faltam seções na especificação de requisitos; (4) faltam referências de figuras, tabelas, e diagramas; (5) falta definição de termos e unidades de medidas.

Edição 01 – **Engenharia de Software Magazine** **69**

Severidade (Doolan, p. 178): ( ) Menor ( ) Maior ( ) Super-maior

Origem: Matheus do Vale Lameira, Lista de Verificação: Perfil do Usuário, item 22.

Item 8: O participante é informado de que pode recusar a participação ou abandonar o estudo a qualquer momento, sem ser penalizado?

Fonte: Página: 141, Capítulo 5. Identificação de Necessidades dos Usuários e Requisitos de IHC (Seção 5.4). Livro: SIMONE DINIZ JUNQUEIRA BARBOSA, BRUNO SANTANA DA SILVA, Interação Humano-Computador, 1ª ed., Editora Campus/Elsevier, 2010.

Foto:

O participante tem **o direito e a liberdade de se recusar a participar** ou retirar seu consentimento e abandonar o estudo em qualquer fase da pesquisa, sem ser penalizado por isso. Sempre que o pesquisador perceber que o participante está passando por algum tipo de constrangimento ou incômodo físico, emocional ou psíquico, deve interromper a pesquisa antes que o participante tenha mais sofrimentos desnecessários. Uma pesquisa de IHC não deve deixar os participantes excessivamente exaustos, nervosos, ou levá-los ao pranto. Ela deve ser interrompida bem antes disso.

Resposta: ( ) Conforme ( ) Não conforme ( ) Não se aplica

Critério de Fagan: se a resposta for Não conforme, o defeito é do tipo Omissão: falta no artefato um elemento importante que o item exige.

Fonte: Página: 69, Definição dos Conceitos (taxonomia de defeitos). Artigo: MARCOS KALINOWSKI, RODRIGO OLIVEIRA SPÍNOLA, Introdução à Inspeção de Software, Engenharia de Software Magazine, edição 01, p. 68 a 74.

Foto:

• **Omissão:** (1) Algum requisito importante relacionado à funcionalidade, ao desempenho, às restrições de projeto, ao atributo, ou à interface externa não foi incluído; (2) não está definida a resposta do software para todas as possíveis situações de entrada de dados; (3) faltam seções na especificação de requisitos; (4) faltam referências de figuras, tabelas, e diagramas; (5) falta definição de termos e unidades de medidas.

Edição 01 – **Engenharia de Software Magazine** **69**

Severidade (Doolan, p. 178): ( ) Menor ( ) Maior ( ) Super-maior

Origem: Matheus do Vale Lameira, Lista de Verificação: Perfil do Usuário, item 23.

Item 9: O planejamento da entrevista define o grau de estruturação (estruturada, semiestruturada ou não estruturada) e organiza um roteiro com os tópicos ou as perguntas?

Fonte: Página: 146, Capítulo 5. Identificação de Necessidades dos Usuários e Requisitos de IHC (Seção 5.5.1). Livro: SIMONE DINIZ JUNQUEIRA BARBOSA, BRUNO SANTANA DA SILVA, Interação Humano-Computador, 1ª ed., Editora Campus/Elsevier, 2010.

Foto:

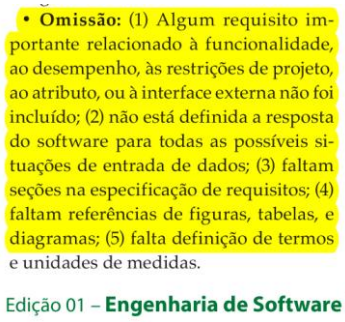
> Em geral, buscamos um meio termo, e conduzimos **entrevistas semiestruturadas**. Nessas entrevistas, o roteiro é composto dos tópicos ou perguntas (geralmente abertas) que devem ser endereçados na entrevista, em uma ordem lógica. O entrevistador tem liberdade para explorar em maior profundidade as respostas fornecidas pelo entrevistado e até mesmo modificar a ordem dos tópicos abordados, mas deve manter o foco nos objetivos da entrevista.

Resposta: ( ) Conforme ( ) Não conforme ( ) Não se aplica

Critério de Fagan: se a resposta for Não conforme, o defeito é do tipo Omissão: falta no artefato um elemento importante que o item exige.

Fonte: Página: 69, Definição dos Conceitos (taxonomia de defeitos). Artigo: MARCOS KALINOWSKI, RODRIGO OLIVEIRA SPÍNOLA, Introdução à Inspeção de Software, Engenharia de Software Magazine, edição 01, p. 68 a 74.

Foto:



• **Omissão:** (1) Algum requisito importante relacionado à funcionalidade, ao desempenho, às restrições de projeto, ao atributo, ou à interface externa não foi incluído; (2) não está definida a resposta do software para todas as possíveis situações de entrada de dados; (3) faltam seções na especificação de requisitos; (4) faltam referências de figuras, tabelas, e diagramas; (5) falta definição de termos e unidades de medidas.

Edição 01 – **Engenharia de Software Magazine** **69**

Severidade (Doolan, p. 178): ( ) Menor ( ) Maior ( ) Super-maior

Origem: Matheus do Vale Lameira, Lista de Verificação: Perfil do Usuário, item 26.

Item 10: O roteiro e a conduta do entrevistador evitam influenciar as respostas com a formulação das perguntas, expressões faciais, gestos ou entonação de voz?

Fonte: Página: 147, Capítulo 5. Identificação de Necessidades dos Usuários e Requisitos de IHC (Seção 5.5.1). Livro: SIMONE DINIZ JUNQUEIRA BARBOSA, BRUNO SANTANA DA SILVA, Interação Humano-Computador, 1ª ed., Editora Campus/Elsevier, 2010.

Foto:

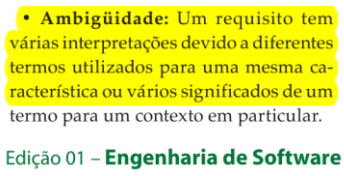
> O entrevistador deve evitar influenciar as respostas dos entrevistados com a formulação das perguntas, expressões faciais, gestos ou entonação de voz. Ao perguntar "Por que você gosta do mecanismo de busca do site CompreMais?", por exemplo, o entrevistador já está pressupondo que o entrevistado gosta desse mecanismo, o que pode não corresponder à realidade. Esse tipo de pergunta pode desmotivar o entrevistado a dar sua opinião sincera e fazer com que ele responda o que acredita que o entrevistador queira ouvir.

Resposta: ( ) Conforme ( ) Não conforme ( ) Não se aplica

Critério de Fagan: se a resposta for Não conforme, o defeito é do tipo Ambiguidade: o artefato admite mais de uma interpretação.

Fonte: Página: 69, Definição dos Conceitos (taxonomia de defeitos). Artigo: MARCOS KALINOWSKI, RODRIGO OLIVEIRA SPÍNOLA, Introdução à Inspeção de Software, Engenharia de Software Magazine, edição 01, p. 68 a 74.

Foto:



• **Ambigüidade:** Um requisito tem várias interpretações devido a diferentes termos utilizados para uma mesma característica ou vários significados de um termo para um contexto em particular.

Edição 01 – **Engenharia de Software Magazine** **69**

Severidade (Doolan, p. 178): ( ) Menor ( ) Maior ( ) Super-maior

Origem: Matheus do Vale Lameira, Lista de Verificação: Perfil do Usuário, item 27.

Item 11: O procedimento pede ao participante permissão para gravar a sua voz ou imagem antes de começar qualquer gravação?

Fonte: Página: 140, Capítulo 5. Identificação de Necessidades dos Usuários e Requisitos de IHC (Seção 5.4). Livro: SIMONE DINIZ JUNQUEIRA BARBOSA, BRUNO SANTANA DA SILVA, Interação Humano-Computador, 1ª ed., Editora Campus/Elsevier, 2010.

Foto:

aos resultados da pesquisa antes que eles sejam publicados.

É necessário obter **permissão para gravar** a voz ou imagem de qualquer pessoa, antes de começar a gravação. Devemos informar aos participantes logo no recrutamento sobre os tipos de gravações que serão realizadas, para evitar mal-entendidos ou desistências no momento da atividade.

Resposta: ( ) Conforme ( ) Não conforme ( ) Não se aplica

Critério de Fagan: se a resposta for Não conforme, o defeito é do tipo Omissão: falta no artefato um elemento importante que o item exige.

Fonte: Página: 69, Definição dos Conceitos (taxonomia de defeitos). Artigo: MARCOS KALINOWSKI, RODRIGO OLIVEIRA SPÍNOLA, Introdução à Inspeção de Software, Engenharia de Software Magazine, edição 01, p. 68 a 74.

Foto:

• **Omissão:** (1) Algum requisito importante relacionado à funcionalidade, ao desempenho, às restrições de projeto, ao atributo, ou à interface externa não foi incluído; (2) não está definida a resposta do software para todas as possíveis situações de entrada de dados; (3) faltam seções na especificação de requisitos; (4) faltam referências de figuras, tabelas, e diagramas; (5) falta definição de termos e unidades de medidas.

Edição 01 – **Engenharia de Software Magazine** **69**

Severidade (Doolan, p. 178): ( ) Menor ( ) Maior ( ) Super-maior

Origem: Matheus do Vale Lameira, Lista de Verificação: Perfil do Usuário, item 29.

Item 12: O planejamento estima quanto tempo o respondente leva para completar o questionário e verifica se esse tempo é adequado?

Fonte: Página: 171, Capítulo 5. Identificação de Necessidades dos Usuários e Requisitos de IHC. Livro: SIMONE DINIZ JUNQUEIRA BARBOSA, BRUNO SANTANA DA SILVA, Interação Humano-Computador, 1ª ed., Editora Campus/Elsevier, 2010.

Foto:

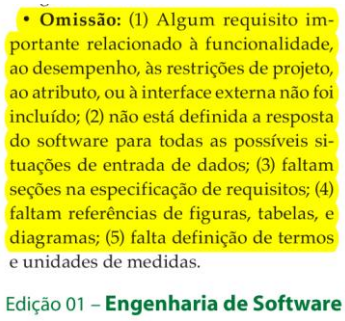
- Quanto tempo você estima que o respondente leve para completar o questionário? Esse tempo é adequado?

Resposta: ( ) Conforme ( ) Não conforme ( ) Não se aplica

Critério de Fagan: se a resposta for Não conforme, o defeito é do tipo Omissão: falta no artefato um elemento importante que o item exige.

Fonte: Página: 69, Definição dos Conceitos (taxonomia de defeitos). Artigo: MARCOS KALINOWSKI, RODRIGO OLIVEIRA SPÍNOLA, Introdução à Inspeção de Software, Engenharia de Software Magazine, edição 01, p. 68 a 74.

Foto:



- **Omissão:** (1) Algum requisito importante relacionado à funcionalidade, ao desempenho, às restrições de projeto, ao atributo, ou à interface externa não foi incluído; (2) não está definida a resposta do software para todas as possíveis situações de entrada de dados; (3) faltam seções na especificação de requisitos; (4) faltam referências de figuras, tabelas, e diagramas; (5) falta definição de termos e unidades de medidas.

Edição 01 – **Engenharia de Software Magazine** **69**

Severidade (Doolan, p. 178): ( ) Menor ( ) Maior ( ) Super-maior

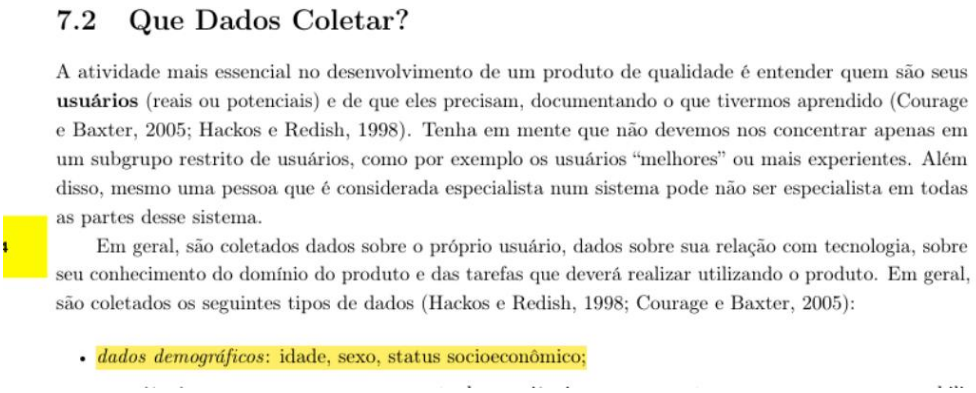
Origem: Matheus do Vale Lameira, Lista de Verificação: Perfil do Usuário, item 33.

4.2 Personas

Item 13: A persona possui dados demográficos definidos, como idade, sexo e status socioeconômico?

Fonte: Página: 137 (PDF disponibilizado no Aprender), Capítulo 7, Seção 7.2. Livro: SIMONE DINIZ JUNQUEIRA BARBOSA, BRUNO SANTANA DA SILVA, Interação Humano-Computador (PDF disponibilizado no Aprender).

Foto:



7.2 Que Dados Coletar?

A atividade mais essencial no desenvolvimento de um produto de qualidade é entender quem são seus **usuários** (reais ou potenciais) e de que eles precisam, documentando o que tivermos aprendido (Courage e Baxter, 2005; Hackos e Redish, 1998). Tenha em mente que não devemos nos concentrar apenas em um subgrupo restrito de usuários, como por exemplo os usuários "melhores" ou mais experientes. Além disso, mesmo uma pessoa que é considerada especialista num sistema pode não ser especialista em todas as partes desse sistema.

Em geral, são coletados dados sobre o próprio usuário, dados sobre sua relação com tecnologia, sobre seu conhecimento do domínio do produto e das tarefas que deverá realizar utilizando o produto. Em geral, são coletados os seguintes tipos de dados (Hackos e Redish, 1998; Courage e Baxter, 2005):

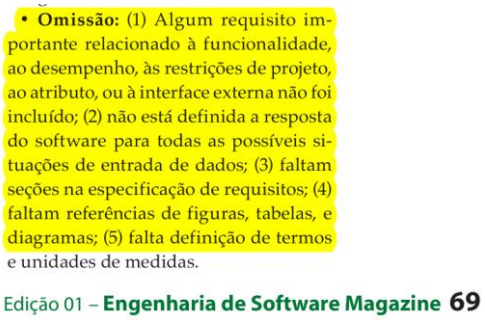
- *dados demográficos*: idade, sexo, status socioeconômico;

Resposta: ( ) Conforme ( ) Não conforme ( ) Não se aplica

Critério de Fagan: se a resposta for Não conforme, o defeito é do tipo Omissão: falta no artefato um elemento importante que o item exige.

Fonte: Página: 69, Definição dos Conceitos (taxonomia de defeitos). Artigo: MARCOS KALINOWSKI, RODRIGO OLIVEIRA SPÍNOLA, Introdução à Inspeção de Software, Engenharia de Software Magazine, edição 01, p. 68 a 74.

Foto:



• **Omissão:** (1) Algum requisito importante relacionado à funcionalidade, ao desempenho, às restrições de projeto, ao atributo, ou à interface externa não foi incluído; (2) não está definida a resposta do software para todas as possíveis situações de entrada de dados; (3) faltam seções na especificação de requisitos; (4) faltam referências de figuras, tabelas, e diagramas; (5) falta definição de termos e unidades de medidas.

Edição 01 – **Engenharia de Software Magazine** **69**

Severidade (Doolan, p. 178): ( ) Menor ( ) Maior ( ) Super-maior

Origem: Victor Andreozzi De La Rocque Couto, Lista de Verificação: Personas, item 1.

Item 14: A persona detalha o nível educacional, o grau de instrução, a área de formação, os cursos realizados e o alfabetismo (o quão bem lê e a disposição para ler textos)?

Fonte: Página: 137 (PDF disponibilizado no Aprender), Capítulo 7, Seção 7.2. Livro: SIMONE DINIZ JUNQUEIRA BARBOSA, BRUNO SANTANA DA SILVA, Interação Humano-Computador (PDF disponibilizado no Aprender).

Foto:

7.2 Que Dados Coletar?

A atividade mais essencial no desenvolvimento de um produto de qualidade é entender quem são seus **usuários** (reais ou potenciais) e de que eles precisam, documentando o que tivermos aprendido (Courage e Baxter, 2005; Hackos e Redish, 1998). Tenha em mente que não devemos nos concentrar apenas em um subgrupo restrito de usuários, como por exemplo os usuários "melhores" ou mais experientes. Além disso, mesmo uma pessoa que é considerada especialista num sistema pode não ser especialista em todas as partes desse sistema.

Em geral, são coletados dados sobre o próprio usuário, dados sobre sua relação com tecnologia, sobre seu conhecimento do domínio do produto e das tarefas que deverá realizar utilizando o produto. Em geral, são coletados os seguintes tipos de dados (Hackos e Redish, 1998; Courage e Baxter, 2005):

- *dados demográficos*: idade, sexo, status socioeconômico;
- *experiência no cargo que ocupa*: cargo atual, experiência nesse cargo, tempo na empresa, responsabilidades, trabalhos e cargos anteriores, plano de carreira;
- *informações sobre a empresa*: tamanho da empresa, área de atuação;
- *educação*: grau de instrução, área de formação, cursos realizados, alfabetismo. O quão bem o usuário lê? Ele tem dificuldade com informação impressa? Tem experiência com textos complexos? Está disposto a ler texto ao utilizar produtos como o que está sendo projetado? Prefere aprender com outras pessoas? Prefere aprender fazendo?;

Resposta: ( ) Conforme ( ) Não conforme ( ) Não se aplica

Critério de Fagan: se a resposta for Não conforme, o defeito é do tipo Omissão: falta no artefato um elemento importante que o item exige.

Fonte: Página: 69, Definição dos Conceitos (taxonomia de defeitos). Artigo: MARCOS KALINOWSKI, RODRIGO OLIVEIRA SPÍNOLA, Introdução à Inspeção de Software, Engenharia de Software Magazine, edição 01, p. 68 a 74.

Foto:

• **Omissão:** (1) Algum requisito importante relacionado à funcionalidade, ao desempenho, às restrições de projeto, ao atributo, ou à interface externa não foi incluído; (2) não está definida a resposta do software para todas as possíveis situações de entrada de dados; (3) faltam seções na especificação de requisitos; (4) faltam referências de figuras, tabelas, e diagramas; (5) falta definição de termos e unidades de medidas.

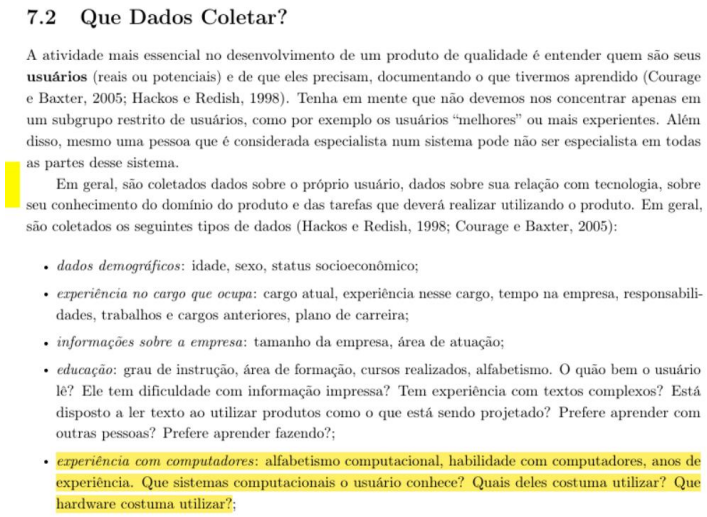
Severidade (Doolan, p. 178): ( ) Menor ( ) Maior ( ) Super-maior

Origem: Victor Andreozzi De La Rocque Couto, Lista de Verificação: Personas, item 4.

Item 15: A persona descreve a experiência com computadores, com o alfabetismo computacional, a habilidade, os anos de experiência e os sistemas e equipamentos que conhece e usa?

Fonte: Página: 137 (PDF disponibilizado no Aprender), Capítulo 7, Seção 7.2. Livro: SIMONE DINIZ JUNQUEIRA BARBOSA, BRUNO SANTANA DA SILVA, Interação Humano-Computador (PDF disponibilizado no Aprender).

Foto:



7.2 Que Dados Coletar?

A atividade mais essencial no desenvolvimento de um produto de qualidade é entender quem são seus **usuários** (reais ou potenciais) e de que eles precisam, documentando o que tivermos aprendido (Courage e Baxter, 2005; Hackos e Redish, 1998). Tenha em mente que não devemos nos concentrar apenas em um subgrupo restrito de usuários, como por exemplo os usuários "melhores" ou mais experientes. Além disso, mesmo uma pessoa que é considerada especialista num sistema pode não ser especialista em todas as partes desse sistema.

Em geral, são coletados dados sobre o próprio usuário, dados sobre sua relação com tecnologia, sobre seu conhecimento do domínio do produto e das tarefas que deverá realizar utilizando o produto. Em geral, são coletados os seguintes tipos de dados (Hackos e Redish, 1998; Courage e Baxter, 2005):

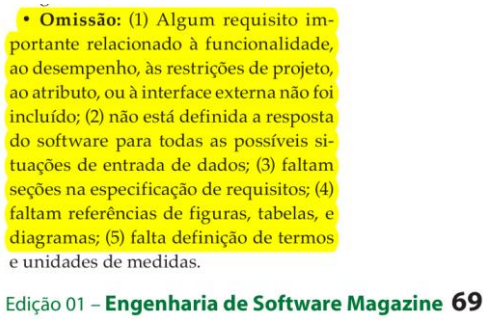
- *dados demográficos*: idade, sexo, status socioeconômico;
- *experiência no cargo que ocupa*: cargo atual, experiência nesse cargo, tempo na empresa, responsabilidades, trabalhos e cargos anteriores, plano de carreira;
- *informações sobre a empresa*: tamanho da empresa, área de atuação;
- *educação*: grau de instrução, área de formação, cursos realizados, alfabetismo. O quão bem o usuário lê? Ele tem dificuldade com informação impressa? Tem experiência com textos complexos? Está disposto a ler texto ao utilizar produtos como o que está sendo projetado? Prefere aprender com outras pessoas? Prefere aprender fazendo?;
- *experiência com computadores*: alfabetismo computacional, habilidade com computadores, anos de experiência. Que sistemas computacionais o usuário conhece? Quais deles costuma utilizar? Que hardware costuma utilizar?;

Resposta: ( ) Conforme ( ) Não conforme ( ) Não se aplica

Critério de Fagan: se a resposta for Não conforme, o defeito é do tipo Omissão: falta no artefato um elemento importante que o item exige.

Fonte: Página: 69, Definição dos Conceitos (taxonomia de defeitos). Artigo: MARCOS KALINOWSKI, RODRIGO OLIVEIRA SPÍNOLA, Introdução à Inspeção de Software, Engenharia de Software Magazine, edição 01, p. 68 a 74.

Foto:



• **Omissão:** (1) Algum requisito importante relacionado à funcionalidade, ao desempenho, às restrições de projeto, ao atributo, ou à interface externa não foi incluído; (2) não está definida a resposta do software para todas as possíveis situações de entrada de dados; (3) faltam seções na especificação de requisitos; (4) faltam referências de figuras, tabelas, e diagramas; (5) falta definição de termos e unidades de medidas.

Edição 01 – **Engenharia de Software Magazine** **69**

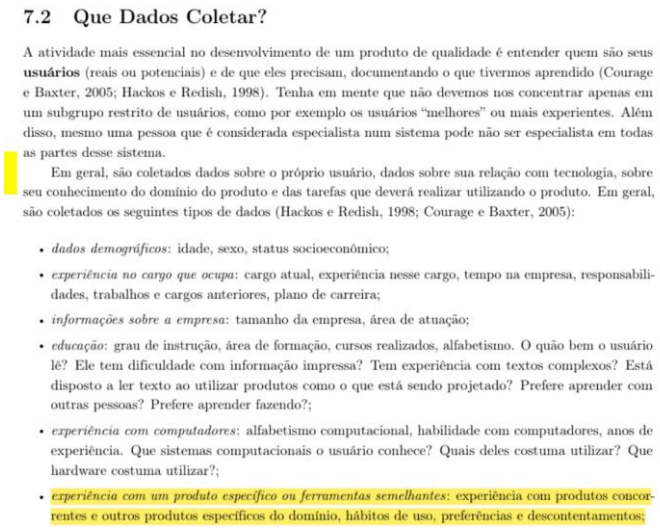
Severidade (Doolan, p. 178): ( ) Menor ( ) Maior ( ) Super-maior

Origem: Victor Andreozzi De La Rocque Couto, Lista de Verificação: Personas, item 5.

Item 16: A persona apresenta a experiência com produtos específicos ou ferramentas semelhantes, incluindo concorrentes, hábitos de uso, preferências e descontentamentos?

Fonte: Página: 137 (PDF disponibilizado no Aprender), Capítulo 7, Seção 7.2. Livro: SIMONE DINIZ JUNQUEIRA BARBOSA, BRUNO SANTANA DA SILVA, Interação Humano-Computador (PDF disponibilizado no Aprender).

Foto:



7.2 Que Dados Coletar?

A atividade mais essencial no desenvolvimento de um produto de qualidade é entender quem são seus **usuários** (reais ou potenciais) e de que eles precisam, documentando o que tivermos aprendido (Courage e Baxter, 2005; Hackos e Redish, 1998). Tenha em mente que não devemos nos concentrar apenas em um subgrupo restrito de usuários, como por exemplo os usuários "melhores" ou mais experientes. Além disso, mesmo uma pessoa que é considerada especialista num sistema pode não ser especialista em todas as partes desse sistema.

Em geral, são coletados dados sobre o próprio usuário, dados sobre sua relação com tecnologia, sobre seu conhecimento do domínio do produto e das tarefas que deverá realizar utilizando o produto. Em geral, são coletados os seguintes tipos de dados (Hackos e Redish, 1998; Courage e Baxter, 2005):

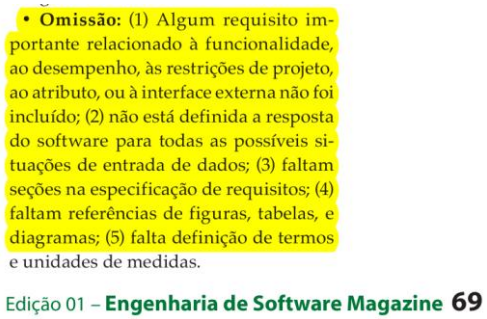
- *dados demográficos*: idade, sexo, status socioeconômico;
- *experiência no cargo que ocupa*: cargo atual, experiência nesse cargo, tempo na empresa, responsabilidades, trabalhos e cargos anteriores, plano de carreira;
- *informações sobre a empresa*: tamanho da empresa, área de atuação;
- *educação*: grau de instrução, área de formação, cursos realizados, alfabetismo. O quão bem o usuário lê? Ele tem dificuldade com informação impressa? Tem experiência com textos complexos? Está disposto a ler texto ao utilizar produtos como o que está sendo projetado? Prefere aprender com outras pessoas? Prefere aprender fazendo?;
- *experiência com computadores*: alfabetismo computacional, habilidade com computadores, anos de experiência. Que sistemas computacionais o usuário conhece? Quais deles costuma utilizar? Que hardware costuma utilizar?;
- *experiência com um produto específico ou ferramentas semelhantes*: experiência com produtos concorrentes e outros produtos específicos do domínio, hábitos de uso, preferências e descontentamentos;

Resposta: ( ) Conforme ( ) Não conforme ( ) Não se aplica

Critério de Fagan: se a resposta for Não conforme, o defeito é do tipo Omissão: falta no artefato um elemento importante que o item exige.

Fonte: Página: 69, Definição dos Conceitos (taxonomia de defeitos). Artigo: MARCOS KALINOWSKI, RODRIGO OLIVEIRA SPÍNOLA, Introdução à Inspeção de Software, Engenharia de Software Magazine, edição 01, p. 68 a 74.

Foto:



• **Omissão:** (1) Algum requisito importante relacionado à funcionalidade, ao desempenho, às restrições de projeto, ao atributo, ou à interface externa não foi incluído; (2) não está definida a resposta do software para todas as possíveis situações de entrada de dados; (3) faltam seções na especificação de requisitos; (4) faltam referências de figuras, tabelas, e diagramas; (5) falta definição de termos e unidades de medidas.

Edição 01 – **Engenharia de Software Magazine** **69**

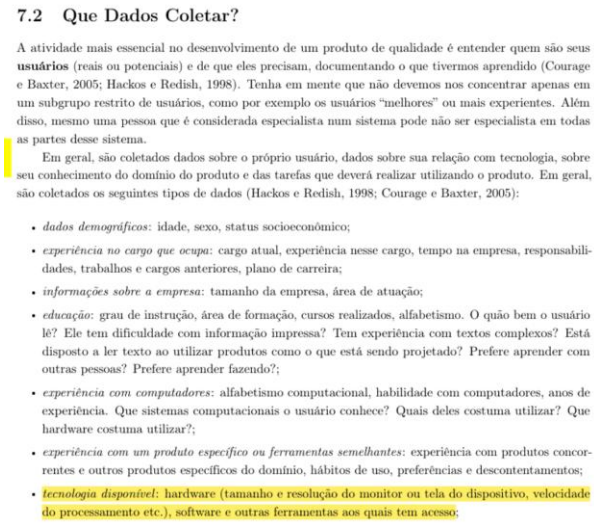
Severidade (Doolan, p. 178): ( ) Menor ( ) Maior ( ) Super-maior

Origem: Victor Andreozzi De La Rocque Couto, Lista de Verificação: Personas, item 6.

Item 17: A persona detalha a tecnologia disponível para o usuário, com o equipamento (tamanho e resolução da tela, processamento) e os programas e ferramentas de acesso?

Fonte: Página: 137 (PDF disponibilizado no Aprender), Capítulo 7, Seção 7.2. Livro: SIMONE DINIZ JUNQUEIRA BARBOSA, BRUNO SANTANA DA SILVA, Interação Humano-Computador (PDF disponibilizado no Aprender).

Foto:



7.2 Que Dados Coletar?

A atividade mais essencial no desenvolvimento de um produto de qualidade é entender quem são seus **usuários** (reais ou potenciais) e de que eles precisam, documentando o que tivermos aprendido (Courage e Baxter, 2005; Hackos e Redish, 1998). Tenha em mente que não devemos nos concentrar apenas em um subgrupo restrito de usuários, como por exemplo os usuários "melhores" ou mais experientes. Além disso, mesmo uma pessoa que é considerada especialista num sistema pode não ser especialista em todas as partes desse sistema.

Em geral, são coletados dados sobre o próprio usuário, dados sobre sua relação com tecnologia, sobre seu conhecimento do domínio do produto e das tarefas que deverá realizar utilizando o produto. Em geral, são coletados os seguintes tipos de dados (Hackos e Redish, 1998; Courage e Baxter, 2005):

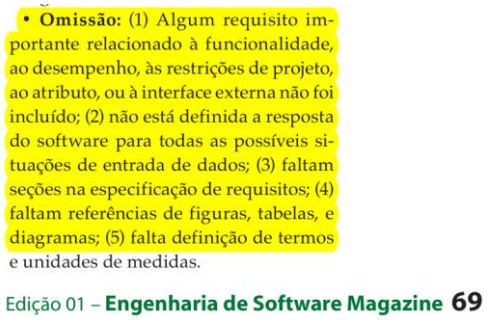
- *dados demográficos*: idade, sexo, status socioeconômico;
- *experiência no cargo que ocupa*: cargo atual, experiência nesse cargo, tempo na empresa, responsabilidades, trabalhos e cargos anteriores, plano de carreira;
- *informações sobre a empresa*: tamanho da empresa, área de atuação;
- *educação*: grau de instrução, área de formação, cursos realizados, alfabetismo. O quão bem o usuário lê? Ele tem dificuldade com informação impressa? Tem experiência com textos complexos? Está disposto a ler texto ao utilizar produtos como o que está sendo projetado? Prefere aprender com outras pessoas? Prefere aprender fazendo?;
- *experiência com computadores*: alfabetismo computacional, habilidade com computadores, anos de experiência. Que sistemas computacionais o usuário conhece? Quais deles costuma utilizar? Que hardware costuma utilizar?;
- *experiência com um produto específico ou ferramentas semelhantes*: experiência com produtos concorrentes e outros produtos específicos do domínio, hábitos de uso, preferências e descontentamentos;
- *tecnologia disponível*: hardware (tamanho e resolução do monitor ou tela do dispositivo, velocidade do processamento etc.), software e outras ferramentas aos quais tem acesso;

Resposta: ( ) Conforme ( ) Não conforme ( ) Não se aplica

Critério de Fagan: se a resposta for Não conforme, o defeito é do tipo Omissão: falta no artefato um elemento importante que o item exige.

Fonte: Página: 69, Definição dos Conceitos (taxonomia de defeitos). Artigo: MARCOS KALINOWSKI, RODRIGO OLIVEIRA SPÍNOLA, Introdução à Inspeção de Software, Engenharia de Software Magazine, edição 01, p. 68 a 74.

Foto:



• **Omissão:** (1) Algum requisito importante relacionado à funcionalidade, ao desempenho, às restrições de projeto, ao atributo, ou à interface externa não foi incluído; (2) não está definida a resposta do software para todas as possíveis situações de entrada de dados; (3) faltam seções na especificação de requisitos; (4) faltam referências de figuras, tabelas, e diagramas; (5) falta definição de termos e unidades de medidas.

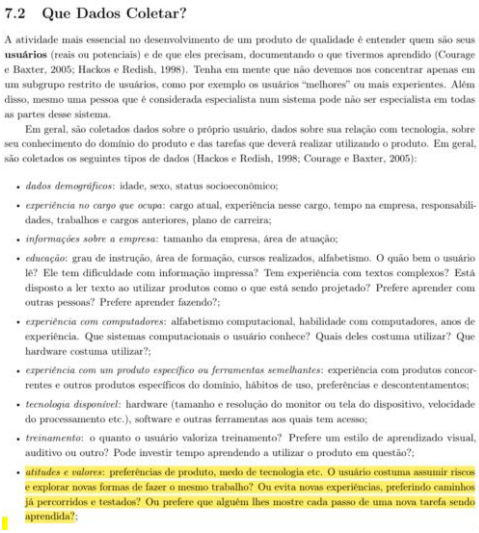
Severidade (Doolan, p. 178): ( ) Menor ( ) Maior ( ) Super-maior

Origem: Victor Andreozzi De La Rocque Couto, Lista de Verificação: Personas, item 7.

Item 18: A persona descreve atitudes e valores do usuário, como preferências de produto, medo de tecnologia e disposição para assumir riscos e explorar novidades ou preferência por caminhos já testados?

Fonte: Página: 137 e 138 (PDF disponibilizado no Aprender), Capítulo 7, Seção 7.2. Livro: SIMONE DINIZ JUNQUEIRA BARBOSA, BRUNO SANTANA DA SILVA, Interação Humano-Computador (PDF disponibilizado no Aprender).

Foto:



7.2 Que Dados Coletar?

A atividade mais essencial no desenvolvimento de um produto de qualidade é entender quem são seus **usuários** (reais ou potenciais) e de que eles precisam, documentando o que tivermos aprendido (Courage e Baxter, 2005; Hackos e Redish, 1998). Tenha em mente que não devemos nos concentrar apenas em um subgrupo restrito de usuários, como por exemplo os usuários "melhores" ou mais experientes. Além disso, mesmo uma pessoa que é considerada especialista num sistema pode não ser especialista em todas as partes desse sistema.

Em geral, são coletados dados sobre o próprio usuário, dados sobre sua relação com tecnologia, sobre seu conhecimento do domínio do produto e das tarefas que deverá realizar utilizando o produto. Em geral, são coletados os seguintes tipos de dados (Hackos e Redish, 1998; Courage e Baxter, 2005):

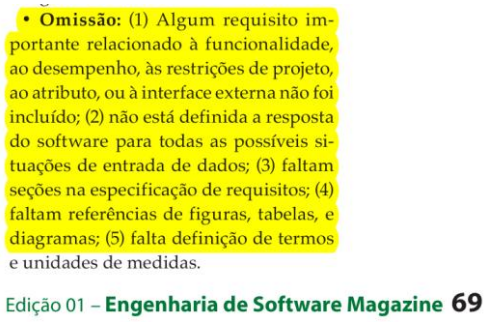
- *dados demográficos*: idade, sexo, status socioeconômico;
- *experiência no cargo que ocupa*: cargo atual, experiência nesse cargo, tempo na empresa, responsabilidades, trabalhos e cargos anteriores, plano de carreira;
- *informações sobre a empresa*: tamanho da empresa, área de atuação;
- *educação*: grau de instrução, área de formação, cursos realizados, alfabetismo. O quão bem o usuário lê? Ele tem dificuldade com informação impressa? Tem experiência com textos complexos? Está disposto a ler texto ao utilizar produtos como o que está sendo projetado? Prefere aprender com outras pessoas? Prefere aprender fazendo?;
- *experiência com computadores*: alfabetismo computacional, habilidade com computadores, anos de experiência. Que sistemas computacionais o usuário conhece? Quais deles costuma utilizar? Que hardware costuma utilizar?;
- *experiência com um produto específico ou ferramentas semelhantes*: experiência com produtos concorrentes e outros produtos específicos do domínio, hábitos de uso, preferências e descontentamentos;
- *tecnologia disponível*: hardware (tamanho e resolução do monitor ou tela do dispositivo, velocidade do processamento etc.), software e outras ferramentas aos quais tem acesso;
- *treinamento*: o quanto o usuário valoriza treinamento? Prefere um estilo de aprendizado visual, auditivo ou outro? Pode investir tempo aprendendo a utilizar o produto em questão?;
- *atitudes e valores*: preferências de produto, medo de tecnologia etc. O usuário costuma assumir riscos e explorar novas formas de fazer o mesmo trabalho? Ou evita novas experiências, preferindo caminhos já percorridos e testados? Ou prefere que alguém lhes mostre cada passo de uma nova tarefa sendo aprendida?;

Resposta: ( ) Conforme ( ) Não conforme ( ) Não se aplica

Critério de Fagan: se a resposta for Não conforme, o defeito é do tipo Omissão: falta no artefato um elemento importante que o item exige.

Fonte: Página: 69, Definição dos Conceitos (taxonomia de defeitos). Artigo: MARCOS KALINOWSKI, RODRIGO OLIVEIRA SPÍNOLA, Introdução à Inspeção de Software, Engenharia de Software Magazine, edição 01, p. 68 a 74.

Foto:



• **Omissão:** (1) Algum requisito importante relacionado à funcionalidade, ao desempenho, às restrições de projeto, ao atributo, ou à interface externa não foi incluído; (2) não está definida a resposta do software para todas as possíveis situações de entrada de dados; (3) faltam seções na especificação de requisitos; (4) faltam referências de figuras, tabelas, e diagramas; (5) falta definição de termos e unidades de medidas.

Edição 01 – **Engenharia de Software Magazine** **69**

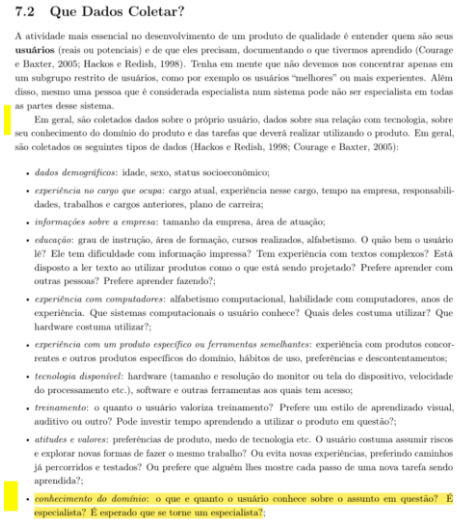
Severidade (Doolan, p. 178): ( ) Menor ( ) Maior ( ) Super-maior

Origem: Victor Andreozzi De La Rocque Couto, Lista de Verificação: Personas, item 8.

Item 19: A persona define o quanto o usuário conhece do domínio e se é esperado que ele se torne especialista?

Fonte: Página: 138 (PDF disponibilizado no Aprender), Capítulo 7, Seção 7.2. Livro: SIMONE DINIZ JUNQUEIRA BARBOSA, BRUNO SANTANA DA SILVA, Interação Humano-Computador (PDF disponibilizado no Aprender).

Foto:



7.2 Que Dados Coletar?

A atividade mais essencial no desenvolvimento de um produto de qualidade é entender quem são seus **usuários** (reais ou potenciais) e de que eles precisam, documentando o que tivermos aprendido (Courage e Baxter, 2005; Hackos e Redish, 1998). Tenha em mente que não devemos nos concentrar apenas em um subgrupo restrito de usuários, como por exemplo os usuários "melhores" ou mais experientes. Além disso, mesmo uma pessoa que é considerada especialista num sistema pode não ser especialista em todas as partes desse sistema.

Em geral, são coletados dados sobre o próprio usuário, dados sobre sua relação com tecnologia, sobre seu conhecimento do domínio do produto e das tarefas que deverá realizar utilizando o produto. Em geral, são coletados os seguintes tipos de dados (Hackos e Redish, 1998; Courage e Baxter, 2005):

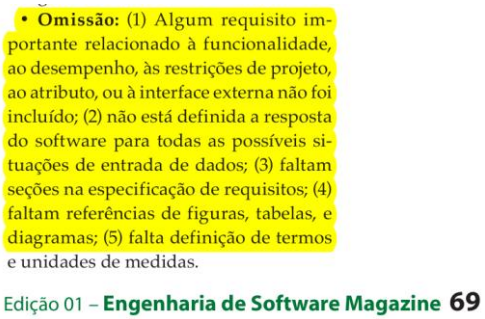
- *dados demográficos*: idade, sexo, status socioeconômico;
- *experiência no cargo que ocupa*: cargo atual, experiência nesse cargo, tempo na empresa, responsabilidades, trabalhos e cargos anteriores, plano de carreira;
- *informações sobre a empresa*: tamanho da empresa, área de atuação;
- *educação*: grau de instrução, área de formação, cursos realizados, alfabetismo. O quão bem o usuário lê? Ele tem dificuldade com informação impressa? Tem experiência com textos complexos? Está disposto a ler texto ao utilizar produtos como o que está sendo projetado? Prefere aprender com outras pessoas? Prefere aprender fazendo?;
- *experiência com computadores*: alfabetismo computacional, habilidade com computadores, anos de experiência. Que sistemas computacionais o usuário conhece? Quais deles costuma utilizar? Que hardware costuma utilizar?;
- *experiência com um produto específico ou ferramentas semelhantes*: experiência com produtos concorrentes e outros produtos específicos do domínio, hábitos de uso, preferências e descontentamentos;
- *tecnologia disponível*: hardware (tamanho e resolução do monitor ou tela do dispositivo, velocidade do processamento etc.), software e outras ferramentas aos quais tem acesso;
- *treinamento*: o quanto o usuário valoriza treinamento? Prefere um estilo de aprendizado visual, auditivo ou outro? Pode investir tempo aprendendo a utilizar o produto em questão?;
- *atitudes e valores*: preferências de produto, medo de tecnologia etc. O usuário costuma assumir riscos e explorar novas formas de fazer o mesmo trabalho? Ou evita novas experiências, preferindo caminhos já percorridos e testados? Ou prefere que alguém lhes mostre cada passo de uma nova tarefa sendo aprendida?;
- *conhecimento do domínio*: o que e quanto o usuário conhece sobre o assunto em questão? É especialista? É esperado que se torne um especialista?;

Resposta: ( ) Conforme ( ) Não conforme ( ) Não se aplica

Critério de Fagan: se a resposta for Não conforme, o defeito é do tipo Omissão: falta no artefato um elemento importante que o item exige.

Fonte: Página: 69, Definição dos Conceitos (taxonomia de defeitos). Artigo: MARCOS KALINOWSKI, RODRIGO OLIVEIRA SPÍNOLA, Introdução à Inspeção de Software, Engenharia de Software Magazine, edição 01, p. 68 a 74.

Foto:



• **Omissão:** (1) Algum requisito importante relacionado à funcionalidade, ao desempenho, às restrições de projeto, ao atributo, ou à interface externa não foi incluído; (2) não está definida a resposta do software para todas as possíveis situações de entrada de dados; (3) faltam seções na especificação de requisitos; (4) faltam referências de figuras, tabelas, e diagramas; (5) falta definição de termos e unidades de medidas.

Edição 01 – **Engenharia de Software Magazine** **69**

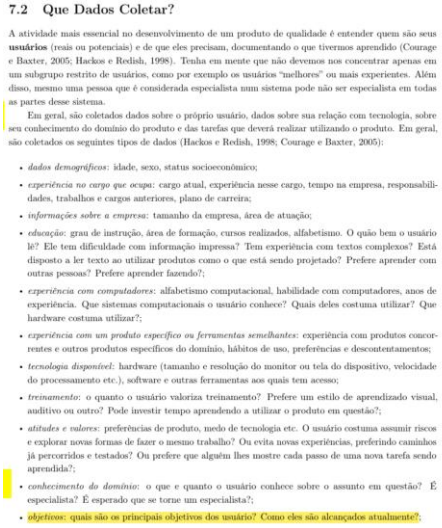
Severidade (Doolan, p. 178): ( ) Menor ( ) Maior ( ) Super-maior

Origem: Victor Andreozzi De La Rocque Couto, Lista de Verificação: Personas, item 9.

Item 20: A persona especifica os principais objetivos do usuário e como eles são alcançados atualmente?

Fonte: Página: 138 (PDF disponibilizado no Aprender), Capítulo 7, Seção 7.2. Livro: SIMONE DINIZ JUNQUEIRA BARBOSA, BRUNO SANTANA DA SILVA, Interação Humano-Computador (PDF disponibilizado no Aprender).

Foto:



7.2 Que Dados Coletar?

A atividade mais essencial no desenvolvimento de um produto de qualidade é entender quem são seus **usuários** (reais ou potenciais) e de que eles precisam, documentando o que tivermos aprendido (Courage e Baxter, 2005; Hackos e Redish, 1998). Tenha em mente que não devemos nos concentrar apenas em um subgrupo restrito de usuários, como por exemplo os usuários "melhores" ou mais experientes. Além disso, mesmo uma pessoa que é considerada especialista num sistema pode não ser especialista em todas as partes desse sistema.

Em geral, são coletados dados sobre o próprio usuário, dados sobre sua relação com tecnologia, sobre seu conhecimento do domínio do produto e das tarefas que deverá realizar utilizando o produto. Em geral, são coletados os seguintes tipos de dados (Hackos e Redish, 1998; Courage e Baxter, 2005):

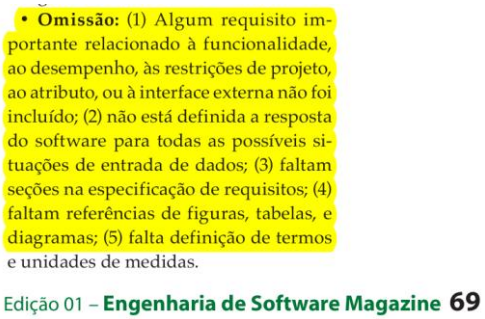
- *dados demográficos*: idade, sexo, status socioeconômico;
- *experiência no cargo que ocupa*: cargo atual, experiência nesse cargo, tempo na empresa, responsabilidades, trabalhos e cargos anteriores, plano de carreira;
- *informações sobre a empresa*: tamanho da empresa, área de atuação;
- *educação*: grau de instrução, área de formação, cursos realizados, alfabetismo. O quão bem o usuário lê? Ele tem dificuldade com informação impressa? Tem experiência com textos complexos? Está disposto a ler texto ao utilizar produtos como o que está sendo projetado? Prefere aprender com outras pessoas? Prefere aprender fazendo?;
- *experiência com computadores*: alfabetismo computacional, habilidade com computadores, anos de experiência. Que sistemas computacionais o usuário conhece? Quais deles costuma utilizar? Que hardware costuma utilizar?;
- *experiência com um produto específico ou ferramentas semelhantes*: experiência com produtos concorrentes e outros produtos específicos do domínio, hábitos de uso, preferências e descontentamentos;
- *tecnologia disponível*: hardware (tamanho e resolução do monitor ou tela do dispositivo, velocidade do processamento etc.), software e outras ferramentas aos quais tem acesso;
- *treinamento*: o quanto o usuário valoriza treinamento? Prefere um estilo de aprendizado visual, auditivo ou outro? Pode investir tempo aprendendo a utilizar o produto em questão?;
- *atitudes e valores*: preferências de produto, medo de tecnologia etc. O usuário costuma assumir riscos e explorar novas formas de fazer o mesmo trabalho? Ou evita novas experiências, preferindo caminhos já percorridos e testados? Ou prefere que alguém lhes mostre cada passo de uma nova tarefa sendo aprendida?;
- *conhecimento do domínio*: o que e quanto o usuário conhece sobre o assunto em questão? É especialista? É esperado que se torne um especialista?;
- *objetivos*: quais são os principais objetivos dos usuário? Como eles são alcançados atualmente?.

Resposta: ( ) Conforme ( ) Não conforme ( ) Não se aplica

Critério de Fagan: se a resposta for Não conforme, o defeito é do tipo Omissão: falta no artefato um elemento importante que o item exige.

Fonte: Página: 69, Definição dos Conceitos (taxonomia de defeitos). Artigo: MARCOS KALINOWSKI, RODRIGO OLIVEIRA SPÍNOLA, Introdução à Inspeção de Software, Engenharia de Software Magazine, edição 01, p. 68 a 74.

Foto:



• **Omissão:** (1) Algum requisito importante relacionado à funcionalidade, ao desempenho, às restrições de projeto, ao atributo, ou à interface externa não foi incluído; (2) não está definida a resposta do software para todas as possíveis situações de entrada de dados; (3) faltam seções na especificação de requisitos; (4) faltam referências de figuras, tabelas, e diagramas; (5) falta definição de termos e unidades de medidas.

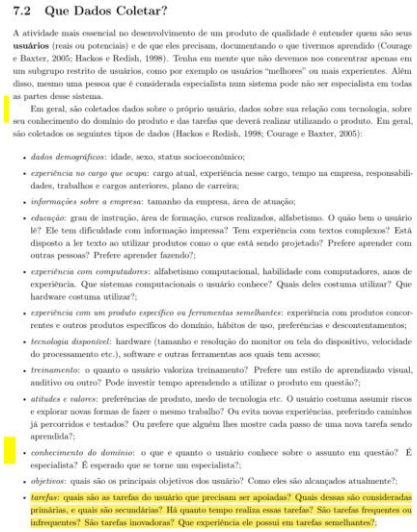
Severidade (Doolan, p. 178): ( ) Menor ( ) Maior ( ) Super-maior

Origem: Victor Andreozzi De La Rocque Couto, Lista de Verificação: Personas, item 10.

Item 21: A persona descreve as tarefas que precisam ser apoiadas, quais são primárias e quais são secundárias, com a frequência, o tempo de realização e a experiência em tarefas semelhantes?

Fonte: Página: 138 (PDF disponibilizado no Aprender), Capítulo 7, Seção 7.2. Livro: SIMONE DINIZ JUNQUEIRA BARBOSA, BRUNO SANTANA DA SILVA, Interação Humano-Computador (PDF disponibilizado no Aprender).

Foto:



7.2 Que Dados Coletar?

A atividade mais essencial no desenvolvimento de um produto de qualidade é entender quem são seus **usuários** (reais ou potenciais) e de que eles precisam, documentando o que tivermos aprendido (Courage e Baxter, 2005; Hackos e Redish, 1998). Tenha em mente que não devemos nos concentrar apenas em um subgrupo restrito de usuários, como por exemplo os usuários "melhores" ou mais experientes. Além disso, mesmo uma pessoa que é considerada especialista num sistema pode não ser especialista em todas as partes desse sistema.

Em geral, são coletados dados sobre o próprio usuário, dados sobre sua relação com tecnologia, sobre seu conhecimento do domínio do produto e das tarefas que deverá realizar utilizando o produto. Em geral, são coletados os seguintes tipos de dados (Hackos e Redish, 1998; Courage e Baxter, 2005):

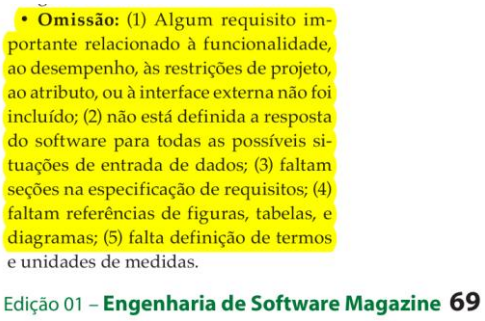
- *dados demográficos*: idade, sexo, status socioeconômico;
- *experiência no cargo que ocupa*: cargo atual, experiência nesse cargo, tempo na empresa, responsabilidades, trabalhos e cargos anteriores, plano de carreira;
- *informações sobre a empresa*: tamanho da empresa, área de atuação;
- *educação*: grau de instrução, área de formação, cursos realizados, alfabetismo. O quão bem o usuário lê? Ele tem dificuldade com informação impressa? Tem experiência com textos complexos? Está disposto a ler texto ao utilizar produtos como o que está sendo projetado? Prefere aprender com outras pessoas? Prefere aprender fazendo?;
- *experiência com computadores*: alfabetismo computacional, habilidade com computadores, anos de experiência. Que sistemas computacionais o usuário conhece? Quais deles costuma utilizar? Que hardware costuma utilizar?;
- *experiência com um produto específico ou ferramentas semelhantes*: experiência com produtos concorrentes e outros produtos específicos do domínio, hábitos de uso, preferências e descontentamentos;
- *tecnologia disponível*: hardware (tamanho e resolução do monitor ou tela do dispositivo, velocidade do processamento etc.), software e outras ferramentas aos quais tem acesso;
- *treinamento*: o quanto o usuário valoriza treinamento? Prefere um estilo de aprendizado visual, auditivo ou outro? Pode investir tempo aprendendo a utilizar o produto em questão?;
- *atitudes e valores*: preferências de produto, medo de tecnologia etc. O usuário costuma assumir riscos e explorar novas formas de fazer o mesmo trabalho? Ou evita novas experiências, preferindo caminhos já percorridos e testados? Ou prefere que alguém lhes mostre cada passo de uma nova tarefa sendo aprendida?;
- *conhecimento do domínio*: o que e quanto o usuário conhece sobre o assunto em questão? É especialista? É esperado que se torne um especialista?;
- *objetivos*: quais são os principais objetivos dos usuário? Como eles são alcançados atualmente?;
- *tarefas*: quais são as tarefas do usuário que precisam ser apoiadas? Quais dessas são consideradas primárias, e quais são secundárias? Há quanto tempo realiza essas tarefas? São tarefas frequentes ou infrequentes? São tarefas inovadoras? Que experiência ele possui em tarefas semelhantes?.

Resposta: ( ) Conforme ( ) Não conforme ( ) Não se aplica

Critério de Fagan: se a resposta for Não conforme, o defeito é do tipo Omissão: falta no artefato um elemento importante que o item exige.

Fonte: Página: 69, Definição dos Conceitos (taxonomia de defeitos). Artigo: MARCOS KALINOWSKI, RODRIGO OLIVEIRA SPÍNOLA, Introdução à Inspeção de Software, Engenharia de Software Magazine, edição 01, p. 68 a 74.

Foto:



• **Omissão:** (1) Algum requisito importante relacionado à funcionalidade, ao desempenho, às restrições de projeto, ao atributo, ou à interface externa não foi incluído; (2) não está definida a resposta do software para todas as possíveis situações de entrada de dados; (3) faltam seções na especificação de requisitos; (4) faltam referências de figuras, tabelas, e diagramas; (5) falta definição de termos e unidades de medidas.

Edição 01 – **Engenharia de Software Magazine** **69**

Severidade (Doolan, p. 178): ( ) Menor ( ) Maior ( ) Super-maior

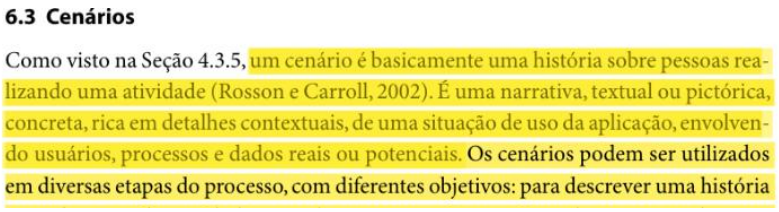
Origem: Victor Andreozzi De La Rocque Couto, Lista de Verificação: Personas, item 11.

4.3 Cenários

Item 22: Cada cenário está escrito como uma narrativa concreta e rica em detalhes de contexto, isto é, uma história sobre pessoas realizando uma atividade, e não como uma lista abstrata de funcionalidades ou de requisitos?

Fonte: Página: 183, Capítulo 6. Organização do Espaço de Problema. Livro: SIMONE DINIZ JUNQUEIRA BARBOSA, BRUNO SANTANA DA SILVA, Interação Humano-Computador, 1ª ed., Editora Campus/Elsevier, 2010.

Foto:



6.3 Cenários

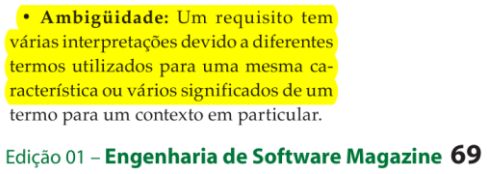
Como visto na Seção 4.3.5, um cenário é basicamente uma história sobre pessoas realizando uma atividade (Rosson e Carroll, 2002). É uma narrativa, textual ou pictórica, concreta, rica em detalhes contextuais, de uma situação de uso da aplicação, envolvendo usuários, processos e dados reais ou potenciais. Os cenários podem ser utilizados em diversas etapas do processo, com diferentes objetivos: para descrever uma história

Resposta: ( ) Conforme ( ) Não conforme ( ) Não se aplica

Critério de Fagan: se a resposta for Não conforme, o defeito é do tipo Ambiguidade: o artefato admite mais de uma interpretação.

Fonte: Página: 69, Definição dos Conceitos (taxonomia de defeitos). Artigo: MARCOS KALINOWSKI, RODRIGO OLIVEIRA SPÍNOLA, Introdução à Inspeção de Software, Engenharia de Software Magazine, edição 01, p. 68 a 74.

Foto:



• **Ambigüidade:** Um requisito tem várias interpretações devido a diferentes termos utilizados para uma mesma característica ou vários significados de um termo para um contexto em particular.

Edição 01 – **Engenharia de Software Magazine** **69**

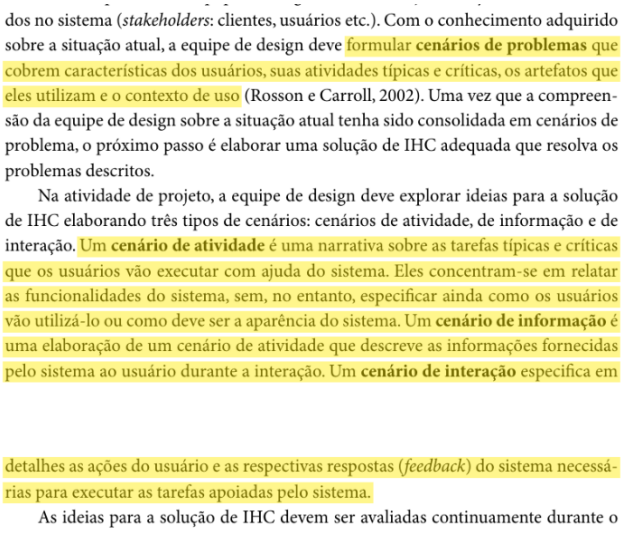
Severidade (Doolan, p. 178): ( ) Menor ( ) Maior ( ) Super-maior

Origem: Victor Andreozzi De La Rocque Couto, Lista de Verificação: Cenários, item 1.

Item 23: O tipo de cada cenário está identificado e é coerente com a etapa do projeto: cenário de problema, de atividade, de informação ou de interação?

Fonte: Página: 113 e 114, Capítulo 4. Processos de Design de IHC. Livro: SIMONE DINIZ JUNQUEIRA BARBOSA, BRUNO SANTANA DA SILVA, Interação Humano-Computador, 1ª ed., Editora Campus/Elsevier, 2010.

Foto:



dos no sistema (*stakeholders*: clientes, usuários etc.). Com o conhecimento adquirido sobre a situação atual, a equipe de design deve formular **cenários de problemas** que cobrem características dos usuários, suas atividades típicas e críticas, os artefatos que eles utilizam e o contexto de uso (Rosson e Carroll, 2002). Uma vez que a compreensão da equipe de design sobre a situação atual tenha sido consolidada em cenários de problema, o próximo passo é elaborar uma solução de IHC adequada que resolva os problemas descritos.

Na atividade de projeto, a equipe de design deve explorar ideias para a solução de IHC elaborando três tipos de cenários: cenários de atividade, de informação e de interação. Um **cenário de atividade** é uma narrativa sobre as tarefas típicas e críticas que os usuários vão executar com ajuda do sistema. Eles concentram-se em relatar as funcionalidades do sistema, sem, no entanto, especificar ainda como os usuários vão utilizá-lo ou como deve ser a aparência do sistema. Um **cenário de informação** é uma elaboração de um cenário de atividade que descreve as informações fornecidas pelo sistema ao usuário durante a interação. Um **cenário de interação** especifica em

detalhes as ações do usuário e as respectivas respostas (*feedback*) do sistema necessárias para executar as tarefas apoiadas pelo sistema.

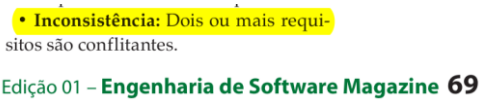
As ideias para a solução de IHC devem ser avaliadas continuamente durante o

Resposta: ( ) Conforme ( ) Não conforme ( ) Não se aplica

Critério de Fagan: se a resposta for Não conforme, o defeito é do tipo Inconsistência: partes do artefato, ou o artefato e outro artefato do projeto, são conflitantes.

Fonte: Página: 69, Definição dos Conceitos (taxonomia de defeitos). Artigo: MARCOS KALINOWSKI, RODRIGO OLIVEIRA SPÍNOLA, Introdução à Inspeção de Software, Engenharia de Software Magazine, edição 01, p. 68 a 74.

Foto:



• **Inconsistência:** Dois ou mais requisitos são conflitantes.

Edição 01 – **Engenharia de Software Magazine** **69**

Severidade (Doolan, p. 178): ( ) Menor ( ) Maior ( ) Super-maior

Origem: Victor Andreozzi De La Rocque Couto, Lista de Verificação: Cenários, item 2.

Item 24: Cada cenário traz os componentes de identificação: um título breve, a relação dos atores, uma descrição da situação inicial e referências a outros cenários que levem ao mesmo objetivo, quando existirem?

Fonte: Página: 183, Capítulo 6. Organização do Espaço de Problema. Livro: SIMONE DINIZ JUNQUEIRA BARBOSA, BRUNO SANTANA DA SILVA, Interação Humano-Computador, 1ª ed., Editora Campus/Elsevier, 2010.

Foto:

uma situação, o cenário pode incluir informações sobre planejamento e avaliação das ações realizadas. Cada cenário costuma ter um título que descreve brevemente a situação, sem muitos detalhes; os atores que participam do cenário; uma breve descrição da situação inicial em que os atores se encontram; e referências a outros cenários que permitam aos atores atingir os mesmos objetivos de diferentes maneiras. Segundo Rosson e Carroll (2002), os elementos característicos de um cenário são:

Resposta: ( ) Conforme ( ) Não conforme ( ) Não se aplica

Critério de Fagan: se a resposta for Não conforme, o defeito é do tipo Omissão: falta no artefato um elemento importante que o item exige.

Fonte: Página: 69, Definição dos Conceitos (taxonomia de defeitos). Artigo: MARCOS KALINOWSKI, RODRIGO OLIVEIRA SPÍNOLA, Introdução à Inspeção de Software, Engenharia de Software Magazine, edição 01, p. 68 a 74.

Foto:

• **Omissão:** (1) Algum requisito importante relacionado à funcionalidade, ao desempenho, às restrições de projeto, ao atributo, ou à interface externa não foi incluído; (2) não está definida a resposta do software para todas as possíveis situações de entrada de dados; (3) faltam seções na especificação de requisitos; (4) faltam referências de figuras, tabelas, e diagramas; (5) falta definição de termos e unidades de medidas.

Edição 01 – **Engenharia de Software Magazine** **69**

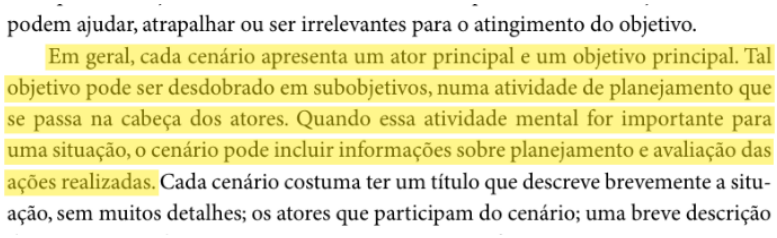
Severidade (Doolan, p. 178): ( ) Menor ( ) Maior ( ) Super-maior

Origem: Victor Andreozzi De La Rocque Couto, Lista de Verificação: Cenários, item 3.

Item 25: Em cada cenário é possível identificar um ator principal e um objetivo principal, e a história tem um enredo com uma sequência de ações e eventos (o que os usuários fazem, o que acontece com eles e que mudanças ocorrem no ambiente)?

Fonte: Página: 183, Capítulo 6. Organização do Espaço de Problema. Livro: SIMONE DINIZ JUNQUEIRA BARBOSA, BRUNO SANTANA DA SILVA, Interação Humano-Computador, 1ª ed., Editora Campus/Elsevier, 2010.

Foto:



podem ajudar, atrapalhar ou ser irrelevantes para o atingimento do objetivo.

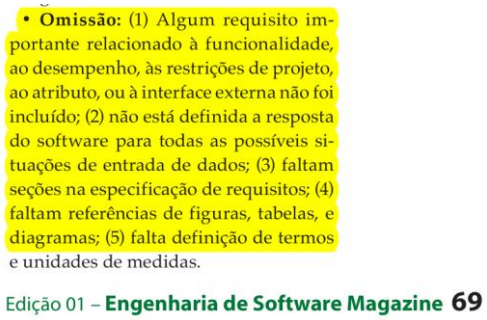
Em geral, cada cenário apresenta um ator principal e um objetivo principal. Tal objetivo pode ser desdobrado em subobjetivos, numa atividade de planejamento que se passa na cabeça dos atores. Quando essa atividade mental for importante para uma situação, o cenário pode incluir informações sobre planejamento e avaliação das ações realizadas. Cada cenário costuma ter um título que descreve brevemente a situação, sem muitos detalhes; os atores que participam do cenário; uma breve descrição

Resposta: ( ) Conforme ( ) Não conforme ( ) Não se aplica

Critério de Fagan: se a resposta for Não conforme, o defeito é do tipo Omissão: falta no artefato um elemento importante que o item exige.

Fonte: Página: 69, Definição dos Conceitos (taxonomia de defeitos). Artigo: MARCOS KALINOWSKI, RODRIGO OLIVEIRA SPÍNOLA, Introdução à Inspeção de Software, Engenharia de Software Magazine, edição 01, p. 68 a 74.

Foto:



• **Omissão:** (1) Algum requisito importante relacionado à funcionalidade, ao desempenho, às restrições de projeto, ao atributo, ou à interface externa não foi incluído; (2) não está definida a resposta do software para todas as possíveis situações de entrada de dados; (3) faltam seções na especificação de requisitos; (4) faltam referências de figuras, tabelas, e diagramas; (5) falta definição de termos e unidades de medidas.

Edição 01 – **Engenharia de Software Magazine** **69**

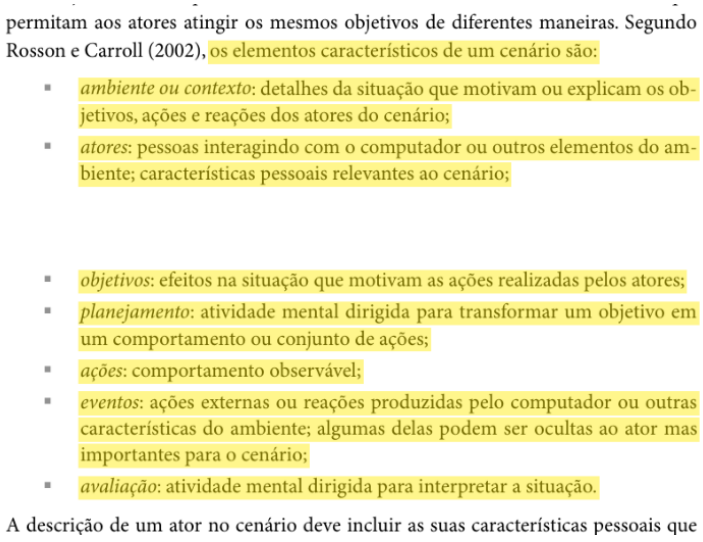
Severidade (Doolan, p. 178): ( ) Menor ( ) Maior ( ) Super-maior

Origem: Victor Andreozzi De La Rocque Couto, Lista de Verificação: Cenários, item 4.

Item 26: Cada cenário contém os sete elementos característicos: ambiente ou contexto, atores, objetivos, planejamento, ações, eventos e avaliação?

Fonte: Página: 183 e 184, Capítulo 6. Organização do Espaço de Problema. Livro: SIMONE DINIZ JUNQUEIRA BARBOSA, BRUNO SANTANA DA SILVA, Interação Humano-Computador, 1ª ed., Editora Campus/Elsevier, 2010.

Foto:



permitam aos atores atingir os mesmos objetivos de diferentes maneiras. Segundo Rosson e Carroll (2002), os elementos característicos de um cenário são:

- *ambiente ou contexto*: detalhes da situação que motivam ou explicam os objetivos, ações e reações dos atores do cenário;
- *atores*: pessoas interagindo com o computador ou outros elementos do ambiente; características pessoais relevantes ao cenário;
- *objetivos*: efeitos na situação que motivam as ações realizadas pelos atores;
- *planejamento*: atividade mental dirigida para transformar um objetivo em um comportamento ou conjunto de ações;
- *ações*: comportamento observável;
- *eventos*: ações externas ou reações produzidas pelo computador ou outras características do ambiente; algumas delas podem ser ocultas ao ator mas importantes para o cenário;
- *avaliação*: atividade mental dirigida para interpretar a situação.

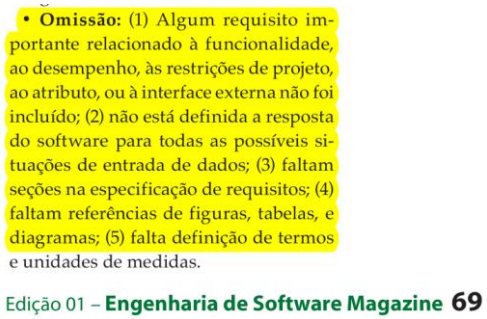
A descrição de um ator no cenário deve incluir as suas características pessoais que

Resposta: ( ) Conforme ( ) Não conforme ( ) Não se aplica

Critério de Fagan: se a resposta for Não conforme, o defeito é do tipo Omissão: falta no artefato um elemento importante que o item exige.

Fonte: Página: 69, Definição dos Conceitos (taxonomia de defeitos). Artigo: MARCOS KALINOWSKI, RODRIGO OLIVEIRA SPÍNOLA, Introdução à Inspeção de Software, Engenharia de Software Magazine, edição 01, p. 68 a 74.

Foto:



• **Omissão:** (1) Algum requisito importante relacionado à funcionalidade, ao desempenho, às restrições de projeto, ao atributo, ou à interface externa não foi incluído; (2) não está definida a resposta do software para todas as possíveis situações de entrada de dados; (3) faltam seções na especificação de requisitos; (4) faltam referências de figuras, tabelas, e diagramas; (5) falta definição de termos e unidades de medidas.

Edição 01 – **Engenharia de Software Magazine** **69**

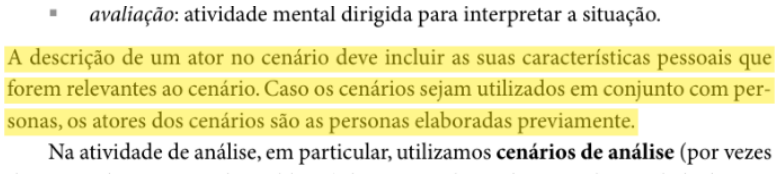
Severidade (Doolan, p. 178): ( ) Menor ( ) Maior ( ) Super-maior

Origem: Victor Andreozzi De La Rocque Couto, Lista de Verificação: Cenários, item 5.

Item 27: A descrição de cada ator inclui as características pessoais relevantes para o cenário e, quando o projeto tem personas, os atores são as próprias personas, com o mesmo nome?

Fonte: Página: 184, Capítulo 6. Organização do Espaço de Problema. Livro: SIMONE DINIZ JUNQUEIRA BARBOSA, BRUNO SANTANA DA SILVA, Interação Humano-Computador, 1ª ed., Editora Campus/Elsevier, 2010.

Foto:



- *avaliação*: atividade mental dirigida para interpretar a situação.

A descrição de um ator no cenário deve incluir as suas características pessoais que forem relevantes ao cenário. Caso os cenários sejam utilizados em conjunto com personas, os atores dos cenários são as personas elaboradas previamente.

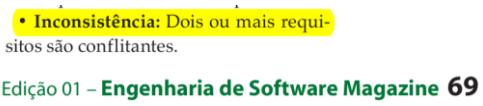
Na atividade de análise, em particular, utilizamos **cenários de análise** (por vezes

Resposta: ( ) Conforme ( ) Não conforme ( ) Não se aplica

Critério de Fagan: se a resposta for Não conforme, o defeito é do tipo Inconsistência: partes do artefato, ou o artefato e outro artefato do projeto, são conflitantes.

Fonte: Página: 69, Definição dos Conceitos (taxonomia de defeitos). Artigo: MARCOS KALINOWSKI, RODRIGO OLIVEIRA SPÍNOLA, Introdução à Inspeção de Software, Engenharia de Software Magazine, edição 01, p. 68 a 74.

Foto:



• **Inconsistência:** Dois ou mais requisitos são conflitantes.

Edição 01 – **Engenharia de Software Magazine** **69**

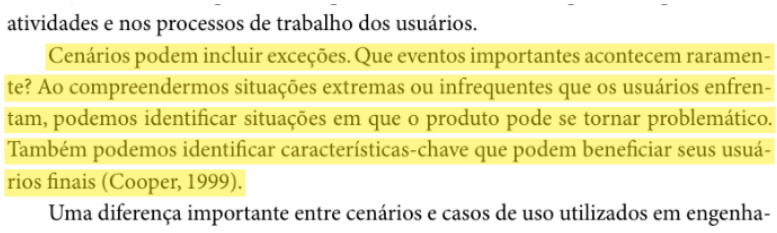
Severidade (Doolan, p. 178): ( ) Menor ( ) Maior ( ) Super-maior

Origem: Victor Andreozzi De La Rocque Couto, Lista de Verificação: Cenários, item 6.

Item 28: O conjunto de cenários inclui exceções, isto é, ao menos um cenário com eventos raros ou situações extremas (um dado faltando, um erro, uma interrupção da tarefa) em que o produto pode se tornar problemático?

Fonte: Página: 185, Capítulo 6. Organização do Espaço de Problema. Livro: SIMONE DINIZ JUNQUEIRA BARBOSA, BRUNO SANTANA DA SILVA, Interação Humano-Computador, 1ª ed., Editora Campus/Elsevier, 2010.

Foto:



atividades e nos processos de trabalho dos usuários.

Cenários podem incluir exceções. Que eventos importantes acontecem raramente? Ao compreendermos situações extremas ou infrequentes que os usuários enfrentam, podemos identificar situações em que o produto pode se tornar problemático. Também podemos identificar características-chave que podem beneficiar seus usuários finais (Cooper, 1999).

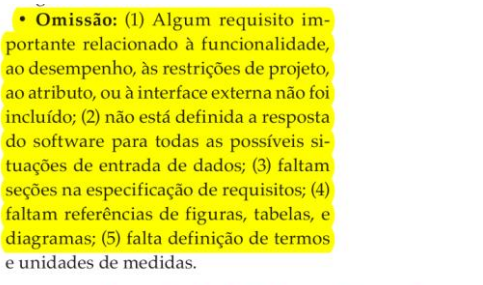
Uma diferença importante entre cenários e casos de uso utilizados em engenha-

Resposta: ( ) Conforme ( ) Não conforme ( ) Não se aplica

Critério de Fagan: se a resposta for Não conforme, o defeito é do tipo Omissão: falta no artefato um elemento importante que o item exige.

Fonte: Página: 69, Definição dos Conceitos (taxonomia de defeitos). Artigo: MARCOS KALINOWSKI, RODRIGO OLIVEIRA SPÍNOLA, Introdução à Inspeção de Software, Engenharia de Software Magazine, edição 01, p. 68 a 74.

Foto:



• **Omissão:** (1) Algum requisito importante relacionado à funcionalidade, ao desempenho, às restrições de projeto, ao atributo, ou à interface externa não foi incluído; (2) não está definida a resposta do software para todas as possíveis situações de entrada de dados; (3) faltam seções na especificação de requisitos; (4) faltam referências de figuras, tabelas, e diagramas; (5) falta definição de termos e unidades de medidas.

Edição 01 – **Engenharia de Software Magazine** **69**

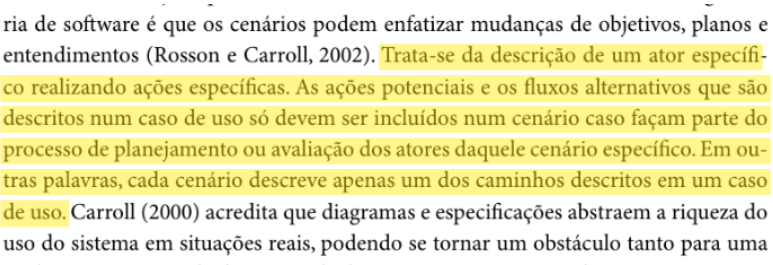
Severidade (Doolan, p. 178): ( ) Menor ( ) Maior ( ) Super-maior

Origem: Victor Andreozzi De La Rocque Couto, Lista de Verificação: Cenários, item 7.

Item 29: Cada cenário descreve um ator específico realizando ações específicas e segue apenas um dos caminhos possíveis da atividade, diferentemente de um caso de uso, que lista todas as ações e fluxos alternativos?

Fonte: Página: 185, Capítulo 6. Organização do Espaço de Problema. Livro: SIMONE DINIZ JUNQUEIRA BARBOSA, BRUNO SANTANA DA SILVA, Interação Humano-Computador, 1ª ed., Editora Campus/Elsevier, 2010.

Foto:



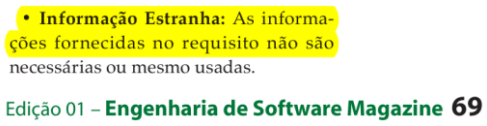
ria de software é que os cenários podem enfatizar mudanças de objetivos, planos e entendimentos (Rosson e Carroll, 2002). Trata-se da descrição de um ator específico realizando ações específicas. As ações potenciais e os fluxos alternativos que são descritos num caso de uso só devem ser incluídos num cenário caso façam parte do processo de planejamento ou avaliação dos atores daquele cenário específico. Em outras palavras, cada cenário descreve apenas um dos caminhos descritos em um caso de uso. Carroll (2000) acredita que diagramas e especificações abstraem a riqueza do uso do sistema em situações reais, podendo se tornar um obstáculo tanto para uma

Resposta: ( ) Conforme ( ) Não conforme ( ) Não se aplica

Critério de Fagan: se a resposta for Não conforme, o defeito é do tipo Informação estranha: o artefato traz informação desnecessária ou que não é usada.

Fonte: Página: 69, Definição dos Conceitos (taxonomia de defeitos). Artigo: MARCOS KALINOWSKI, RODRIGO OLIVEIRA SPÍNOLA, Introdução à Inspeção de Software, Engenharia de Software Magazine, edição 01, p. 68 a 74.

Foto:



• **Informação Estranha:** As informações fornecidas no requisito não são necessárias ou mesmo usadas.

Edição 01 – **Engenharia de Software Magazine** **69**

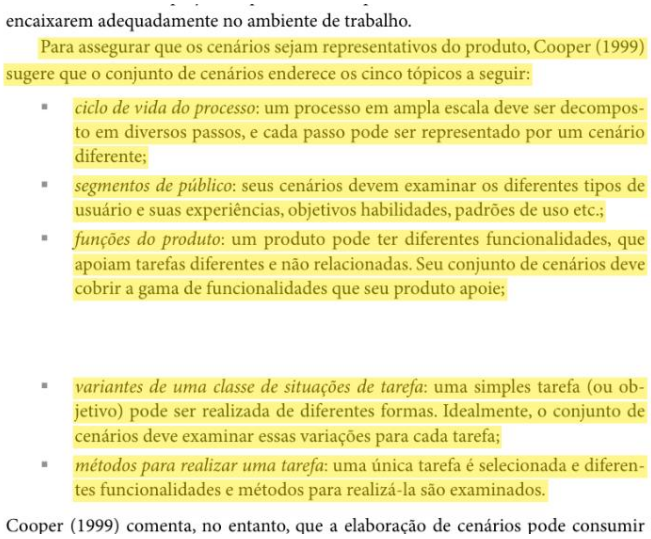
Severidade (Doolan, p. 178): ( ) Menor ( ) Maior ( ) Super-maior

Origem: Victor Andreozzi De La Rocque Couto, Lista de Verificação: Cenários, item 8.

Item 30: O conjunto de cenários é representativo do produto, cobrindo os cinco tópicos de Cooper: ciclo de vida do processo, segmentos de público, funções do produto, variantes de uma situação de tarefa e métodos para realizar uma tarefa?

Fonte: Página: 188 e 189, Capítulo 6. Organização do Espaço de Problema. Livro: SIMONE DINIZ JUNQUEIRA BARBOSA, BRUNO SANTANA DA SILVA, Interação Humano-Computador, 1ª ed., Editora Campus/Elsevier, 2010.

Foto:



encaixarem adequadamente no ambiente de trabalho.

Para assegurar que os cenários sejam representativos do produto, Cooper (1999) sugere que o conjunto de cenários enderece os cinco tópicos a seguir:

- *ciclo de vida do processo*: um processo em ampla escala deve ser decomposto em diversos passos, e cada passo pode ser representado por um cenário diferente;
- *segmentos de público*: seus cenários devem examinar os diferentes tipos de usuário e suas experiências, objetivos habilidades, padrões de uso etc.;
- *funções do produto*: um produto pode ter diferentes funcionalidades, que apoiam tarefas diferentes e não relacionadas. Seu conjunto de cenários deve cobrir a gama de funcionalidades que seu produto apoie;
- *variantes de uma classe de situações de tarefa*: uma simples tarefa (ou objetivo) pode ser realizada de diferentes formas. Idealmente, o conjunto de cenários deve examinar essas variações para cada tarefa;
- *métodos para realizar uma tarefa*: uma única tarefa é selecionada e diferentes funcionalidades e métodos para realizá-la são examinados.

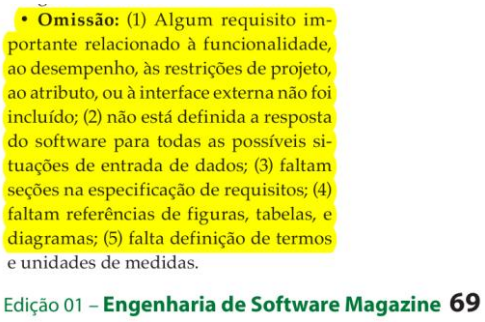
Cooper (1999) comenta, no entanto, que a elaboração de cenários pode consumir

Resposta: ( ) Conforme ( ) Não conforme ( ) Não se aplica

Critério de Fagan: se a resposta for Não conforme, o defeito é do tipo Omissão: falta no artefato um elemento importante que o item exige.

Fonte: Página: 69, Definição dos Conceitos (taxonomia de defeitos). Artigo: MARCOS KALINOWSKI, RODRIGO OLIVEIRA SPÍNOLA, Introdução à Inspeção de Software, Engenharia de Software Magazine, edição 01, p. 68 a 74.

Foto:



- **Omissão:** (1) Algum requisito importante relacionado à funcionalidade, ao desempenho, às restrições de projeto, ao atributo, ou à interface externa não foi incluído; (2) não está definida a resposta do software para todas as possíveis situações de entrada de dados; (3) faltam seções na especificação de requisitos; (4) faltam referências de figuras, tabelas, e diagramas; (5) falta definição de termos e unidades de medidas.

Edição 01 – **Engenharia de Software Magazine** **69**

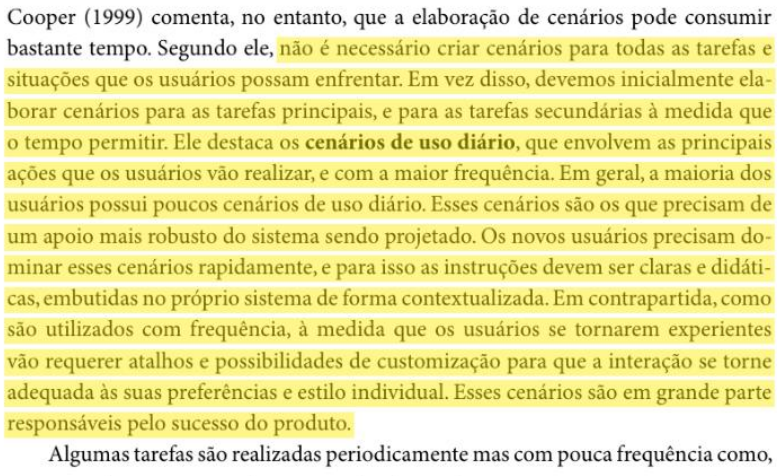
Severidade (Doolan, p. 178): ( ) Menor ( ) Maior ( ) Super-maior

Origem: Victor Andreozzi De La Rocque Couto, Lista de Verificação: Cenários, item 9.

Item 31: Os cenários de uso diário, com as tarefas principais e mais frequentes, foram elaborados com prioridade, deixando os de tarefas secundárias para quando houver tempo?

Fonte: Página: 189, Capítulo 6. Organização do Espaço de Problema. Livro: SIMONE DINIZ JUNQUEIRA BARBOSA, BRUNO SANTANA DA SILVA, Interação Humano-Computador, 1ª ed., Editora Campus/Elsevier, 2010.

Foto:



Cooper (1999) comenta, no entanto, que a elaboração de cenários pode consumir bastante tempo. Segundo ele, não é necessário criar cenários para todas as tarefas e situações que os usuários possam enfrentar. Em vez disso, devemos inicialmente elaborar cenários para as tarefas principais, e para as tarefas secundárias à medida que o tempo permitir. Ele destaca os **cenários de uso diário**, que envolvem as principais ações que os usuários vão realizar, e com a maior frequência. Em geral, a maioria dos usuários possui poucos cenários de uso diário. Esses cenários são os que precisam de um apoio mais robusto do sistema sendo projetado. Os novos usuários precisam dominar esses cenários rapidamente, e para isso as instruções devem ser claras e didáticas, embutidas no próprio sistema de forma contextualizada. Em contrapartida, como são utilizados com frequência, à medida que os usuários se tornarem experientes vão requerer atalhos e possibilidades de customização para que a interação se torne adequada às suas preferências e estilo individual. Esses cenários são em grande parte responsáveis pelo sucesso do produto.

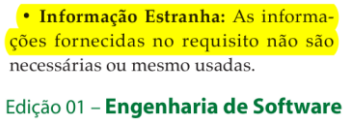
Algumas tarefas são realizadas periodicamente mas com pouca frequência como,

Resposta: ( ) Conforme ( ) Não conforme ( ) Não se aplica

Critério de Fagan: se a resposta for Não conforme, o defeito é do tipo Informação estranha: o artefato traz informação desnecessária ou que não é usada.

Fonte: Página: 69, Definição dos Conceitos (taxonomia de defeitos). Artigo: MARCOS KALINOWSKI, RODRIGO OLIVEIRA SPÍNOLA, Introdução à Inspeção de Software, Engenharia de Software Magazine, edição 01, p. 68 a 74.

Foto:



• **Informação Estranha:** As informações fornecidas no requisito não são necessárias ou mesmo usadas.

Severidade (Doolan, p. 178): ( ) Menor ( ) Maior ( ) Super-maior

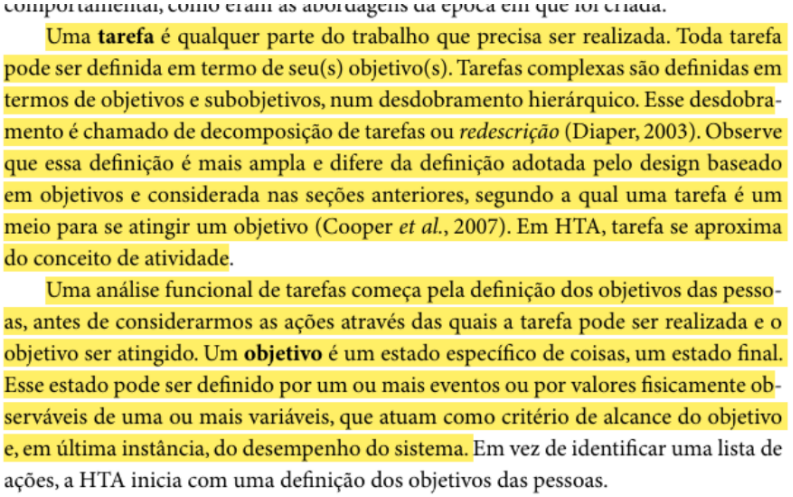
Origem: Victor Andreozzi De La Rocque Couto, Lista de Verificação: Cenários, item 10.

4.4 Análise hierárquica de tarefas (HTA)

Item 32: A HTA define o objetivo de alto nível (o estado final desejado pelo usuário) e o decompõe em subobjetivos mutuamente exclusivos e exaustivos, sem sobreposição?

Fonte: Página: 192, Capítulo 6. Organização do Espaço de Problema. Livro: SIMONE DINIZ JUNQUEIRA BARBOSA, BRUNO SANTANA DA SILVA, Interação Humano-Computador, 1ª ed., Editora Campus/Elsevier, 2010.

Foto:



comportamental, como eram as abordagens da época em que foi criada.

Uma **tarefa** é qualquer parte do trabalho que precisa ser realizada. Toda tarefa pode ser definida em termo de seu(s) objetivo(s). Tarefas complexas são definidas em termos de objetivos e subobjetivos, num desdobramento hierárquico. Esse desdobramento é chamado de decomposição de tarefas ou *redescrição* (Diaper, 2003). Observe que essa definição é mais ampla e difere da definição adotada pelo design baseado em objetivos e considerada nas seções anteriores, segundo a qual uma tarefa é um meio para se atingir um objetivo (Cooper *et al.*, 2007). Em HTA, tarefa se aproxima do conceito de atividade.

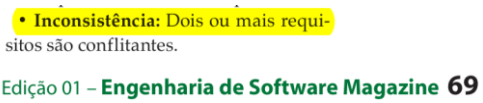
Uma análise funcional de tarefas começa pela definição dos objetivos das pessoas, antes de considerarmos as ações através das quais a tarefa pode ser realizada e o objetivo ser atingido. Um **objetivo** é um estado específico de coisas, um estado final. Esse estado pode ser definido por um ou mais eventos ou por valores fisicamente observáveis de uma ou mais variáveis, que atuam como critério de alcance do objetivo e, em última instância, do desempenho do sistema. Em vez de identificar uma lista de ações, a HTA inicia com uma definição dos objetivos das pessoas.

Resposta: ( ) Conforme ( ) Não conforme ( ) Não se aplica

Critério de Fagan: se a resposta for Não conforme, o defeito é do tipo Inconsistência: partes do artefato, ou o artefato e outro artefato do projeto, são conflitantes.

Fonte: Página: 69, Definição dos Conceitos (taxonomia de defeitos). Artigo: MARCOS KALINOWSKI, RODRIGO OLIVEIRA SPÍNOLA, Introdução à Inspeção de Software, Engenharia de Software Magazine, edição 01, p. 68 a 74.

Foto:



• **Inconsistência:** Dois ou mais requisitos são conflitantes.

Edição 01 – **Engenharia de Software Magazine** **69**

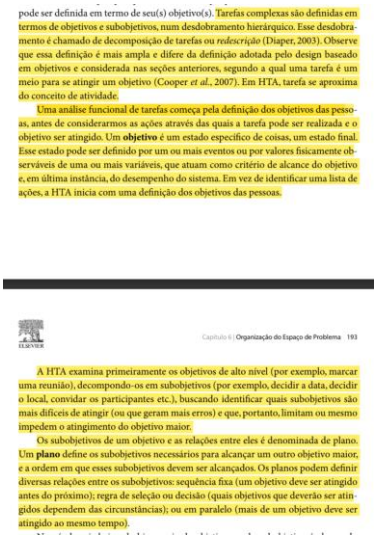
Severidade (Doolan, p. 178): ( ) Menor ( ) Maior ( ) Super-maior

Origem: Victor Andreozzi De La Rocque Couto, Lista de Verificação: HTA, item 1.

Item 33: A HTA especifica os planos que definem a ordem dos subobjetivos, indicando se há sequência fixa, regra de seleção dependente das circunstâncias ou execução em paralelo?

Fonte: Página: 192 e 193, Capítulo 6. Organização do Espaço de Problema. Livro: SIMONE DINIZ JUNQUEIRA BARBOSA, BRUNO SANTANA DA SILVA, Interação Humano-Computador, 1ª ed., Editora Campus/Elsevier, 2010.

Foto:



pode ser definida em termo de seu(s) objetivo(s). Tarefas complexas são definidas em termos de objetivos e subobjetivos, num desdobramento hierárquico. Esse desdobramento é chamado de decomposição de tarefas ou *redescrição* (Diaper, 2003). Observe que essa definição é mais ampla e difere da definição adotada pelo design baseado em objetivos e considerada nas seções anteriores, segundo a qual uma tarefa é um meio para se atingir um objetivo (Cooper *et al.*, 2007). Em HTA, tarefa se aproxima do conceito de atividade.

Uma análise funcional de tarefas começa pela definição dos objetivos das pessoas, antes de considerarmos as ações através das quais a tarefa pode ser realizada e o objetivo ser atingido. Um **objetivo** é um estado específico de coisas, um estado final. Esse estado pode ser definido por um ou mais eventos ou por valores fisicamente observáveis de uma ou mais variáveis, que atuam como critério de alcance do objetivo e, em última instância, do desempenho do sistema. Em vez de identificar uma lista de ações, a HTA inicia com uma definição dos objetivos das pessoas.

A HTA examina primeiramente os objetivos de alto nível (por exemplo, marcar uma reunião), decompondo-os em subobjetivos (por exemplo, decidir a data, decidir o local, convidar os participantes etc.), buscando identificar quais subobjetivos são mais difíceis de atingir (ou que geram mais erros) e que, portanto, limitam ou mesmo impedem o atingimento do objetivo maior.

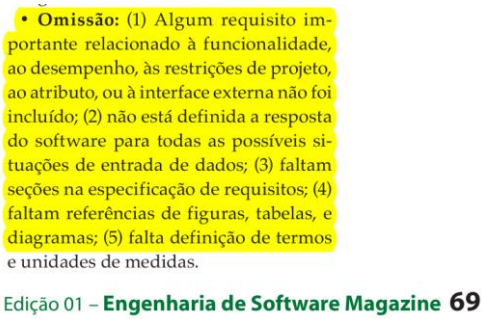
Os subobjetivos de um objetivo e as relações entre eles é denominada de plano. Um **plano** define os subobjetivos necessários para alcançar um outro objetivo maior, e a ordem em que esses subobjetivos devem ser alcançados. Os planos podem definir diversas relações entre os subobjetivos: sequência fixa (um objetivo deve ser atingido antes do próximo); regra de seleção ou decisão (quais objetivos que deverão ser atingidos dependem das circunstâncias); ou em paralelo (mais de um objetivo deve ser atingido ao mesmo tempo).

Resposta: ( ) Conforme ( ) Não conforme ( ) Não se aplica

Critério de Fagan: se a resposta for Não conforme, o defeito é do tipo Omissão: falta no artefato um elemento importante que o item exige.

Fonte: Página: 69, Definição dos Conceitos (taxonomia de defeitos). Artigo: MARCOS KALINOWSKI, RODRIGO OLIVEIRA SPÍNOLA, Introdução à Inspeção de Software, Engenharia de Software Magazine, edição 01, p. 68 a 74.

Foto:



• **Omissão:** (1) Algum requisito importante relacionado à funcionalidade, ao desempenho, às restrições de projeto, ao atributo, ou à interface externa não foi incluído; (2) não está definida a resposta do software para todas as possíveis situações de entrada de dados; (3) faltam seções na especificação de requisitos; (4) faltam referências de figuras, tabelas, e diagramas; (5) falta definição de termos e unidades de medidas.

Severidade (Doolan, p. 178): ( ) Menor ( ) Maior ( ) Super-maior

Origem: Victor Andreozzi De La Rocque Couto, Lista de Verificação: HTA, item 2.

Item 34: As operações do nível mais baixo da hierarquia estão especificadas com as circunstâncias de ativação (entradas), as ações a desempenhar e as condições de satisfação do estado final (feedback)?

Fonte: Página: 193, Capítulo 6. Organização do Espaço de Problema. Livro: SIMONE DINIZ JUNQUEIRA BARBOSA, BRUNO SANTANA DA SILVA, Interação Humano-Computador, 1ª ed., Editora Campus/Elsevier, 2010.

Foto:

No nível mais baixo da hierarquia de objetivos, cada subobjetivo é alcançado por uma **operação**, que é a unidade fundamental em HTA. A Figura 6.1 apresenta os elementos de um diagrama HTA.

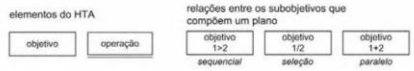


Figura 6.1 Elementos de um diagrama HTA.

Uma **operação** é especificada pelas circunstâncias nas quais o objetivo é ativado (*input* ou *entrada*), pelas atividades ou ações (*actions*) que contribuem para atingi-lo e pelas condições que indicam o seu atingimento (*feedback*). Uma **ação** pode ser entendida como uma instrução para fazer algo sob certas circunstâncias, o ***input*** como estados e o ***feedback*** como testes ou avaliação do estado final. A ação pode ser vista formalmente como uma regra de transformação entre estados. Por exemplo, em um sistema de agenda, é possível ter a seguinte tupla de <*inputs*, ações, *feedback*>: <[data, local, participantes], [convidar os participantes], [presença dos participantes confirmada]>. Em outras palavras, as principais características de uma operação são as diversas ações que devem ser desempenhadas para atingir um objetivo e as condições que o satisfazem. Dessa maneira, a análise visa identificar principalmente como um sistema possibilita ou impede as pessoas de alcançarem seus objetivos. Essa análise permite ainda identificar problemas potenciais de cada ação, bem como elaborar recomendações para evitá-los.

Resposta: ( ) Conforme ( ) Não conforme ( ) Não se aplica

Critério de Fagan: se a resposta for Não conforme, o defeito é do tipo Omissão: falta no artefato um elemento importante que o item exige.

Fonte: Página: 69, Definição dos Conceitos (taxonomia de defeitos). Artigo: MARCOS KALINOWSKI, RODRIGO OLIVEIRA SPÍNOLA, Introdução à Inspeção de Software, Engenharia de Software Magazine, edição 01, p. 68 a 74.

Foto:

• **Omissão:** (1) Algum requisito importante relacionado à funcionalidade, ao desempenho, às restrições de projeto, ao atributo, ou à interface externa não foi incluído; (2) não está definida a resposta do software para todas as possíveis situações de entrada de dados; (3) faltam seções na especificação de requisitos; (4) faltam referências de figuras, tabelas, e diagramas; (5) falta definição de termos e unidades de medidas.

Edição 01 – **Engenharia de Software Magazine** **69**

Severidade (Doolan, p. 178): ( ) Menor ( ) Maior ( ) Super-maior

Origem: Victor Andreozzi De La Rocque Couto, Lista de Verificação: HTA, item 3.

Item 35: A HTA é apresentada como diagrama hierárquico e como tabela equivalente, com colunas para objetivos ou operações, plano, entradas, ações e feedback?

Fonte: Página: 193, Capítulo 6. Organização do Espaço de Problema. Livro: SIMONE DINIZ JUNQUEIRA BARBOSA, BRUNO SANTANA DA SILVA, Interação Humano-Computador, 1ª ed., Editora Campus/Elsevier, 2010.

Foto:

No nível mais baixo da hierarquia de objetivos, cada subobjetivo é alcançado por uma **operação**, que é a unidade fundamental em HTA. A Figura 6.1 apresenta os elementos de um diagrama HTA.

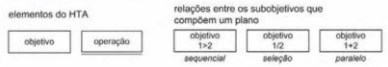


Figura 6.1 Elementos de um diagrama HTA.

Uma **operação** é especificada pelas circunstâncias nas quais o objetivo é ativado (*input* ou *entrada*), pelas atividades ou ações (*actions*) que contribuem para atingi-lo e pelas condições que indicam o seu atingimento (*feedback*). Uma **ação** pode ser entendida como uma instrução para fazer algo sob certas circunstâncias, o ***input*** como estados e o ***feedback*** como testes ou avaliação do estado final. A ação pode ser vista formalmente como uma regra de transformação entre estados. Por exemplo, em um sistema de agenda, é possível ter a seguinte tupla de <*inputs*, ações, *feedback*>: <[data, local, participantes], [convidar os participantes], [presença dos participantes confirmada]>. Em outras palavras, as principais características de uma operação são as diversas ações que devem ser desempenhadas para atingir um objetivo e as condições que o satisfazem. Dessa maneira, a análise visa identificar principalmente como um sistema possibilita ou impede as pessoas de alcançarem seus objetivos. Essa análise permite ainda identificar problemas potenciais de cada ação, bem como elaborar recomendações para evitá-los.

A Figura 6.2 ilustra um diagrama HTA para o cadastro de um projeto final em um sistema acadêmico, e a Tabela 6.3 apresenta a tabela equivalente ao diagrama.

Resposta: ( ) Conforme ( ) Não conforme ( ) Não se aplica

Critério de Fagan: se a resposta for Não conforme, o defeito é do tipo Omissão: falta no artefato um elemento importante que o item exige.

Fonte: Página: 69, Definição dos Conceitos (taxonomia de defeitos). Artigo: MARCOS KALINOWSKI, RODRIGO OLIVEIRA SPÍNOLA, Introdução à Inspeção de Software, Engenharia de Software Magazine, edição 01, p. 68 a 74.

Foto:

• **Omissão:** (1) Algum requisito importante relacionado à funcionalidade, ao desempenho, às restrições de projeto, ao atributo, ou à interface externa não foi incluído; (2) não está definida a resposta do software para todas as possíveis situações de entrada de dados; (3) faltam seções na especificação de requisitos; (4) faltam referências de figuras, tabelas, e diagramas; (5) falta definição de termos e unidades de medidas.

Edição 01 – **Engenharia de Software Magazine** **69**

Severidade (Doolan, p. 178): ( ) Menor ( ) Maior ( ) Super-maior

Origem: Victor Andreozzi De La Rocque Couto, Lista de Verificação: HTA, item 4.

Item 36: Quando a tarefa envolve mais de um agente (diferentes papéis de usuário ou usuário e sistema), a HTA separa as ações de cada agente em colunas distintas?

Fonte: Página: 191, Capítulo 6. Organização do Espaço de Problema. Livro: SIMONE DINIZ JUNQUEIRA BARBOSA, BRUNO SANTANA DA SILVA, Interação Humano-Computador, 1ª ed., Editora Campus/Elsevier, 2010.

Foto:

Em IHC, a análise de tarefas pode ser utilizada nas três atividades habituais: para análise da situação atual (apoiada ou não por um sistema computacional), para o (re)design de um sistema computacional ou para a avaliação do resultado de uma intervenção que inclua a introdução de um (novo) sistema computacional. Quando visa a avaliar um sistema computacional existente, a análise de tarefas pode ser bem concreta, descrevendo o comportamento de forma detalhada. Já numa situação de design, a análise de tarefas geralmente será realizada num nível maior de abstração, pois diversos pontos ainda não terão sido definidos no início da atividade de design.

Um dos primeiros passos numa análise de tarefas é coletarmos um conjunto de objetivos, definidos em termos psicológicos, ou seja, objetivos das pessoas. É claro que organizações também podem possuir objetivos, como, por exemplo, "aumentar o lucro", mas estes não costumam ser endereçados em análises de tarefa. Para cada objetivo, elaboramos uma lista das ações realizadas (no mundo físico e através do sistema computacional) por um agente para alcançar esse objetivo. Quando há múltiplos agentes, Diaper recomenda representar as ações de cada agente numa coluna diferente.

Resposta: ( ) Conforme ( ) Não conforme ( ) Não se aplica

Critério de Fagan: se a resposta for Não conforme, o defeito é do tipo Omissão: falta no artefato um elemento importante que o item exige.

Fonte: Página: 69, Definição dos Conceitos (taxonomia de defeitos). Artigo: MARCOS KALINOWSKI, RODRIGO OLIVEIRA SPÍNOLA, Introdução à Inspeção de Software, Engenharia de Software Magazine, edição 01, p. 68 a 74.

Foto:

• **Omissão:** (1) Algum requisito importante relacionado à funcionalidade, ao desempenho, às restrições de projeto, ao atributo, ou à interface externa não foi incluído; (2) não está definida a resposta do software para todas as possíveis situações de entrada de dados; (3) faltam seções na especificação de requisitos; (4) faltam referências de figuras, tabelas, e diagramas; (5) falta definição de termos e unidades de medidas.

Edição 01 – **Engenharia de Software Magazine** **69**

Severidade (Doolan, p. 178): ( ) Menor ( ) Maior ( ) Super-maior

Origem: Victor Andreozzi De La Rocque Couto, Lista de Verificação: HTA, item 5.

Item 37: A HTA identifica problemas ou dificuldades de desempenho em cada subobjetivo ou operação (gargalos, etapas confusas, pontos propensos a erro) e apresenta recomendações de design para evitá-los?

Fonte: Página: 193, Capítulo 6. Organização do Espaço de Problema. Livro: SIMONE DINIZ JUNQUEIRA BARBOSA, BRUNO SANTANA DA SILVA, Interação Humano-Computador, 1ª ed., Editora Campus/Elsevier, 2010.

Foto:

No nível mais baixo da hierarquia de objetivos, cada subobjetivo é alcançado por uma **operação**, que é a unidade fundamental em HTA. A Figura 6.1 apresenta os elementos de um diagrama HTA.

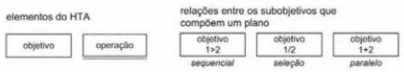


Figura 6.1 Elementos de um diagrama HTA.

Uma **operação** é especificada pelas circunstâncias nas quais o objetivo é ativado (*input* ou *entrada*), pelas atividades ou ações (*actions*) que contribuem para atingi-lo e pelas condições que indicam o seu atingimento (*feedback*). Uma **ação** pode ser entendida como uma instrução para fazer algo sob certas circunstâncias, o ***input*** como estados e o ***feedback*** como testes ou avaliação do estado final. A ação pode ser vista formalmente como uma regra de transformação entre estados. Por exemplo, em um sistema de agenda, é possível ter a seguinte tupla de <*inputs*, ações, *feedback*>: <[data, local, participantes], [convidar os participantes], [presença dos participantes confirmada]>. Em outras palavras, as principais características de uma operação são as diversas ações que devem ser desempenhadas para atingir um objetivo e as condições que o satisfazem. Dessa maneira, a análise visa identificar principalmente como um sistema possibilita ou impede as pessoas de alcançarem seus objetivos. Essa análise permite ainda identificar problemas potenciais de cada ação, bem como elaborar recomendações para evitá-los.

A Figura 6.2 ilustra um diagrama HTA para o cadastro de um projeto final em

Resposta: ( ) Conforme ( ) Não conforme ( ) Não se aplica

Critério de Fagan: se a resposta for Não conforme, o defeito é do tipo Omissão: falta no artefato um elemento importante que o item exige.

Fonte: Página: 69, Definição dos Conceitos (taxonomia de defeitos). Artigo: MARCOS KALINOWSKI, RODRIGO OLIVEIRA SPÍNOLA, Introdução à Inspeção de Software, Engenharia de Software Magazine, edição 01, p. 68 a 74.

Foto:

• **Omissão:** (1) Algum requisito importante relacionado à funcionalidade, ao desempenho, às restrições de projeto, ao atributo, ou à interface externa não foi incluído; (2) não está definida a resposta do software para todas as possíveis situações de entrada de dados; (3) faltam seções na especificação de requisitos; (4) faltam referências de figuras, tabelas, e diagramas; (5) falta definição de termos e unidades de medidas.

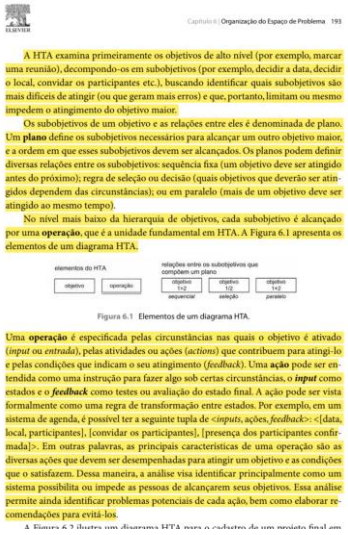
Severidade (Doolan, p. 178): ( ) Menor ( ) Maior ( ) Super-maior

Origem: Victor Andreozzi De La Rocque Couto, Lista de Verificação: HTA, item 6.

Item 38: A HTA indica que a decomposição das tarefas e os planos foram validados com as partes interessadas (usuários representativos e especialistas)?

Fonte: Página: 193, Capítulo 6. Organização do Espaço de Problema. Livro: SIMONE DINIZ JUNQUEIRA BARBOSA, BRUNO SANTANA DA SILVA, Interação Humano-Computador, 1ª ed., Editora Campus/Elsevier, 2010.

Foto:



Capítulo 6 | Organização do Espaço de Problema 193

A HTA examina primeiramente os objetivos de alto nível (por exemplo, marcar uma reunião), decompondo-os em subobjetivos (por exemplo, decidir a data, decidir o local, convidar os participantes etc.), buscando identificar quais subobjetivos são mais difíceis de atingir (ou que geram mais erros) e que, portanto, limitam ou mesmo impedem o atingimento do objetivo maior.

Os subobjetivos de um objetivo e as relações entre eles é denominada de plano. Um **plano** define os subobjetivos necessários para alcançar um outro objetivo maior, e a ordem em que esses subobjetivos devem ser alcançados. Os planos podem definir diversas relações entre os subobjetivos: sequência fixa (um objetivo deve ser atingido antes do próximo); regra de seleção ou decisão (quais objetivos que deverão ser atingidos dependem das circunstâncias); ou em paralelo (mais de um objetivo deve ser atingido ao mesmo tempo).

No nível mais baixo da hierarquia de objetivos, cada subobjetivo é alcançado por uma **operação**, que é a unidade fundamental em HTA. A Figura 6.1 apresenta os elementos de um diagrama HTA.

Figura 6.1 Elementos de um diagrama HTA.

Uma **operação** é especificada pelas circunstâncias nas quais o objetivo é ativado (*input* ou *entrada*), pelas atividades ou ações (*actions*) que contribuem para atingi-lo e pelas condições que indicam o seu atingimento (*feedback*). Uma **ação** pode ser entendida como uma instrução para fazer algo sob certas circunstâncias, o ***input*** como estados e o ***feedback*** como testes ou avaliação do estado final. A ação pode ser vista formalmente como uma regra de transformação entre estados. Por exemplo, em um sistema de agenda, é possível ter a seguinte tupla de <*inputs*, ações, *feedback*>: <[data, local, participantes], [convidar os participantes], [presença dos participantes confirmada]>. Em outras palavras, as principais características de uma operação são as diversas ações que devem ser desempenhadas para atingir um objetivo e as condições que o satisfazem. Dessa maneira, a análise visa identificar principalmente como um sistema possibilita ou impede as pessoas de alcançarem seus objetivos. Essa análise permite ainda identificar problemas potenciais de cada ação, bem como elaborar recomendações para evitá-los.

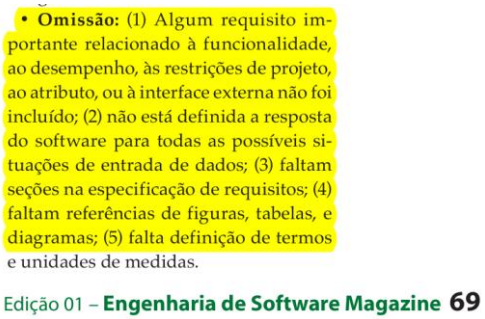
A Figura 6.2 ilustra um diagrama HTA para o cadastro de um projeto final em

Resposta: ( ) Conforme ( ) Não conforme ( ) Não se aplica

Critério de Fagan: se a resposta for Não conforme, o defeito é do tipo Omissão: falta no artefato um elemento importante que o item exige.

Fonte: Página: 69, Definição dos Conceitos (taxonomia de defeitos). Artigo: MARCOS KALINOWSKI, RODRIGO OLIVEIRA SPÍNOLA, Introdução à Inspeção de Software, Engenharia de Software Magazine, edição 01, p. 68 a 74.

Foto:



• **Omissão:** (1) Algum requisito importante relacionado à funcionalidade, ao desempenho, às restrições de projeto, ao atributo, ou à interface externa não foi incluído; (2) não está definida a resposta do software para todas as possíveis situações de entrada de dados; (3) faltam seções na especificação de requisitos; (4) faltam referências de figuras, tabelas, e diagramas; (5) falta definição de termos e unidades de medidas.

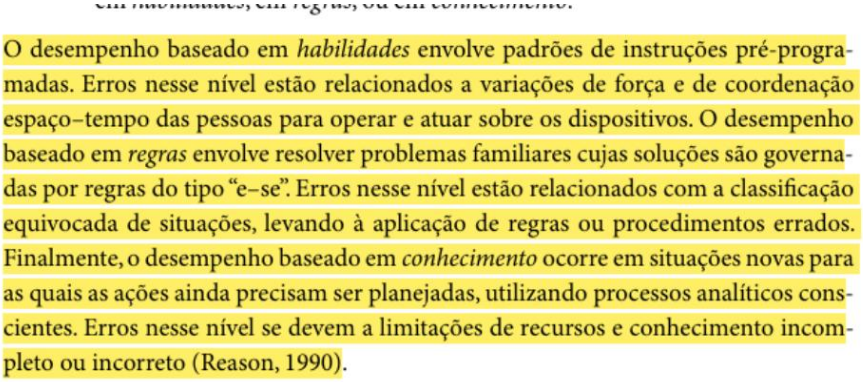
Severidade (Doolan, p. 178): ( ) Menor ( ) Maior ( ) Super-maior

Origem: Victor Andreozzi De La Rocque Couto, Lista de Verificação: HTA, item 7.

Item 39: A HTA examina e antecipa erros humanos nas operações com base na classificação de Reason (desempenho baseado em habilidades, em regras ou em conhecimento)?

Fonte: Página: 196, Capítulo 6. Organização do Espaço de Problema. Livro: SIMONE DINIZ JUNQUEIRA BARBOSA, BRUNO SANTANA DA SILVA, Interação Humano-Computador, 1ª ed., Editora Campus/Elsevier, 2010.

Foto:



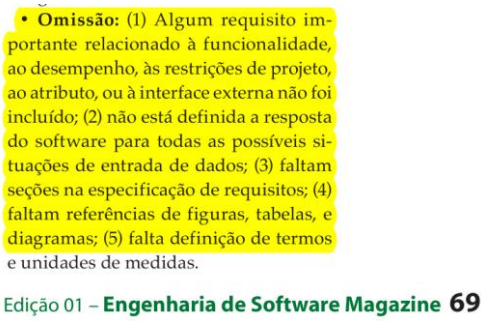
O desempenho baseado em *habilidades* envolve padrões de instruções pré-programadas. Erros nesse nível estão relacionados a variações de força e de coordenação espaço–tempo das pessoas para operar e atuar sobre os dispositivos. O desempenho baseado em *regras* envolve resolver problemas familiares cujas soluções são governadas por regras do tipo "e–se". Erros nesse nível estão relacionados com a classificação equivocada de situações, levando à aplicação de regras ou procedimentos errados. Finalmente, o desempenho baseado em *conhecimento* ocorre em situações novas para as quais as ações ainda precisam ser planejadas, utilizando processos analíticos conscientes. Erros nesse nível se devem a limitações de recursos e conhecimento incompleto ou incorreto (Reason, 1990).

Resposta: ( ) Conforme ( ) Não conforme ( ) Não se aplica

Critério de Fagan: se a resposta for Não conforme, o defeito é do tipo Omissão: falta no artefato um elemento importante que o item exige.

Fonte: Página: 69, Definição dos Conceitos (taxonomia de defeitos). Artigo: MARCOS KALINOWSKI, RODRIGO OLIVEIRA SPÍNOLA, Introdução à Inspeção de Software, Engenharia de Software Magazine, edição 01, p. 68 a 74.

Foto:



• **Omissão:** (1) Algum requisito importante relacionado à funcionalidade, ao desempenho, às restrições de projeto, ao atributo, ou à interface externa não foi incluído; (2) não está definida a resposta do software para todas as possíveis situações de entrada de dados; (3) faltam seções na especificação de requisitos; (4) faltam referências de figuras, tabelas, e diagramas; (5) falta definição de termos e unidades de medidas.

Edição 01 – **Engenharia de Software Magazine** **69**

Severidade (Doolan, p. 178): ( ) Menor ( ) Maior ( ) Super-maior

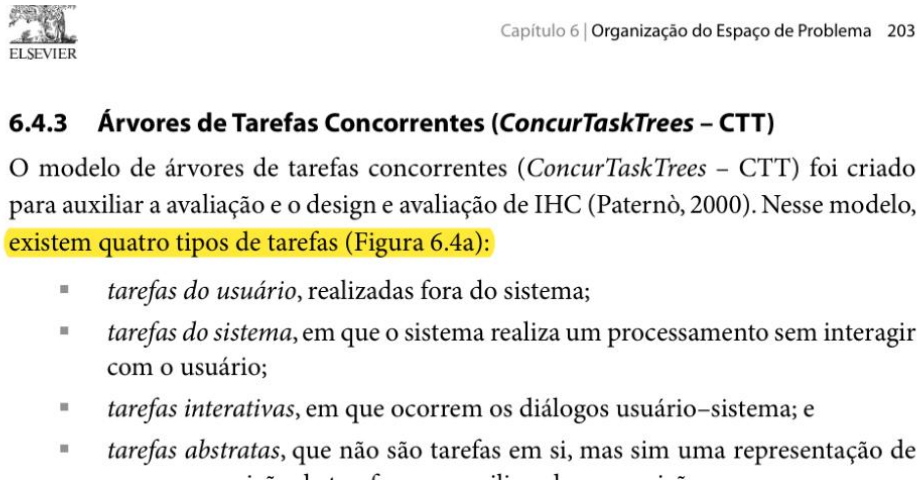
Origem: Victor Andreozzi De La Rocque Couto, Lista de Verificação: HTA, item 8.

4.5 Árvore de tarefas concorrentes (CTT)

Item 40: Cada tarefa do modelo CTT está classificada em um dos quatro tipos: tarefa do usuário (realizada fora do sistema), tarefa do sistema (processamento sem interação com o usuário), tarefa interativa (diálogo usuário-sistema) ou tarefa abstrata (composição de tarefas que auxilia a decomposição)?

Fonte: Página: 203, Capítulo 6. Organização do Espaço de Problema. Livro: SIMONE DINIZ JUNQUEIRA BARBOSA, BRUNO SANTANA DA SILVA, Interação Humano-Computador, 1ª ed., Editora Campus/Elsevier, 2010.

Foto:



ELSEVIER

Capítulo 6 | Organização do Espaço de Problema 203

6.4.3 Árvores de Tarefas Concorrentes (*ConcurTaskTrees* – CTT)

O modelo de árvores de tarefas concorrentes (*ConcurTaskTrees* – CTT) foi criado para auxiliar a avaliação e o design e avaliação de IHC (Paternò, 2000). Nesse modelo, existem quatro tipos de tarefas (Figura 6.4a):

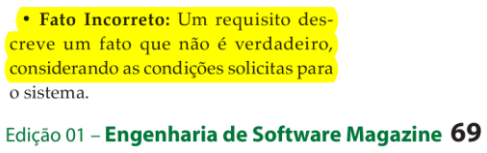
- *tarefas do usuário*, realizadas fora do sistema;
- *tarefas do sistema*, em que o sistema realiza um processamento sem interagir com o usuário;
- *tarefas interativas*, em que ocorrem os diálogos usuário–sistema; e
- *tarefas abstratas*, que não são tarefas em si, mas sim uma representação de

Resposta: ( ) Conforme ( ) Não conforme ( ) Não se aplica

Critério de Fagan: se a resposta for Não conforme, o defeito é do tipo Fato incorreto: o artefato descreve algo que não é verdadeiro nas condições do projeto.

Fonte: Página: 69, Definição dos Conceitos (taxonomia de defeitos). Artigo: MARCOS KALINOWSKI, RODRIGO OLIVEIRA SPÍNOLA, Introdução à Inspeção de Software, Engenharia de Software Magazine, edição 01, p. 68 a 74.

Foto:



• **Fato Incorreto:** Um requisito descreve um fato que não é verdadeiro, considerando as condições solicitas para o sistema.

Edição 01 – **Engenharia de Software Magazine** 69

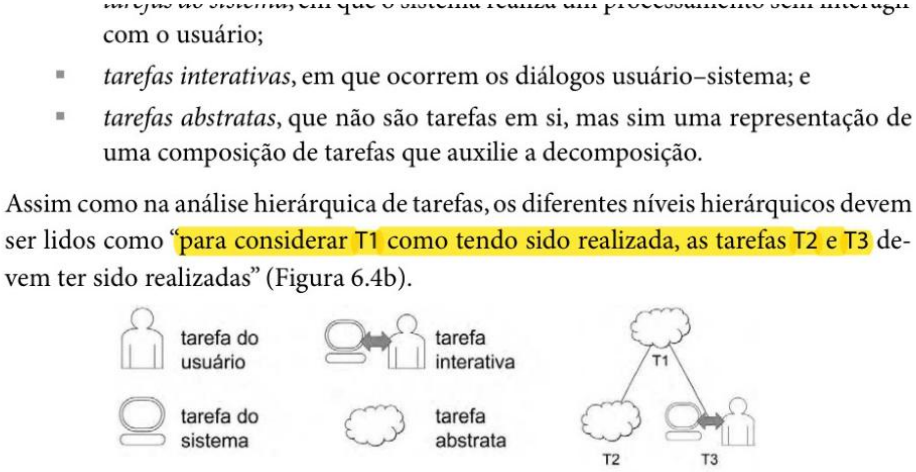
Severidade (Doolan, p. 178): ( ) Menor ( ) Maior ( ) Super-maior

Origem: Felipe Lopes Pedroza, Lista de Verificação: CTT, item 1.

Item 41: A hierarquia do diagrama CTT foi construída de modo que uma tarefa-pai só é considerada realizada quando as tarefas-filhas correspondentes tiverem sido realizadas?

Fonte: Página: 203, Capítulo 6. Organização do Espaço de Problema. Livro: SIMONE DINIZ JUNQUEIRA BARBOSA, BRUNO SANTANA DA SILVA, Interação Humano-Computador, 1ª ed., Editora Campus/Elsevier, 2010.

Foto:



com o usuário;

- *tarefas interativas*, em que ocorrem os diálogos usuário–sistema; e
- *tarefas abstratas*, que não são tarefas em si, mas sim uma representação de uma composição de tarefas que auxilie a decomposição.

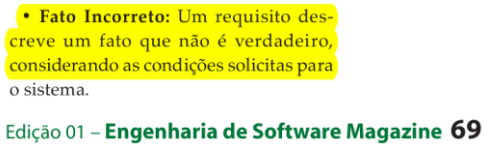
Assim como na análise hierárquica de tarefas, os diferentes níveis hierárquicos devem ser lidos como "para considerar T1 como tendo sido realizada, as tarefas T2 e T3 devem ter sido realizadas" (Figura 6.4b).

Resposta: ( ) Conforme ( ) Não conforme ( ) Não se aplica

Critério de Fagan: se a resposta for Não conforme, o defeito é do tipo Fato incorreto: o artefato descreve algo que não é verdadeiro nas condições do projeto.

Fonte: Página: 69, Definição dos Conceitos (taxonomia de defeitos). Artigo: MARCOS KALINOWSKI, RODRIGO OLIVEIRA SPÍNOLA, Introdução à Inspeção de Software, Engenharia de Software Magazine, edição 01, p. 68 a 74.

Foto:



• **Fato Incorreto:** Um requisito descreve um fato que não é verdadeiro, considerando as condições solicitas para o sistema.

Severidade (Doolan, p. 178): ( ) Menor ( ) Maior ( ) Super-maior

Origem: Felipe Lopes Pedroza, Lista de Verificação: CTT, item 2.

Item 42: A relação de ativação (T1 >> T2) foi usada corretamente para indicar que a segunda tarefa só pode iniciar após a primeira terminar?

Fonte: Página: 203, Capítulo 6. Organização do Espaço de Problema. Livro: SIMONE DINIZ JUNQUEIRA BARBOSA, BRUNO SANTANA DA SILVA, Interação Humano-Computador, 1ª ed., Editora Campus/Elsevier, 2010.

Foto:

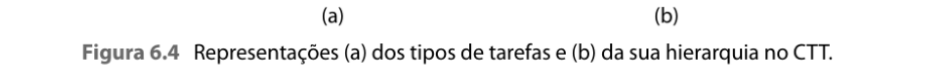


Figura 6.4 Representações (a) dos tipos de tarefas e (b) da sua hierarquia no CTT.

Além da hierarquia, o CTT permite representar diversas relações entre as tarefas, que aumentam a expressividade da notação (Figura 6.5). Os significados dssas relações são os seguintes:

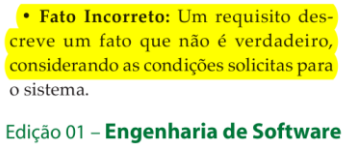
- *ativação*: T1 >> T2 significa que a segunda tarefa (T2) só pode iniciar após a primeira tarefa (T1) terminar;
- *ativação com passagem de informação*: T1 [ ] >> T2 especifica que, além de T2 só poder ser iniciada após T1, a informação produzida por T1 é passada para T2;
- *escolha* (tarefas alternativas): T1 [] T2 especifica duas tarefas que estejam habilitadas num momento, mas que, uma vez que uma delas é iniciada, a outra é desabilitada;

Resposta: ( ) Conforme ( ) Não conforme ( ) Não se aplica

Critério de Fagan: se a resposta for Não conforme, o defeito é do tipo Fato incorreto: o artefato descreve algo que não é verdadeiro nas condições do projeto.

Fonte: Página: 69, Definição dos Conceitos (taxonomia de defeitos). Artigo: MARCOS KALINOWSKI, RODRIGO OLIVEIRA SPÍNOLA, Introdução à Inspeção de Software, Engenharia de Software Magazine, edição 01, p. 68 a 74.

Foto:



• **Fato Incorreto:** Um requisito descreve um fato que não é verdadeiro, considerando as condições solicitas para o sistema.

Severidade (Doolan, p. 178): ( ) Menor ( ) Maior ( ) Super-maior

Origem: Felipe Lopes Pedroza, Lista de Verificação: CTT, item 3.

Item 43: A relação de ativação com passagem de informação (T1 []>> T2) foi usada corretamente para indicar que, além de a segunda tarefa só iniciar após a primeira, a informação produzida pela primeira é passada para a segunda?

Fonte: Página: 203, Capítulo 6. Organização do Espaço de Problema. Livro: SIMONE DINIZ JUNQUEIRA BARBOSA, BRUNO SANTANA DA SILVA, Interação Humano-Computador, 1ª ed., Editora Campus/Elsevier, 2010.

Foto:

Além da hierarquia, o CTT permite representar diversas relações entre as tarefas, que aumentam a expressividade da notação (Figura 6.5). Os significados dssas relações são os seguintes:

- *ativação*: T1 >> T2 significa que a segunda tarefa (T2) só pode iniciar após a primeira tarefa (T1) terminar;
- *ativação com passagem de informação*: T1 [ ] >> T2 especifica que, além de T2 só poder ser iniciada após T1, a informação produzida por T1 é passada para T2;
- *escolha* (tarefas alternativas): T1 [] T2 especifica duas tarefas que estejam habilitadas num momento, mas que, uma vez que uma delas é iniciada, a outra é desabilitada;
- *tarefas concorrentes*: T1 ||| T2 especifica que as tarefas podem ser realizadas em qualquer ordem ou ao mesmo tempo;

Resposta: ( ) Conforme ( ) Não conforme ( ) Não se aplica

Critério de Fagan: se a resposta for Não conforme, o defeito é do tipo Fato incorreto: o artefato descreve algo que não é verdadeiro nas condições do projeto.

Fonte: Página: 69, Definição dos Conceitos (taxonomia de defeitos). Artigo: MARCOS KALINOWSKI, RODRIGO OLIVEIRA SPÍNOLA, Introdução à Inspeção de Software, Engenharia de Software Magazine, edição 01, p. 68 a 74.

Foto:

• **Fato Incorreto:** Um requisito descreve um fato que não é verdadeiro, considerando as condições solicitas para o sistema.

Edição 01 – **Engenharia de Software Magazine** **69**

Severidade (Doolan, p. 178): ( ) Menor ( ) Maior ( ) Super-maior

Origem: Felipe Lopes Pedroza, Lista de Verificação: CTT, item 4.

Item 44: A relação de escolha (T1 [] T2) foi usada corretamente para indicar duas tarefas habilitadas ao mesmo tempo, das quais, ao ser iniciada uma, a outra é desabilitada?

Fonte: Página: 203, Capítulo 6. Organização do Espaço de Problema. Livro: SIMONE DINIZ JUNQUEIRA BARBOSA, BRUNO SANTANA DA SILVA, Interação Humano-Computador, 1ª ed., Editora Campus/Elsevier, 2010.

Foto:

são os seguintes:

- *ativação*: T1 >> T2 significa que a segunda tarefa (T2) só pode iniciar após a primeira tarefa (T1) terminar;
- *ativação com passagem de informação*: T1 [ ] >> T2 especifica que, além de T2 só poder ser iniciada após T1, a informação produzida por T1 é passada para T2;
- *escolha* (tarefas alternativas): T1 [] T2 especifica duas tarefas que estejam habilitadas num momento, mas que, uma vez que uma delas é iniciada, a outra é desabilitada;
- *tarefas concorrentes*: T1 ||| T2 especifica que as tarefas podem ser realizadas em qualquer ordem ou ao mesmo tempo;

Resposta: ( ) Conforme ( ) Não conforme ( ) Não se aplica

Critério de Fagan: se a resposta for Não conforme, o defeito é do tipo Fato incorreto: o artefato descreve algo que não é verdadeiro nas condições do projeto.

Fonte: Página: 69, Definição dos Conceitos (taxonomia de defeitos). Artigo: MARCOS KALINOWSKI, RODRIGO OLIVEIRA SPÍNOLA, Introdução à Inspeção de Software, Engenharia de Software Magazine, edição 01, p. 68 a 74.

Foto:

- **Fato Incorreto:** Um requisito descreve um fato que não é verdadeiro, considerando as condições solicitas para o sistema.

Edição 01 – **Engenharia de Software Magazine** **69**

Severidade (Doolan, p. 178): ( ) Menor ( ) Maior ( ) Super-maior

Origem: Felipe Lopes Pedroza, Lista de Verificação: CTT, item 5.

Item 45: A relação de tarefas concorrentes (T1 ||| T2) foi usada corretamente para indicar que as tarefas podem ser realizadas em qualquer ordem ou ao mesmo tempo?

Fonte: Página: 203, Capítulo 6. Organização do Espaço de Problema. Livro: SIMONE DINIZ JUNQUEIRA BARBOSA, BRUNO SANTANA DA SILVA, Interação Humano-Computador, 1ª ed., Editora Campus/Elsevier, 2010.

Foto:

- *ativação com passagem de informação*: T1 [ ] >> T2 especifica que, além de T2 só poder ser iniciada após T1, a informação produzida por T1 é passada para T2;
- *escolha* (tarefas alternativas): T1 [] T2 especifica duas tarefas que estejam habilitadas num momento, mas que, uma vez que uma delas é iniciada, a outra é desabilitada;
- *tarefas concorrentes*: T1 ||| T2 especifica que as tarefas podem ser realizadas em qualquer ordem ou ao mesmo tempo;

Resposta: ( ) Conforme ( ) Não conforme ( ) Não se aplica

Critério de Fagan: se a resposta for Não conforme, o defeito é do tipo Fato incorreto: o artefato descreve algo que não é verdadeiro nas condições do projeto.

Fonte: Página: 69, Definição dos Conceitos (taxonomia de defeitos). Artigo: MARCOS KALINOWSKI, RODRIGO OLIVEIRA SPÍNOLA, Introdução à Inspeção de Software, Engenharia de Software Magazine, edição 01, p. 68 a 74.

Foto:

• **Fato Incorreto:** Um requisito descreve um fato que não é verdadeiro, considerando as condições solicitas para o sistema.

Edição 01 – **Engenharia de Software Magazine** **69**

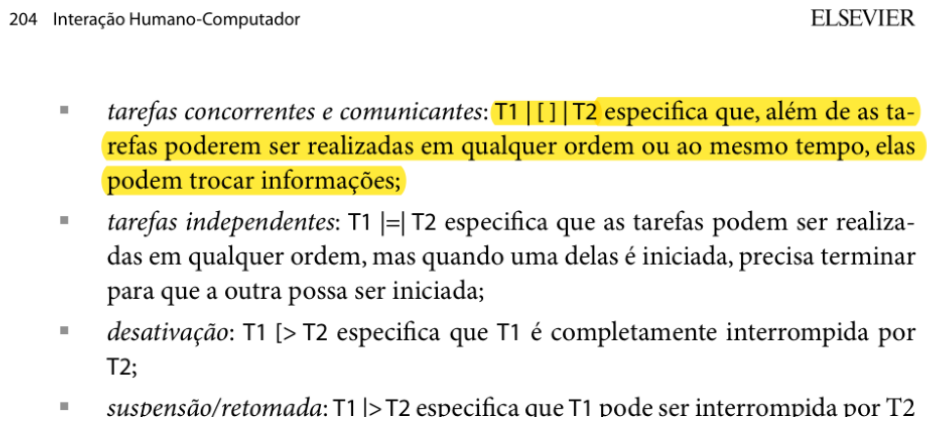
Severidade (Doolan, p. 178): ( ) Menor ( ) Maior ( ) Super-maior

Origem: Felipe Lopes Pedroza, Lista de Verificação: CTT, item 6.

Item 46: A relação de tarefas concorrentes e comunicantes (T1 |[]| T2) foi usada corretamente para indicar que, além de poderem ser realizadas em qualquer ordem ou ao mesmo tempo, as tarefas trocam informações entre si?

Fonte: Página: 204, Capítulo 6. Organização do Espaço de Problema. Livro: SIMONE DINIZ JUNQUEIRA BARBOSA, BRUNO SANTANA DA SILVA, Interação Humano-Computador, 1ª ed., Editora Campus/Elsevier, 2010.

Foto:



204 Interação Humano-Computador ELSEVIER

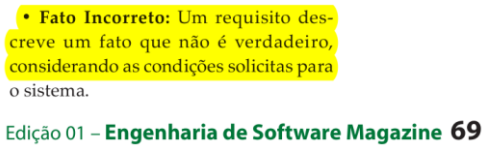
- *tarefas concorrentes e comunicantes*: T1 | [] | T2 especifica que, além de as tarefas poderem ser realizadas em qualquer ordem ou ao mesmo tempo, elas podem trocar informações;
- *tarefas independentes*: T1 |=| T2 especifica que as tarefas podem ser realizadas em qualquer ordem, mas quando uma delas é iniciada, precisa terminar para que a outra possa ser iniciada;
- *desativação*: T1 [> T2 especifica que T1 é completamente interrompida por T2;
- *suspensão/retomada*: T1 |> T2 especifica que T1 pode ser interrompida por T2

Resposta: ( ) Conforme ( ) Não conforme ( ) Não se aplica

Critério de Fagan: se a resposta for Não conforme, o defeito é do tipo Fato incorreto: o artefato descreve algo que não é verdadeiro nas condições do projeto.

Fonte: Página: 69, Definição dos Conceitos (taxonomia de defeitos). Artigo: MARCOS KALINOWSKI, RODRIGO OLIVEIRA SPÍNOLA, Introdução à Inspeção de Software, Engenharia de Software Magazine, edição 01, p. 68 a 74.

Foto:



• **Fato Incorreto:** Um requisito descreve um fato que não é verdadeiro, considerando as condições solicitas para o sistema.

Edição 01 – **Engenharia de Software Magazine** 69

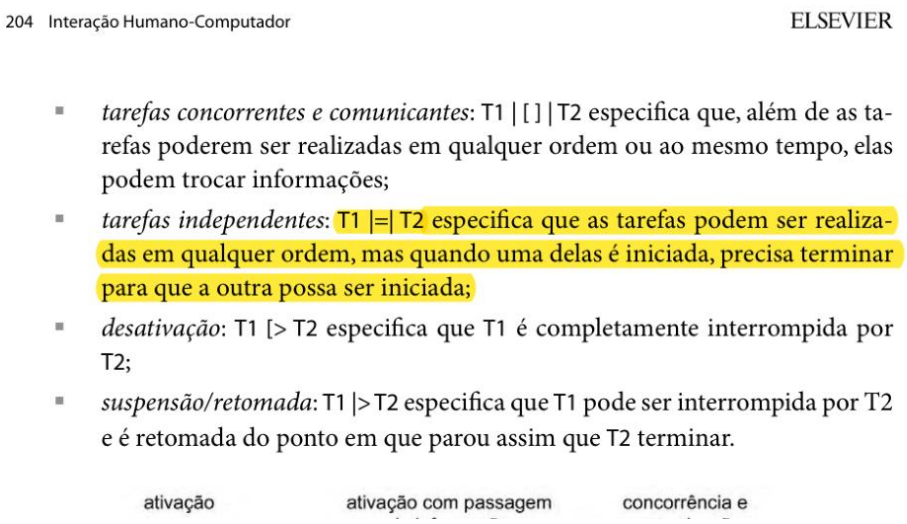
Severidade (Doolan, p. 178): ( ) Menor ( ) Maior ( ) Super-maior

Origem: Felipe Lopes Pedroza, Lista de Verificação: CTT, item 7.

Item 47: A relação de tarefas independentes (T1 |=| T2) foi usada corretamente para indicar que as tarefas podem ser realizadas em qualquer ordem, mas que, iniciada uma delas, ela precisa terminar antes que a outra comece?

Fonte: Página: 204, Capítulo 6. Organização do Espaço de Problema. Livro: SIMONE DINIZ JUNQUEIRA BARBOSA, BRUNO SANTANA DA SILVA, Interação Humano-Computador, 1ª ed., Editora Campus/Elsevier, 2010.

Foto:



204 Interação Humano-Computador ELSEVIER

- *tarefas concorrentes e comunicantes*: T1 | [ ] | T2 especifica que, além de as tarefas poderem ser realizadas em qualquer ordem ou ao mesmo tempo, elas podem trocar informações;
- *tarefas independentes*: T1 |=| T2 especifica que as tarefas podem ser realizadas em qualquer ordem, mas quando uma delas é iniciada, precisa terminar para que a outra possa ser iniciada;
- *desativação*: T1 [> T2 especifica que T1 é completamente interrompida por T2;
- *suspensão/retomada*: T1 |> T2 especifica que T1 pode ser interrompida por T2 e é retomada do ponto em que parou assim que T2 terminar.

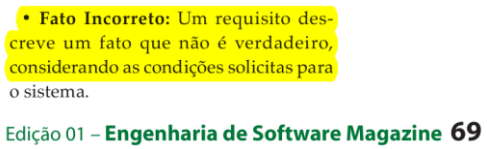
ativação ativação com passagem concorrência e

Resposta: ( ) Conforme ( ) Não conforme ( ) Não se aplica

Critério de Fagan: se a resposta for Não conforme, o defeito é do tipo Fato incorreto: o artefato descreve algo que não é verdadeiro nas condições do projeto.

Fonte: Página: 69, Definição dos Conceitos (taxonomia de defeitos). Artigo: MARCOS KALINOWSKI, RODRIGO OLIVEIRA SPÍNOLA, Introdução à Inspeção de Software, Engenharia de Software Magazine, edição 01, p. 68 a 74.

Foto:



• **Fato Incorreto:** Um requisito descreve um fato que não é verdadeiro, considerando as condições solicitas para o sistema.

Edição 01 – **Engenharia de Software Magazine** **69**

Severidade (Doolan, p. 178): ( ) Menor ( ) Maior ( ) Super-maior

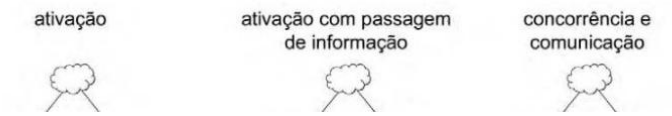
Origem: Felipe Lopes Pedroza, Lista de Verificação: CTT, item 8.

Item 48: A relação de desativação (T1 [> T2) foi usada corretamente para indicar que a primeira tarefa é completamente interrompida pela segunda?

Fonte: Página: 204, Capítulo 6. Organização do Espaço de Problema. Livro: SIMONE DINIZ JUNQUEIRA BARBOSA, BRUNO SANTANA DA SILVA, Interação Humano-Computador, 1ª ed., Editora Campus/Elsevier, 2010.

Foto:

- *tarefas concorrentes e comunicantes*: T1 | [] | T2 especifica que, além de as tarefas poderem ser realizadas em qualquer ordem ou ao mesmo tempo, elas podem trocar informações;
- *tarefas independentes*: T1 |=| T2 especifica que as tarefas podem ser realizadas em qualquer ordem, mas quando uma delas é iniciada, precisa terminar para que a outra possa ser iniciada;
- *desativação*: T1 [> T2 especifica que T1 é completamente interrompida por T2;
- *suspensão/retomada*: T1 |> T2 especifica que T1 pode ser interrompida por T2 e é retomada do ponto em que parou assim que T2 terminar.

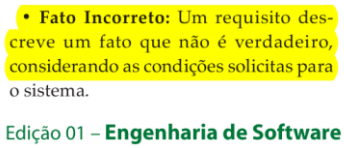


Resposta: ( ) Conforme ( ) Não conforme ( ) Não se aplica

Critério de Fagan: se a resposta for Não conforme, o defeito é do tipo Fato incorreto: o artefato descreve algo que não é verdadeiro nas condições do projeto.

Fonte: Página: 69, Definição dos Conceitos (taxonomia de defeitos). Artigo: MARCOS KALINOWSKI, RODRIGO OLIVEIRA SPÍNOLA, Introdução à Inspeção de Software, Engenharia de Software Magazine, edição 01, p. 68 a 74.

Foto:


• **Fato Incorreto:** Um requisito descreve um fato que não é verdadeiro, considerando as condições solicitas para o sistema.

Severidade (Doolan, p. 178): ( ) Menor ( ) Maior ( ) Super-maior

Origem: Felipe Lopes Pedroza, Lista de Verificação: CTT, item 9.

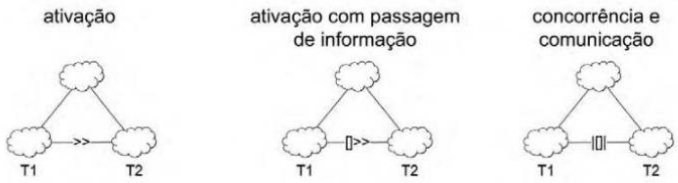
Item 49: A relação de suspensão e retomada (T1 |> T2) foi usada corretamente para indicar que a primeira tarefa pode ser interrompida pela segunda e é retomada do ponto em que parou assim que a segunda terminar?

Fonte: Página: 204, Capítulo 6. Organização do Espaço de Problema. Livro: SIMONE DINIZ JUNQUEIRA BARBOSA, BRUNO SANTANA DA SILVA, Interação Humano-Computador, 1ª ed., Editora Campus/Elsevier, 2010.

Foto:

podem trocar informações;

- *tarefas independentes*: T1 |=| T2 especifica que as tarefas podem ser realizadas em qualquer ordem, mas quando uma delas é iniciada, precisa terminar para que a outra possa ser iniciada;
- *desativação*: T1 [> T2 especifica que T1 é completamente interrompida por T2;
- *suspensão/retomada*: T1 |> T2 especifica que T1 pode ser interrompida por T2 e é retomada do ponto em que parou assim que T2 terminar.



Resposta: ( ) Conforme ( ) Não conforme ( ) Não se aplica

Critério de Fagan: se a resposta for Não conforme, o defeito é do tipo Fato incorreto: o artefato descreve algo que não é verdadeiro nas condições do projeto.

Fonte: Página: 69, Definição dos Conceitos (taxonomia de defeitos). Artigo: MARCOS KALINOWSKI, RODRIGO OLIVEIRA SPÍNOLA, Introdução à Inspeção de Software, Engenharia de Software Magazine, edição 01, p. 68 a 74.

Foto:

• **Fato Incorreto:** Um requisito descreve um fato que não é verdadeiro, considerando as condições solicitas para o sistema.

Edição 01 – **Engenharia de Software Magazine** **69**

Severidade (Doolan, p. 178): ( ) Menor ( ) Maior ( ) Super-maior

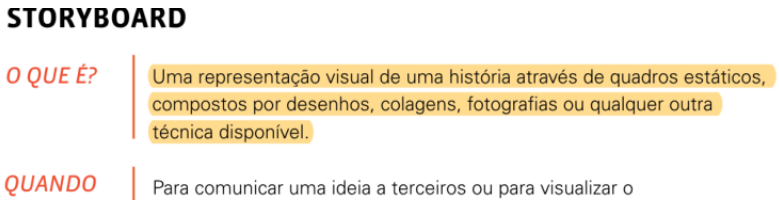
Origem: Felipe Lopes Pedroza, Lista de Verificação: CTT, item 10.

4.6 Storyboard

Item 50: O storyboard representa visualmente uma história por meio de quadros estáticos, compostos por desenhos, colagens, fotografias ou outra técnica disponível, e não apenas por texto?

Fonte: Página: 135, Capítulo Prototipação (ferramenta Storyboard). Livro: MAURÍCIO VIANNA, YSMAR VIANNA, ISABEL K. ADLER, BRENDA LUCENA, BEATRIZ RUSSO, Design Thinking: Inovação em Negócios, Editora MJV Press, 2012.

Foto:

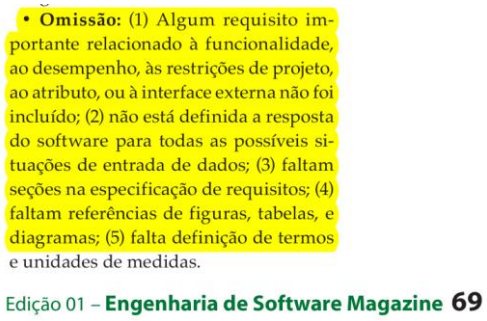


Resposta: ( ) Conforme ( ) Não conforme ( ) Não se aplica

Critério de Fagan: se a resposta for Não conforme, o defeito é do tipo Omissão: falta no artefato um elemento importante que o item exige.

Fonte: Página: 69, Definição dos Conceitos (taxonomia de defeitos). Artigo: MARCOS KALINOWSKI, RODRIGO OLIVEIRA SPÍNOLA, Introdução à Inspeção de Software, Engenharia de Software Magazine, edição 01, p. 68 a 74.

Foto:


• **Omissão:** (1) Algum requisito importante relacionado à funcionalidade, ao desempenho, às restrições de projeto, ao atributo, ou à interface externa não foi incluído; (2) não está definida a resposta do software para todas as possíveis situações de entrada de dados; (3) faltam seções na especificação de requisitos; (4) faltam referências de figuras, tabelas, e diagramas; (5) falta definição de termos e unidades de medidas.

Edição 01 – **Engenharia de Software Magazine** **69**

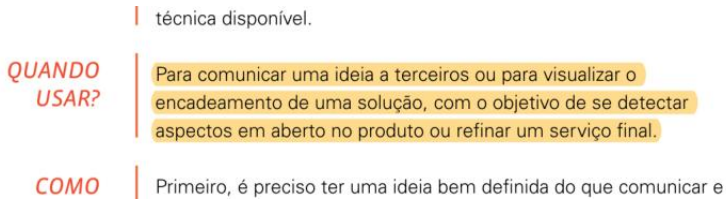
Severidade (Doolan, p. 178): ( ) Menor ( ) Maior ( ) Super-maior

Origem: Felipe Lopes Pedroza, Lista de Verificação: Storyboard, item 1.

Item 51: A finalidade do storyboard está definida como comunicar uma ideia a terceiros ou visualizar o encadeamento de uma solução, para detectar aspectos em aberto no produto ou refinar um serviço?

Fonte: Página: 135, Capítulo Prototipação (ferramenta Storyboard). Livro: MAURÍCIO VIANNA, YSMAR VIANNA, ISABEL K. ADLER, BRENDA LUCENA, BEATRIZ RUSSO, Design Thinking: Inovação em Negócios, Editora MJV Press, 2012.

Foto:


técnica disponível.

QUANDO USAR?

Para comunicar uma ideia a terceiros ou para visualizar o encadeamento de uma solução, com o objetivo de se detectar aspectos em aberto no produto ou refinar um serviço final.

COMO

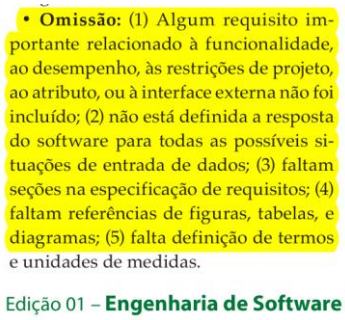
Primeiro, é preciso ter uma ideia bem definida do que comunicar e

Resposta: ( ) Conforme ( ) Não conforme ( ) Não se aplica

Critério de Fagan: se a resposta for Não conforme, o defeito é do tipo Omissão: falta no artefato um elemento importante que o item exige.

Fonte: Página: 69, Definição dos Conceitos (taxonomia de defeitos). Artigo: MARCOS KALINOWSKI, RODRIGO OLIVEIRA SPÍNOLA, Introdução à Inspeção de Software, Engenharia de Software Magazine, edição 01, p. 68 a 74.

Foto:


• **Omissão:** (1) Algum requisito importante relacionado à funcionalidade, ao desempenho, às restrições de projeto, ao atributo, ou à interface externa não foi incluído; (2) não está definida a resposta do software para todas as possíveis situações de entrada de dados; (3) faltam seções na especificação de requisitos; (4) faltam referências de figuras, tabelas, e diagramas; (5) falta definição de termos e unidades de medidas.

Edição 01 – **Engenharia de Software Magazine** **69**

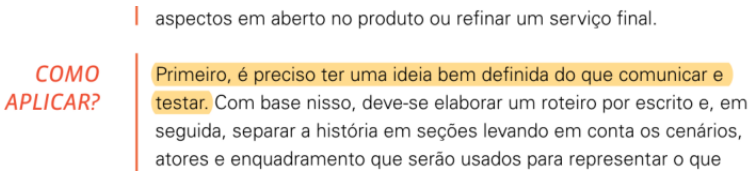
Severidade (Doolan, p. 178): ( ) Menor ( ) Maior ( ) Super-maior

Origem: Felipe Lopes Pedroza, Lista de Verificação: Storyboard, item 2.

Item 52: Há uma ideia bem definida do que se quer comunicar e testar com o storyboard (um conceito, uma funcionalidade ou um serviço)?

Fonte: Página: 135, Capítulo Prototipação (ferramenta Storyboard). Livro: MAURÍCIO VIANNA, YSMAR VIANNA, ISABEL K. ADLER, BRENDA LUCENA, BEATRIZ RUSSO, Design Thinking: Inovação em Negócios, Editora MJV Press, 2012.

Foto:



aspectos em aberto no produto ou refinar um serviço final.

COMO APLICAR?

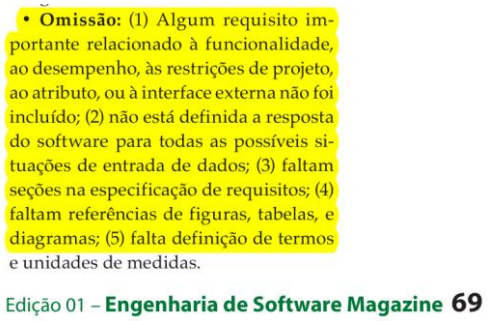
Primeiro, é preciso ter uma ideia bem definida do que comunicar e testar. Com base nisso, deve-se elaborar um roteiro por escrito e, em seguida, separar a história em seções levando em conta os cenários, atores e enquadramento que serão usados para representar o que

Resposta: ( ) Conforme ( ) Não conforme ( ) Não se aplica

Critério de Fagan: se a resposta for Não conforme, o defeito é do tipo Omissão: falta no artefato um elemento importante que o item exige.

Fonte: Página: 69, Definição dos Conceitos (taxonomia de defeitos). Artigo: MARCOS KALINOWSKI, RODRIGO OLIVEIRA SPÍNOLA, Introdução à Inspeção de Software, Engenharia de Software Magazine, edição 01, p. 68 a 74.

Foto:



• **Omissão:** (1) Algum requisito importante relacionado à funcionalidade, ao desempenho, às restrições de projeto, ao atributo, ou à interface externa não foi incluído; (2) não está definida a resposta do software para todas as possíveis situações de entrada de dados; (3) faltam seções na especificação de requisitos; (4) faltam referências de figuras, tabelas, e diagramas; (5) falta definição de termos e unidades de medidas.

Edição 01 – **Engenharia de Software Magazine** 69

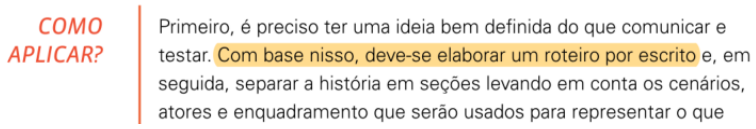
Severidade (Doolan, p. 178): ( ) Menor ( ) Maior ( ) Super-maior

Origem: Felipe Lopes Pedroza, Lista de Verificação: Storyboard, item 3.

Item 53: Foi elaborado um roteiro por escrito que descreve a história a ser representada no storyboard?

Fonte: Página: 135, Capítulo Prototipação (ferramenta Storyboard). Livro: MAURÍCIO VIANNA, YSMAR VIANNA, ISABEL K. ADLER, BRENDA LUCENA, BEATRIZ RUSSO, Design Thinking: Inovação em Negócios, Editora MJV Press, 2012.

Foto:



COMO APLICAR?

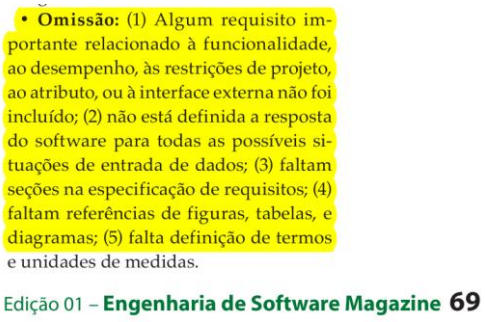
Primeiro, é preciso ter uma ideia bem definida do que comunicar e testar. Com base nisso, deve-se elaborar um roteiro por escrito e, em seguida, separar a história em seções levando em conta os cenários, atores e enquadramento que serão usados para representar o que

Resposta: ( ) Conforme ( ) Não conforme ( ) Não se aplica

Critério de Fagan: se a resposta for Não conforme, o defeito é do tipo Omissão: falta no artefato um elemento importante que o item exige.

Fonte: Página: 69, Definição dos Conceitos (taxonomia de defeitos). Artigo: MARCOS KALINOWSKI, RODRIGO OLIVEIRA SPÍNOLA, Introdução à Inspeção de Software, Engenharia de Software Magazine, edição 01, p. 68 a 74.

Foto:



• **Omissão:** (1) Algum requisito importante relacionado à funcionalidade, ao desempenho, às restrições de projeto, ao atributo, ou à interface externa não foi incluído; (2) não está definida a resposta do software para todas as possíveis situações de entrada de dados; (3) faltam seções na especificação de requisitos; (4) faltam referências de figuras, tabelas, e diagramas; (5) falta definição de termos e unidades de medidas.

Edição 01 – **Engenharia de Software Magazine** **69**

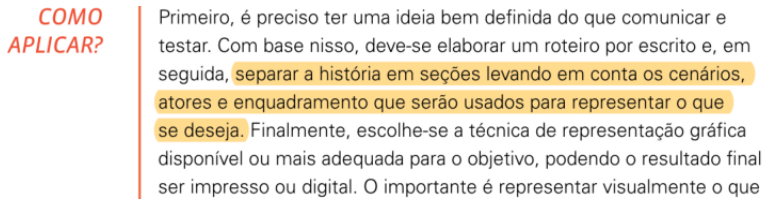
Severidade (Doolan, p. 178): ( ) Menor ( ) Maior ( ) Super-maior

Origem: Felipe Lopes Pedroza, Lista de Verificação: Storyboard, item 4.

Item 54: A história do storyboard está separada em seções que consideram os cenários (as situações em que a história acontece), os atores (quem participa) e o enquadramento (o recorte de cada cena)?

Fonte: Página: 135, Capítulo Prototipação (ferramenta Storyboard). Livro: MAURÍCIO VIANNA, YSMAR VIANNA, ISABEL K. ADLER, BRENDA LUCENA, BEATRIZ RUSSO, Design Thinking: Inovação em Negócios, Editora MJV Press, 2012.

Foto:



COMO APLICAR?

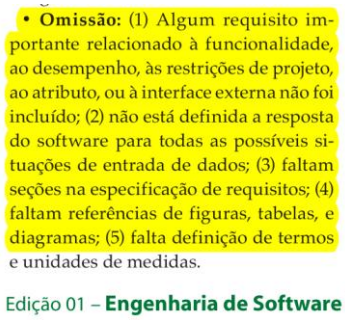
Primeiro, é preciso ter uma ideia bem definida do que comunicar e testar. Com base nisso, deve-se elaborar um roteiro por escrito e, em seguida, separar a história em seções levando em conta os cenários, atores e enquadramento que serão usados para representar o que se deseja. Finalmente, escolhe-se a técnica de representação gráfica disponível ou mais adequada para o objetivo, podendo o resultado final ser impresso ou digital. O importante é representar visualmente o que

Resposta: ( ) Conforme ( ) Não conforme ( ) Não se aplica

Critério de Fagan: se a resposta for Não conforme, o defeito é do tipo Omissão: falta no artefato um elemento importante que o item exige.

Fonte: Página: 69, Definição dos Conceitos (taxonomia de defeitos). Artigo: MARCOS KALINOWSKI, RODRIGO OLIVEIRA SPÍNOLA, Introdução à Inspeção de Software, Engenharia de Software Magazine, edição 01, p. 68 a 74.

Foto:



• **Omissão:** (1) Algum requisito importante relacionado à funcionalidade, ao desempenho, às restrições de projeto, ao atributo, ou à interface externa não foi incluído; (2) não está definida a resposta do software para todas as possíveis situações de entrada de dados; (3) faltam seções na especificação de requisitos; (4) faltam referências de figuras, tabelas, e diagramas; (5) falta definição de termos e unidades de medidas.

Edição 01 – **Engenharia de Software Magazine** **69**

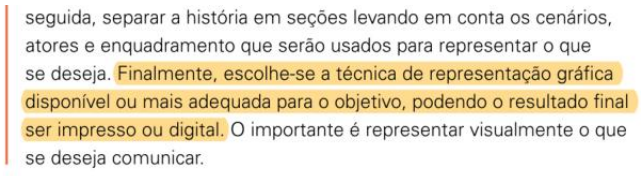
Severidade (Doolan, p. 178): ( ) Menor ( ) Maior ( ) Super-maior

Origem: Felipe Lopes Pedroza, Lista de Verificação: Storyboard, item 5.

Item 55: A técnica de representação gráfica (desenho, colagem ou fotografia, impressa ou digital) é a disponível ou a mais adequada ao objetivo do storyboard?

Fonte: Página: 135, Capítulo Prototipação (ferramenta Storyboard). Livro: MAURÍCIO VIANNA, YSMAR VIANNA, ISABEL K. ADLER, BRENDA LUCENA, BEATRIZ RUSSO, Design Thinking: Inovação em Negócios, Editora MJV Press, 2012.

Foto:



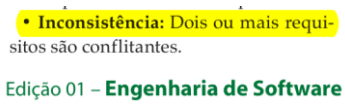
seguida, separar a história em seções levando em conta os cenários, atores e enquadramento que serão usados para representar o que se deseja. Finalmente, escolhe-se a técnica de representação gráfica disponível ou mais adequada para o objetivo, podendo o resultado final ser impresso ou digital. O importante é representar visualmente o que se deseja comunicar.

Resposta: ( ) Conforme ( ) Não conforme ( ) Não se aplica

Critério de Fagan: se a resposta for Não conforme, o defeito é do tipo Inconsistência: partes do artefato, ou o artefato e outro artefato do projeto, são conflitantes.

Fonte: Página: 69, Definição dos Conceitos (taxonomia de defeitos). Artigo: MARCOS KALINOWSKI, RODRIGO OLIVEIRA SPÍNOLA, Introdução à Inspeção de Software, Engenharia de Software Magazine, edição 01, p. 68 a 74.

Foto:



• **Inconsistência:** Dois ou mais requisitos são conflitantes.

Edição 01 – **Engenharia de Software Magazine** **69**

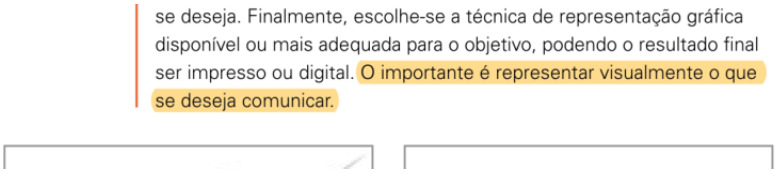
Severidade (Doolan, p. 178): ( ) Menor ( ) Maior ( ) Super-maior

Origem: Felipe Lopes Pedroza, Lista de Verificação: Storyboard, item 6.

Item 56: Os quadros representam visualmente o que se deseja comunicar, de modo que quem os observa entenda a ideia, a solução ou o serviço apresentado?

Fonte: Página: 135, Capítulo Prototipação (ferramenta Storyboard). Livro: MAURÍCIO VIANNA, YSMAR VIANNA, ISABEL K. ADLER, BRENDA LUCENA, BEATRIZ RUSSO, Design Thinking: Inovação em Negócios, Editora MJV Press, 2012.

Foto:



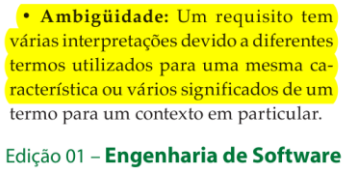
se deseja. Finalmente, escolhe-se a técnica de representação gráfica disponível ou mais adequada para o objetivo, podendo o resultado final ser impresso ou digital. O importante é representar visualmente o que se deseja comunicar.

Resposta: ( ) Conforme ( ) Não conforme ( ) Não se aplica

Critério de Fagan: se a resposta for Não conforme, o defeito é do tipo Ambiguidade: o artefato admite mais de uma interpretação.

Fonte: Página: 69, Definição dos Conceitos (taxonomia de defeitos). Artigo: MARCOS KALINOWSKI, RODRIGO OLIVEIRA SPÍNOLA, Introdução à Inspeção de Software, Engenharia de Software Magazine, edição 01, p. 68 a 74.

Foto:



• **Ambigüidade:** Um requisito tem várias interpretações devido a diferentes termos utilizados para uma mesma característica ou vários significados de um termo para um contexto em particular.

Edição 01 – **Engenharia de Software Magazine** **69**

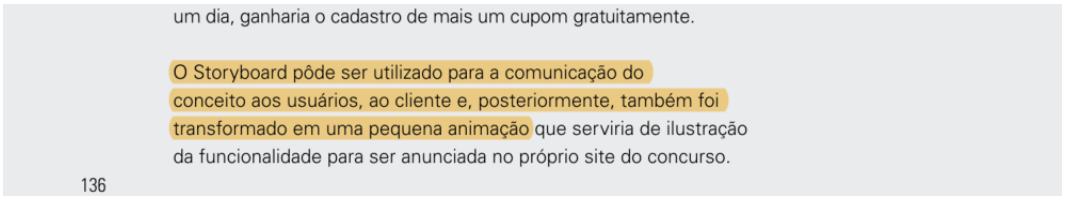
Severidade (Doolan, p. 178): ( ) Menor ( ) Maior ( ) Super-maior

Origem: Felipe Lopes Pedroza, Lista de Verificação: Storyboard, item 7.

Item 57: O storyboard comunica o conceito (por exemplo, uma nova funcionalidade) aos usuários e ao cliente, podendo ainda ser transformado em uma pequena animação?

Fonte: Página: 136, Capítulo Prototipação (ferramenta Storyboard). Livro: MAURÍCIO VIANNA, YSMAR VIANNA, ISABEL K. ADLER, BRENDA LUCENA, BEATRIZ RUSSO, Design Thinking: Inovação em Negócios, Editora MJV Press, 2012.

Foto:



um dia, ganharia o cadastro de mais um cupom gratuitamente.

O Storyboard pôde ser utilizado para a comunicação do conceito aos usuários, ao cliente e, posteriormente, também foi transformado em uma pequena animação que serviria de ilustração da funcionalidade para ser anunciada no próprio site do concurso.

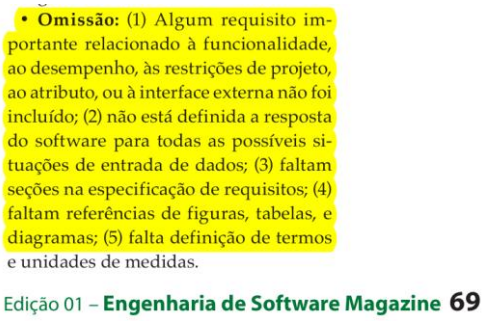
136

Resposta: ( ) Conforme ( ) Não conforme ( ) Não se aplica

Critério de Fagan: se a resposta for Não conforme, o defeito é do tipo Omissão: falta no artefato um elemento importante que o item exige.

Fonte: Página: 69, Definição dos Conceitos (taxonomia de defeitos). Artigo: MARCOS KALINOWSKI, RODRIGO OLIVEIRA SPÍNOLA, Introdução à Inspeção de Software, Engenharia de Software Magazine, edição 01, p. 68 a 74.

Foto:



• **Omissão:** (1) Algum requisito importante relacionado à funcionalidade, ao desempenho, às restrições de projeto, ao atributo, ou à interface externa não foi incluído; (2) não está definida a resposta do software para todas as possíveis situações de entrada de dados; (3) faltam seções na especificação de requisitos; (4) faltam referências de figuras, tabelas, e diagramas; (5) falta definição de termos e unidades de medidas.

Edição 01 – **Engenharia de Software Magazine** **69**

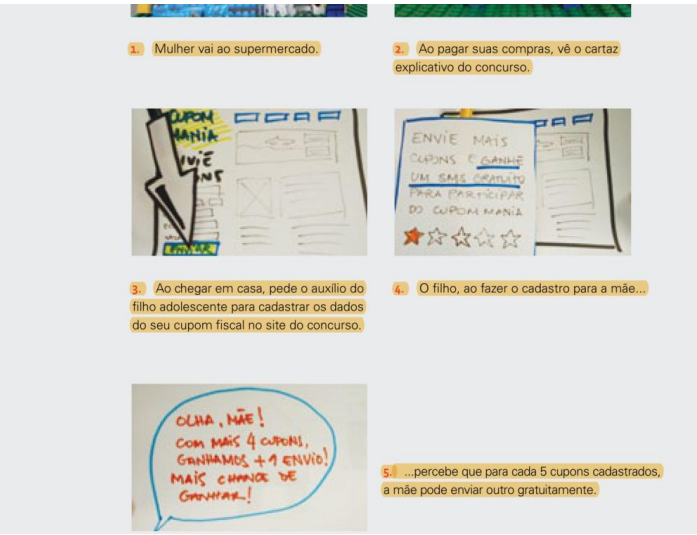
Severidade (Doolan, p. 178): ( ) Menor ( ) Maior ( ) Super-maior

Origem: Felipe Lopes Pedroza, Lista de Verificação: Storyboard, item 8.

Item 58: Os quadros estão numerados em sequência e cada quadro traz uma legenda curta que descreve a ação mostrada?

Fonte: Página: 137, Capítulo Prototipação (ferramenta Storyboard). Livro: MAURÍCIO VIANNA, YSMAR VIANNA, ISABEL K. ADLER, BRENDA LUCENA, BEATRIZ RUSSO, Design Thinking: Inovação em Negócios, Editora MJV Press, 2012.

Foto:

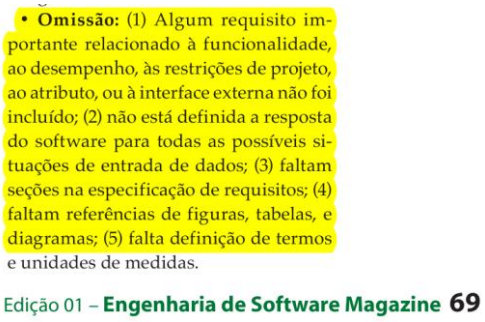


Resposta: ( ) Conforme ( ) Não conforme ( ) Não se aplica

Critério de Fagan: se a resposta for Não conforme, o defeito é do tipo Omissão: falta no artefato um elemento importante que o item exige.

Fonte: Página: 69, Definição dos Conceitos (taxonomia de defeitos). Artigo: MARCOS KALINOWSKI, RODRIGO OLIVEIRA SPÍNOLA, Introdução à Inspeção de Software, Engenharia de Software Magazine, edição 01, p. 68 a 74.

Foto:


• **Omissão:** (1) Algum requisito importante relacionado à funcionalidade, ao desempenho, às restrições de projeto, ao atributo, ou à interface externa não foi incluído; (2) não está definida a resposta do software para todas as possíveis situações de entrada de dados; (3) faltam seções na especificação de requisitos; (4) faltam referências de figuras, tabelas, e diagramas; (5) falta definição de termos e unidades de medidas.

Edição 01 – **Engenharia de Software Magazine** **69**

Severidade (Doolan, p. 178): ( ) Menor ( ) Maior ( ) Super-maior

Origem: Felipe Lopes Pedroza, Lista de Verificação: Storyboard, item 9.

Item 59: O storyboard é usado para explorar ideias sobre como melhorar a prática de trabalho dos usuários com o apoio da tecnologia, a partir dos modelos de trabalho consolidados?

Fonte: Página: 112, Capítulo 4. Processos de Design de IHC (Seção 4.3.4 Design Contextual). Livro: SIMONE DINIZ JUNQUEIRA BARBOSA, BRUNO SANTANA DA SILVA, Interação Humano-Computador, 1ª ed., Editora Campus/Elsevier, 2010.

Foto:

brainstorming.

A consolidação dos modelos de trabalho fornece insumos para o designer **re-projetar** a forma como os usuários trabalham. O designer utiliza *storyboards* para explorar ideias sobre como melhorar a prática de trabalho com o suporte oferecido pela tecnologia.

Uma vez concebida uma nova maneira de trabalhar, o designer segue **projetando uma solução de interação e de interface** que apoie essa nova forma de trabalhar.

Resposta: ( ) Conforme ( ) Não conforme ( ) Não se aplica

Critério de Fagan: se a resposta for Não conforme, o defeito é do tipo Omissão: falta no artefato um elemento importante que o item exige.

Fonte: Página: 69, Definição dos Conceitos (taxonomia de defeitos). Artigo: MARCOS KALINOWSKI, RODRIGO OLIVEIRA SPÍNOLA, Introdução à Inspeção de Software, Engenharia de Software Magazine, edição 01, p. 68 a 74.

Foto:

• **Omissão:** (1) Algum requisito importante relacionado à funcionalidade, ao desempenho, às restrições de projeto, ao atributo, ou à interface externa não foi incluído; (2) não está definida a resposta do software para todas as possíveis situações de entrada de dados; (3) faltam seções na especificação de requisitos; (4) faltam referências de figuras, tabelas, e diagramas; (5) falta definição de termos e unidades de medidas.

Edição 01 – **Engenharia de Software Magazine** **69**

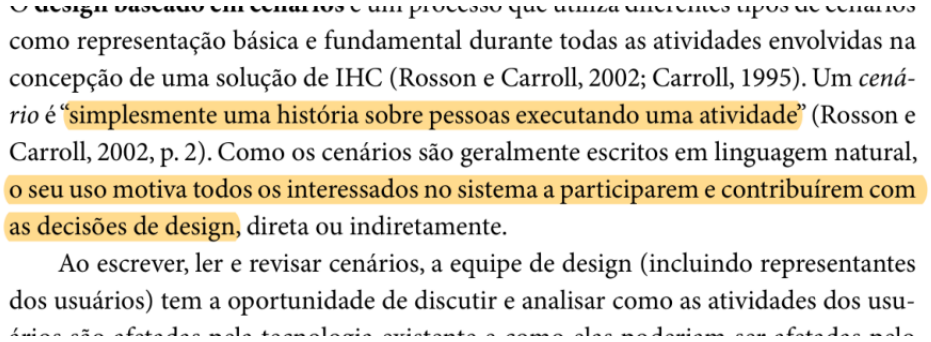
Severidade (Doolan, p. 178): ( ) Menor ( ) Maior ( ) Super-maior

Origem: Felipe Lopes Pedroza, Lista de Verificação: Storyboard, item 10.

Item 60: A história contada pelo storyboard é um cenário, isto é, uma história sobre pessoas executando uma atividade, escrita de forma que os interessados (incluindo os usuários) a entendam e contribuam com as decisões de design?

Fonte: Página: 112, Capítulo 4. Processos de Design de IHC (Seção 4.3.5 Design Baseado em Cenários). Livro: SIMONE DINIZ JUNQUEIRA BARBOSA, BRUNO SANTANA DA SILVA, Interação Humano-Computador, 1ª ed., Editora Campus/Elsevier, 2010.

Foto:



como representação básica e fundamental durante todas as atividades envolvidas na concepção de uma solução de IHC (Rosson e Carroll, 2002; Carroll, 1995). Um *cenário* é "simplesmente uma história sobre pessoas executando uma atividade" (Rosson e Carroll, 2002, p. 2). Como os cenários são geralmente escritos em linguagem natural, o seu uso motiva todos os interessados no sistema a participarem e contribuírem com as decisões de design, direta ou indiretamente.

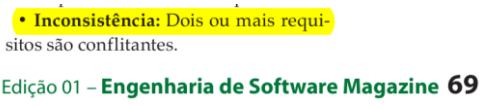
Ao escrever, ler e revisar cenários, a equipe de design (incluindo representantes dos usuários) tem a oportunidade de discutir e analisar como as atividades dos usu-

Resposta: ( ) Conforme ( ) Não conforme ( ) Não se aplica

Critério de Fagan: se a resposta for Não conforme, o defeito é do tipo Inconsistência: partes do artefato, ou o artefato e outro artefato do projeto, são conflitantes.

Fonte: Página: 69, Definição dos Conceitos (taxonomia de defeitos). Artigo: MARCOS KALINOWSKI, RODRIGO OLIVEIRA SPÍNOLA, Introdução à Inspeção de Software, Engenharia de Software Magazine, edição 01, p. 68 a 74.

Foto:



• **Inconsistência:** Dois ou mais requisitos são conflitantes.

Edição 01 – **Engenharia de Software Magazine** **69**

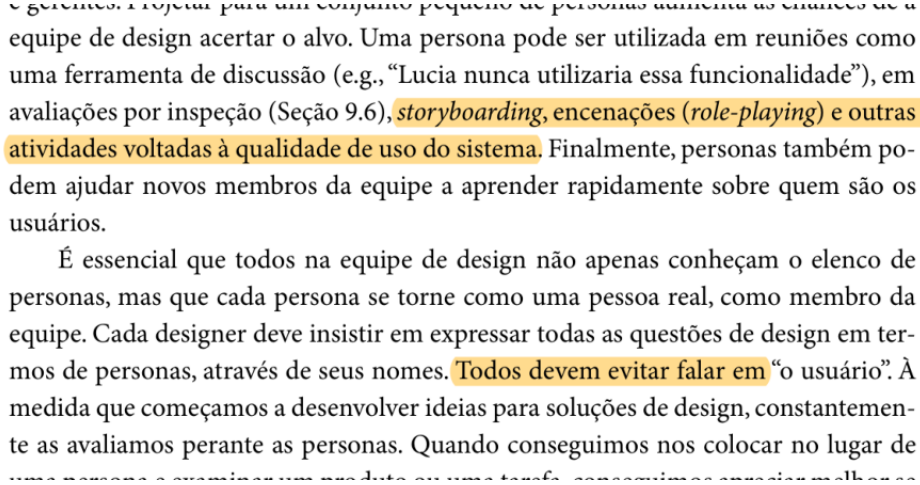
Severidade (Doolan, p. 178): ( ) Menor ( ) Maior ( ) Super-maior

Origem: Felipe Lopes Pedroza, Lista de Verificação: Storyboard, item 11.

Item 61: Os personagens do storyboard são personas do projeto (usuários arquétipos com nome, habilidades, motivações e objetivos), identificadas pelo nome e não pela expressão genérica “o usuário”?

Fonte: Página: 179, Capítulo 6. Organização do Espaço de Problema. Livro: SIMONE DINIZ JUNQUEIRA BARBOSA, BRUNO SANTANA DA SILVA, Interação Humano-Computador, 1ª ed., Editora Campus/Elsevier, 2010.

Foto:



e gerentes. Projetar para um conjunto pequeno de personas aumenta as chances de a equipe de design acertar o alvo. Uma persona pode ser utilizada em reuniões como uma ferramenta de discussão (e.g., “Lucia nunca utilizaria essa funcionalidade”), em avaliações por inspeção (Seção 9.6), *storyboarding*, encenações (*role-playing*) e outras atividades voltadas à qualidade de uso do sistema. Finalmente, personas também podem ajudar novos membros da equipe a aprender rapidamente sobre quem são os usuários.

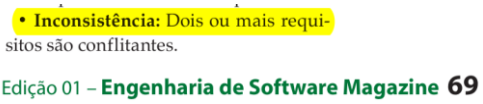
É essencial que todos na equipe de design não apenas conheçam o elenco de personas, mas que cada persona se torne como uma pessoa real, como membro da equipe. Cada designer deve insistir em expressar todas as questões de design em termos de personas, através de seus nomes. Todos devem evitar falar em “o usuário”. À medida que começamos a desenvolver ideias para soluções de design, constantemente as avaliamos perante as personas. Quando conseguimos nos colocar no lugar de uma persona e examinar um produto ou uma tarefa, conseguimos apreciar melhor se

Resposta: ( ) Conforme ( ) Não conforme ( ) Não se aplica

Critério de Fagan: se a resposta for Não conforme, o defeito é do tipo Inconsistência: partes do artefato, ou o artefato e outro artefato do projeto, são conflitantes.

Fonte: Página: 69, Definição dos Conceitos (taxonomia de defeitos). Artigo: MARCOS KALINOWSKI, RODRIGO OLIVEIRA SPÍNOLA, Introdução à Inspeção de Software, Engenharia de Software Magazine, edição 01, p. 68 a 74.

Foto:



• **Inconsistência:** Dois ou mais requisitos são conflitantes.

Edição 01 – **Engenharia de Software Magazine** **69**

Severidade (Doolan, p. 178): ( ) Menor ( ) Maior ( ) Super-maior

Origem: Felipe Lopes Pedroza, Lista de Verificação: Storyboard, item 12.

Item 62: Os cenários de interação que orientam o storyboard especificam as ações do usuário e as respostas (feedback) do sistema, sem incluir, nas fases iniciais, detalhes da interface como textos, rótulos ou tipos de elemento de interface?

Fonte: Página: 209, Capítulo 7. Design de IHC (Seção 7.2 Cenários de Interação). Livro: SIMONE DINIZ JUNQUEIRA BARBOSA, BRUNO SANTANA DA SILVA, Interação Humano-Computador, 1ª ed., Editora Campus/Elsevier, 2010.

Foto:

de problemas investigados na atividade de análise descrevem a situação atual, evidenciando problemas e oportunidades de melhoria. Já os cenários de interação representam intervenções para endereçar esses problemas e oportunidades.

Um **cenário de interação** especifica em mais detalhes as ações do usuário e as respectivas respostas (*feedback*) do sistema necessárias para alcançar os objetivos apoiados pelo sistema. Apesar de ricos em detalhes e contextualizados, os cenários de interação elaborados em fases iniciais de design não devem conter detalhes da *interface* propriamente dita, como textos e rótulos, tipos de elementos de interface (*widgets*) utilizados etc. Pretendemos com isso evitar um comprometimento precoce dos designers com uma solução de interface a ser adotada, o que dificultaria a exploração de soluções alternativas que emergissem da modelagem de tarefas e do projeto cuidadoso da interação. Por exemplo, não devemos dizer que "Joana seleciona

Resposta: ( ) Conforme ( ) Não conforme ( ) Não se aplica

Critério de Fagan: se a resposta for Não conforme, o defeito é do tipo Informação estranha: o artefato traz informação desnecessária ou que não é usada.

Fonte: Página: 69, Definição dos Conceitos (taxonomia de defeitos). Artigo: MARCOS KALINOWSKI, RODRIGO OLIVEIRA SPÍNOLA, Introdução à Inspeção de Software, Engenharia de Software Magazine, edição 01, p. 68 a 74.

Foto:

• **Informação Estranha:** As informações fornecidas no requisito não são necessárias ou mesmo usadas.

Edição 01 – **Engenharia de Software Magazine** **69**

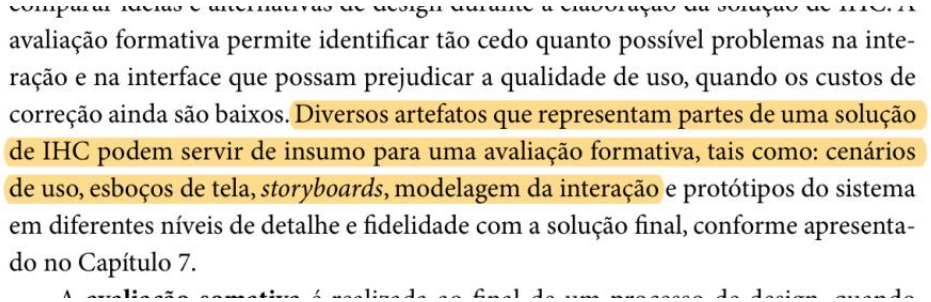
Severidade (Doolan, p. 178): ( ) Menor ( ) Maior ( ) Super-maior

Origem: Felipe Lopes Pedroza, Lista de Verificação: Storyboard, item 13.

Item 63: O storyboard pode ser usado como insumo de uma avaliação formativa (feita durante a elaboração da solução), junto com cenários, esboços de tela, modelagem da interação e protótipos?

Fonte: Página: 294, Capítulo 9. Planejamento da Avaliação de IHC (Seção 9.3 Quando Avaliar o Uso de um Sistema?). Livro: SIMONE DINIZ JUNQUEIRA BARBOSA, BRUNO SANTANA DA SILVA, Interação Humano-Computador, 1ª ed., Editora Campus/Elsevier, 2010.

Foto:



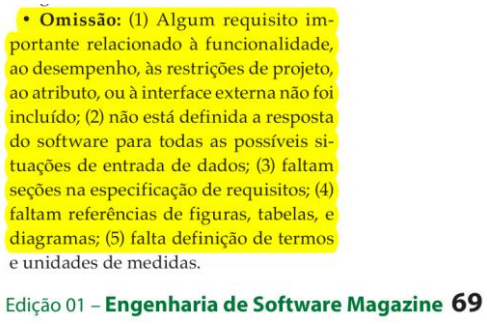
avaliação formativa permite identificar tão cedo quanto possível problemas na interação e na interface que possam prejudicar a qualidade de uso, quando os custos de correção ainda são baixos. Diversos artefatos que representam partes de uma solução de IHC podem servir de insumo para uma avaliação formativa, tais como: cenários de uso, esboços de tela, *storyboards*, modelagem da interação e protótipos do sistema em diferentes níveis de detalhe e fidelidade com a solução final, conforme apresentado no Capítulo 7.

Resposta: ( ) Conforme ( ) Não conforme ( ) Não se aplica

Critério de Fagan: se a resposta for Não conforme, o defeito é do tipo Omissão: falta no artefato um elemento importante que o item exige.

Fonte: Página: 69, Definição dos Conceitos (taxonomia de defeitos). Artigo: MARCOS KALINOWSKI, RODRIGO OLIVEIRA SPÍNOLA, Introdução à Inspeção de Software, Engenharia de Software Magazine, edição 01, p. 68 a 74.

Foto:



• **Omissão:** (1) Algum requisito importante relacionado à funcionalidade, ao desempenho, às restrições de projeto, ao atributo, ou à interface externa não foi incluído; (2) não está definida a resposta do software para todas as possíveis situações de entrada de dados; (3) faltam seções na especificação de requisitos; (4) faltam referências de figuras, tabelas, e diagramas; (5) falta definição de termos e unidades de medidas.

Edição 01 – **Engenharia de Software Magazine** **69**

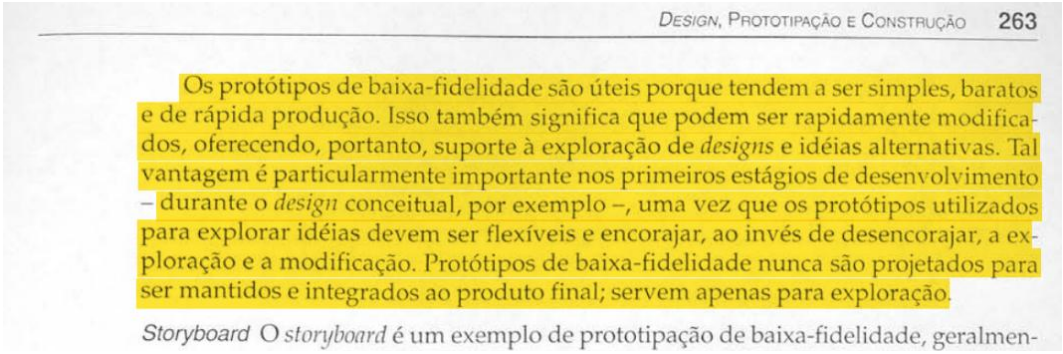
Severidade (Doolan, p. 178): ( ) Menor ( ) Maior ( ) Super-maior

Origem: Felipe Lopes Pedroza, Lista de Verificação: Storyboard, item 14.

Item 64: O storyboard foi feito como protótipo de baixa fidelidade simples, barato, rápido e fácil de modificar, para explorar ideias alternativas, e não como algo a ser mantido e integrado ao produto final?

Fonte: Página: 263, Capítulo 8. Design, Prototipação e Construção (Seção 8.2.3 Prototipação de baixa-fidelidade). Livro: JENNIFER PREECE, YVONNE ROGERS, HELEN SHARP, Design de Interação: além da interação homem-computador, Editora Bookman, 2005.

Foto:



DESIGN, PROTOTIPAÇÃO E CONSTRUÇÃO 263

Os protótipos de baixa-fidelidade são úteis porque tendem a ser simples, baratos e de rápida produção. Isso também significa que podem ser rapidamente modificados, oferecendo, portanto, suporte à exploração de *designs* e idéias alternativas. Tal vantagem é particularmente importante nos primeiros estágios de desenvolvimento – durante o *design* conceitual, por exemplo –, uma vez que os protótipos utilizados para explorar idéias devem ser flexíveis e encorajar, ao invés de desencorajar, a exploração e a modificação. Protótipos de baixa-fidelidade nunca são projetados para ser mantidos e integrados ao produto final; servem apenas para exploração.

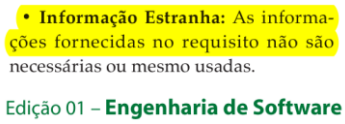
Storyboard O *storyboard* é um exemplo de prototipação de baixa-fidelidade, geralmen-

Resposta: ( ) Conforme ( ) Não conforme ( ) Não se aplica

Critério de Fagan: se a resposta for Não conforme, o defeito é do tipo Informação estranha: o artefato traz informação desnecessária ou que não é usada.

Fonte: Página: 69, Definição dos Conceitos (taxonomia de defeitos). Artigo: MARCOS KALINOWSKI, RODRIGO OLIVEIRA SPÍNOLA, Introdução à Inspeção de Software, Engenharia de Software Magazine, edição 01, p. 68 a 74.

Foto:



• **Informação Estranha:** As informações fornecidas no requisito não são necessárias ou mesmo usadas.

Edição 01 – **Engenharia de Software Magazine** 69

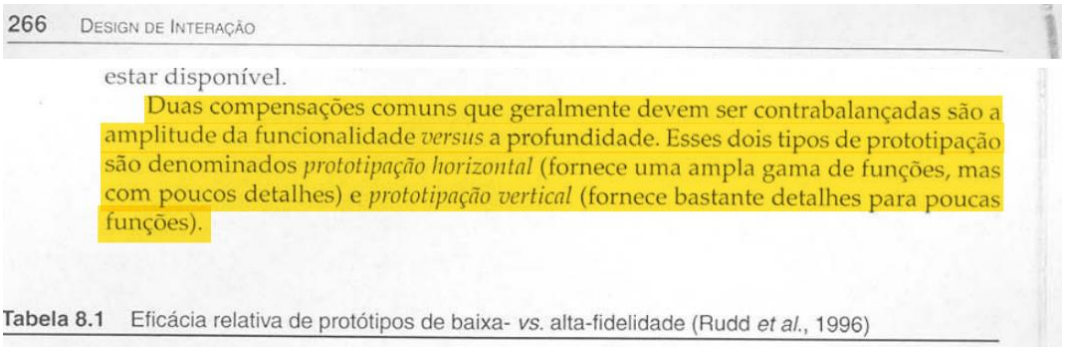
Severidade (Doolan, p. 178): ( ) Menor ( ) Maior ( ) Super-maior

Origem: Matheus do Vale Lameira, Lista de Verificação: Storyboard, item 6.

Item 65: Fica claro se o storyboard é um protótipo horizontal (muitas funções, com poucos detalhes) ou vertical (muitos detalhes, para poucas funções)?

Fonte: Página: 266, Capítulo 8. Design, Prototipação e Construção (Seção 8.2.5 Concessões na prototipação). Livro: JENNIFER PREECE, YVONNE ROGERS, HELEN SHARP, Design de Interação: além da interação homem-computador, Editora Bookman, 2005.

Foto:



266 DESIGN DE INTERAÇÃO

estar disponível.

Duas compensações comuns que geralmente devem ser contrabalançadas são a amplitude da funcionalidade *versus* a profundidade. Esses dois tipos de prototipação são denominados *prototipação horizontal* (fornece uma ampla gama de funções, mas com poucos detalhes) e *prototipação vertical* (fornece bastante detalhes para poucas funções).

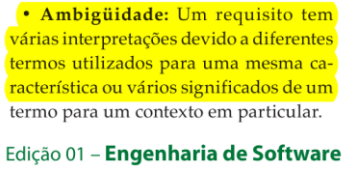
Tabela 8.1 Eficácia relativa de protótipos de baixa- *vs.* alta-fidelidade (Rudd *et al.*, 1996)

Resposta: ( ) Conforme ( ) Não conforme ( ) Não se aplica

Critério de Fagan: se a resposta for Não conforme, o defeito é do tipo Ambiguidade: o artefato admite mais de uma interpretação.

Fonte: Página: 69, Definição dos Conceitos (taxonomia de defeitos). Artigo: MARCOS KALINOWSKI, RODRIGO OLIVEIRA SPÍNOLA, Introdução à Inspeção de Software, Engenharia de Software Magazine, edição 01, p. 68 a 74.

Foto:



• **Ambigüidade:** Um requisito tem várias interpretações devido a diferentes termos utilizados para uma mesma característica ou vários significados de um termo para um contexto em particular.

Edição 01 – **Engenharia de Software Magazine** 69

Severidade (Doolan, p. 178): ( ) Menor ( ) Maior ( ) Super-maior

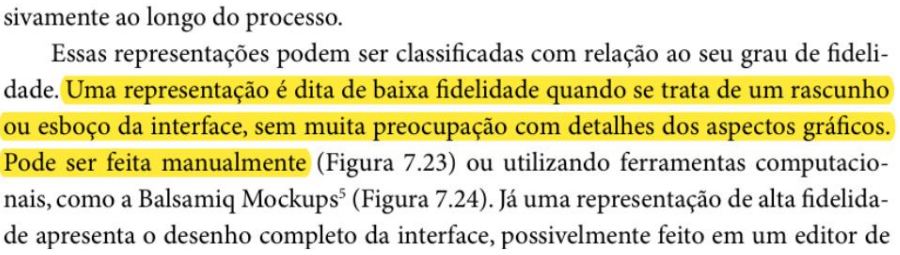
Origem: Matheus do Vale Lameira, Lista de Verificação: Storyboard, item 8.

4.7 Protótipo de papel e prototipação

Item 66: O protótipo de baixa fidelidade é um rascunho ou esboço da interface, sem muita preocupação com os detalhes gráficos, e pode ser feito à mão?

Fonte: Página: 249, Capítulo 7. Design de IHC. Livro: SIMONE DINIZ JUNQUEIRA BARBOSA, BRUNO SANTANA DA SILVA, Interação Humano-Computador, 1ª ed., Editora Campus/Elsevier, 2010.

Foto:



sivamente ao longo do processo.

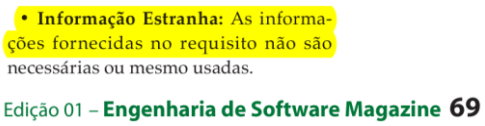
Essas representações podem ser classificadas com relação ao seu grau de fidelidade. Uma representação é dita de baixa fidelidade quando se trata de um rascunho ou esboço da interface, sem muita preocupação com detalhes dos aspectos gráficos. Pode ser feita manualmente (Figura 7.23) ou utilizando ferramentas computacionais, como a Balsamiq Mockups[5] (Figura 7.24). Já uma representação de alta fidelidade apresenta o desenho completo da interface, possivelmente feito em um editor de

Resposta: ( ) Conforme ( ) Não conforme ( ) Não se aplica

Critério de Fagan: se a resposta for Não conforme, o defeito é do tipo Informação estranha: o artefato traz informação desnecessária ou que não é usada.

Fonte: Página: 69, Definição dos Conceitos (taxonomia de defeitos). Artigo: MARCOS KALINOWSKI, RODRIGO OLIVEIRA SPÍNOLA, Introdução à Inspeção de Software, Engenharia de Software Magazine, edição 01, p. 68 a 74.

Foto:



• **Informação Estranha:** As informações fornecidas no requisito não são necessárias ou mesmo usadas.

Edição 01 – **Engenharia de Software Magazine** **69**

Severidade (Doolan, p. 178): ( ) Menor ( ) Maior ( ) Super-maior

Origem: Felipe Lopes Pedroza, Lista de Verificação: Entrega 6 (protótipo de papel, relato e planejamento da avaliação), item 1.

Item 67: As telas estão representadas em papel, em geral desenhadas à mão livre, sem detalhes que não sejam relevantes para a avaliação, e destacam os elementos principais com os quais o usuário vai interagir?

Fonte: Página: 358, Capítulo 10. Métodos de Avaliação de IHC. Livro: SIMONE DINIZ JUNQUEIRA BARBOSA, BRUNO SANTANA DA SILVA, Interação Humano-Computador, 1ª ed., Editora Campus/Elsevier, 2010.

Foto:

Na atividade de *preparação*, além do material de apoio elaborado na maioria das avaliações com usuários (veja Seção 9.7.2), o avaliador deve elaborar protótipos em papel. Representamos as telas do sistema em papel, em geral desenhadas à mão livre e sem nos preocuparmos com detalhes de interface que não sejam relevantes para a avaliação. A intenção é representar e destacar os elementos principais da interface com os quais o usuário vai interagir durante a simulação da interação. Além das telas estáticas, o avaliador também deve preparar outros pedaços de papel com partes da interface que se modificam durante a interação, como menus, dicas sobre elementos

Resposta: ( ) Conforme ( ) Não conforme ( ) Não se aplica

Critério de Fagan: se a resposta for Não conforme, o defeito é do tipo Informação estranha: o artefato traz informação desnecessária ou que não é usada.

Fonte: Página: 69, Definição dos Conceitos (taxonomia de defeitos). Artigo: MARCOS KALINOWSKI, RODRIGO OLIVEIRA SPÍNOLA, Introdução à Inspeção de Software, Engenharia de Software Magazine, edição 01, p. 68 a 74.

Foto:

• **Informação Estranha:** As informações fornecidas no requisito não são necessárias ou mesmo usadas.

Edição 01 – **Engenharia de Software Magazine** **69**

Severidade (Doolan, p. 178): ( ) Menor ( ) Maior ( ) Super-maior

Origem: Felipe Lopes Pedroza, Lista de Verificação: Entrega 6 (protótipo de papel, relato e planejamento da avaliação), item 2.

Item 68: Além das telas estáticas, o protótipo inclui pedaços de papel com as partes da interface que mudam durante a interação, como menus, dicas, itens de lista, resultados de busca e diálogos?

Fonte: Página: 358, Capítulo 10. Métodos de Avaliação de IHC. Livro: SIMONE DINIZ JUNQUEIRA BARBOSA, BRUNO SANTANA DA SILVA, Interação Humano-Computador, 1ª ed., Editora Campus/Elsevier, 2010.

Foto:

e sem nos preocuparmos com detalhes de interface que não sejam relevantes para a avaliação. A intenção é representar e destacar os elementos principais da interface com os quais o usuário vai interagir durante a simulação da interação. Além das telas estáticas, o avaliador também deve preparar outros pedaços de papel com partes da interface que se modificam durante a interação, como menus, dicas sobre elementos de interface, itens de alguma lista, resultados de busca e diálogos, por exemplo. O que for possível prever deve ser preparado antes das simulações de uso. O que não for possível será desenhado no papel durante as simulações.

Resposta: ( ) Conforme ( ) Não conforme ( ) Não se aplica

Critério de Fagan: se a resposta for Não conforme, o defeito é do tipo Omissão: falta no artefato um elemento importante que o item exige.

Fonte: Página: 69, Definição dos Conceitos (taxonomia de defeitos). Artigo: MARCOS KALINOWSKI, RODRIGO OLIVEIRA SPÍNOLA, Introdução à Inspeção de Software, Engenharia de Software Magazine, edição 01, p. 68 a 74.

Foto:

• **Omissão:** (1) Algum requisito importante relacionado à funcionalidade, ao desempenho, às restrições de projeto, ao atributo, ou à interface externa não foi incluído; (2) não está definida a resposta do software para todas as possíveis situações de entrada de dados; (3) faltam seções na especificação de requisitos; (4) faltam referências de figuras, tabelas, e diagramas; (5) falta definição de termos e unidades de medidas.

Edição 01 – **Engenharia de Software Magazine** **69**

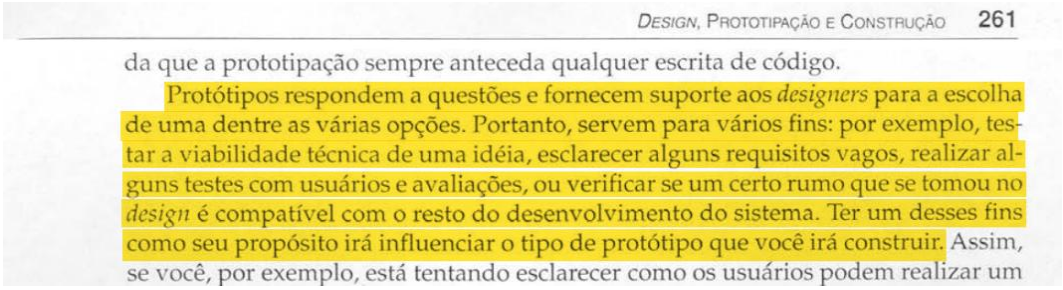
Severidade (Doolan, p. 178): ( ) Menor ( ) Maior ( ) Super-maior

Origem: Felipe Lopes Pedroza, Lista de Verificação: Entrega 6 (protótipo de papel, relato e planejamento da avaliação), item 3.

Item 69: O propósito do protótipo está declarado como pelo menos um destes: testar a viabilidade técnica de uma ideia, esclarecer requisitos vagos, fazer testes com usuários e avaliações ou verificar se o rumo do design é compatível com o resto do desenvolvimento?

Fonte: Página: 261, Capítulo 8. Design, Prototipação e Construção (Seção 8.2.2 Por que fazer protótipos?). Livro: JENNIFER PREECE, YVONNE ROGERS, HELEN SHARP, Design de Interação: além da interação homem-computador, Editora Bookman, 2005.

Foto:



DESIGN, PROTOTIPAÇÃO E CONSTRUÇÃO 261

da que a prototipação sempre anteceda qualquer escrita de código.

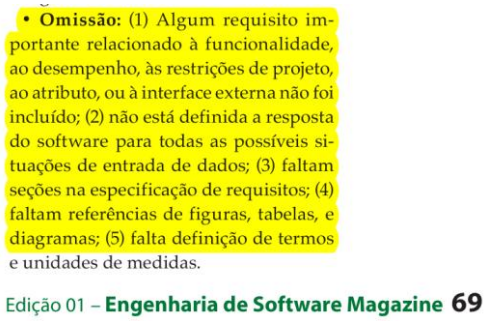
Protótipos respondem a questões e fornecem suporte aos *designers* para a escolha de uma dentre as várias opções. Portanto, servem para vários fins: por exemplo, testar a viabilidade técnica de uma idéia, esclarecer alguns requisitos vagos, realizar alguns testes com usuários e avaliações, ou verificar se um certo rumo que se tomou no *design* é compatível com o resto do desenvolvimento do sistema. Ter um desses fins como seu propósito irá influenciar o tipo de protótipo que você irá construir. Assim, se você, por exemplo, está tentando esclarecer como os usuários podem realizar um

Resposta: ( ) Conforme ( ) Não conforme ( ) Não se aplica

Critério de Fagan: se a resposta for Não conforme, o defeito é do tipo Omissão: falta no artefato um elemento importante que o item exige.

Fonte: Página: 69, Definição dos Conceitos (taxonomia de defeitos). Artigo: MARCOS KALINOWSKI, RODRIGO OLIVEIRA SPÍNOLA, Introdução à Inspeção de Software, Engenharia de Software Magazine, edição 01, p. 68 a 74.

Foto:



• **Omissão:** (1) Algum requisito importante relacionado à funcionalidade, ao desempenho, às restrições de projeto, ao atributo, ou à interface externa não foi incluído; (2) não está definida a resposta do software para todas as possíveis situações de entrada de dados; (3) faltam seções na especificação de requisitos; (4) faltam referências de figuras, tabelas, e diagramas; (5) falta definição de termos e unidades de medidas.

Edição 01 – **Engenharia de Software Magazine** 69

Severidade (Doolan, p. 178): ( ) Menor ( ) Maior ( ) Super-maior

Origem: Matheus do Vale Lameira, Lista de Verificação: Storyboard, item 22.

Item 70: Foram elaborados os cenários de interação e definidas as tarefas concretas que os participantes vão executar durante a avaliação?

Fonte: Página: 359, Capítulo 10. Métodos de Avaliação de IHC (Prototipação em Papel). Livro: SIMONE DINIZ JUNQUEIRA BARBOSA, BRUNO SANTANA DA SILVA, Interação Humano-Computador, 1ª ed., Editora Campus/Elsevier, 2010.

Foto:

Tabela 10.7 Atividades do método de prototipação em papel

prototipação em papel	
atividade	**tarefa**
Preparação	▪ definir tarefas para os participantes executarem ▪ definir o perfil dos participantes e recrutá-los ▪ criar protótipos em papel da interface para executar as tarefas ▪ executar um teste-piloto
Coleta de dados	▪ cada usuário deve executar as tarefas propostas interagindo com os protótipos em papel, mediado pelo avaliador
Interpretação	▪ avaliador deve – listar os problemas encontrados – refinar os protótipos em papel para resolver os problemas mais simples
Consolidação dos resultados	▪ priorizar a correção dos problemas não resolvidos ▪ sugerir correções
Relato dos resultados	▪ relatar os problemas encontrados e sugestões de correção

O Exemplo 10.10 apresenta um esboço de tela utilizado em uma sessão de prototipação em papel. Observe que há anotações em pedaços de papel adesivo que serão manipulados durante a sessão de avaliação.

Resposta: ( ) Conforme ( ) Não conforme ( ) Não se aplica

Critério de Fagan: se a resposta for Não conforme, o defeito é do tipo Omissão: falta no artefato um elemento importante que o item exige.

Fonte: Página: 69, Definição dos Conceitos (taxonomia de defeitos). Artigo: MARCOS KALINOWSKI, RODRIGO OLIVEIRA SPÍNOLA, Introdução à Inspeção de Software, Engenharia de Software Magazine, edição 01, p. 68 a 74.

Foto:

• **Omissão:** (1) Algum requisito importante relacionado à funcionalidade, ao desempenho, às restrições de projeto, ao atributo, ou à interface externa não foi incluído; (2) não está definida a resposta do software para todas as possíveis situações de entrada de dados; (3) faltam seções na especificação de requisitos; (4) faltam referências de figuras, tabelas, e diagramas; (5) falta definição de termos e unidades de medidas.

Edição 01 – **Engenharia de Software Magazine** **69**

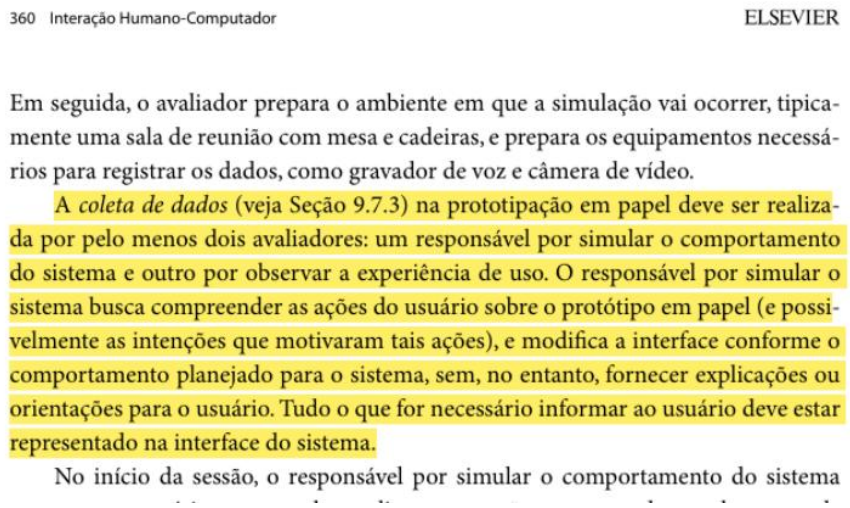
Severidade (Doolan, p. 178): ( ) Menor ( ) Maior ( ) Super-maior

Origem: Victor Andreozzi De La Rocque Couto, Lista de Verificação: Protótipo de Papel, item 5.

Item 71: A equipe tem pelo menos dois avaliadores, com o papel de cada um definido: um atua como o "computador", manipulando os papéis e simulando a reação do sistema, e o outro observa e anota a experiência de uso?

Fonte: Página: 360, Capítulo 10. Métodos de Avaliação de IHC (Prototipação em Papel). Livro: SIMONE DINIZ JUNQUEIRA BARBOSA, BRUNO SANTANA DA SILVA, Interação Humano-Computador, 1ª ed., Editora Campus/Elsevier, 2010.

Foto:



360 Interação Humano-Computador ELSEVIER

Em seguida, o avaliador prepara o ambiente em que a simulação vai ocorrer, tipicamente uma sala de reunião com mesa e cadeiras, e prepara os equipamentos necessários para registrar os dados, como gravador de voz e câmera de vídeo.

A *coleta de dados* (veja Seção 9.7.3) na prototipação em papel deve ser realizada por pelo menos dois avaliadores: um responsável por simular o comportamento do sistema e outro por observar a experiência de uso. O responsável por simular o sistema busca compreender as ações do usuário sobre o protótipo em papel (e possivelmente as intenções que motivaram tais ações), e modifica a interface conforme o comportamento planejado para o sistema, sem, no entanto, fornecer explicações ou orientações para o usuário. Tudo o que for necessário informar ao usuário deve estar representado na interface do sistema.

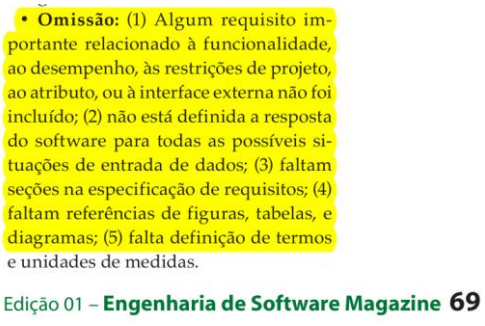
No início da sessão, o responsável por simular o comportamento do sistema

Resposta: ( ) Conforme ( ) Não conforme ( ) Não se aplica

Critério de Fagan: se a resposta for Não conforme, o defeito é do tipo Omissão: falta no artefato um elemento importante que o item exige.

Fonte: Página: 69, Definição dos Conceitos (taxonomia de defeitos). Artigo: MARCOS KALINOWSKI, RODRIGO OLIVEIRA SPÍNOLA, Introdução à Inspeção de Software, Engenharia de Software Magazine, edição 01, p. 68 a 74.

Foto:



• **Omissão:** (1) Algum requisito importante relacionado à funcionalidade, ao desempenho, às restrições de projeto, ao atributo, ou à interface externa não foi incluído; (2) não está definida a resposta do software para todas as possíveis situações de entrada de dados; (3) faltam seções na especificação de requisitos; (4) faltam referências de figuras, tabelas, e diagramas; (5) falta definição de termos e unidades de medidas.

Edição 01 – **Engenharia de Software Magazine** **69**

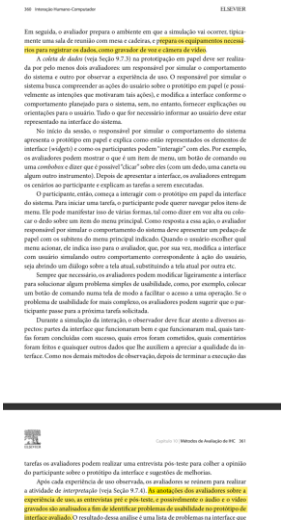
Severidade (Doolan, p. 178): ( ) Menor ( ) Maior ( ) Super-maior

Origem: Victor Andreozzi De La Rocque Couto, Lista de Verificação: Protótipo de Papel, item 7.

Item 72: Foram especificados os métodos de coleta de dados das sessões, como gravação de áudio e vídeo, anotações das ações e dos erros, protocolo de pensar em voz alta e questionários ou entrevistas antes e depois do teste?

Fonte: Página: 360 e 361, Capítulo 10. Métodos de Avaliação de IHC (Prototipação em Papel). Livro: SIMONE DINIZ JUNQUEIRA BARBOSA, BRUNO SANTANA DA SILVA, Interação Humano-Computador, 1ª ed., Editora Campus/Elsevier, 2010.

Foto:



360 Interação Humano-Computador ELSEVIER

Em seguida, o avaliador prepara o ambiente em que a simulação vai ocorrer, tipicamente uma sala de reunião com mesa e cadeiras, e prepara os equipamentos necessários para registrar os dados, como gravador de voz e câmera de vídeo.

A *coleta de dados* (veja Seção 9.7.3) na prototipação em papel deve ser realizada por pelo menos dois avaliadores: um responsável por simular o comportamento do sistema e outro por observar a experiência de uso. O responsável por simular o sistema busca compreender as ações do usuário sobre o protótipo em papel (e possivelmente as intenções que motivaram tais ações), e modifica a interface conforme o comportamento planejado para o sistema, sem, no entanto, fornecer explicações ou orientações para o usuário. Tudo o que for necessário informar ao usuário deve estar representado na interface do sistema.

No início da sessão, o responsável por simular o comportamento do sistema apresenta o protótipo em papel e explica como estão representados os elementos de interface (widgets) e como os participantes podem "interagir" com eles. Por exemplo, os avaliadores podem mostrar o que é um item de menu, um botão de comando ou uma *combobox* e dizer que é possível "clicar" sobre eles (com um dedo, uma caneta ou algum outro instrumento). Depois de apresentar a interface, os avaliadores entregam os cenários ao participante e explicam as tarefas a serem executadas.

O participante, então, começa a interagir com o protótipo em papel da interface do sistema. Para iniciar uma tarefa, o participante pode querer navegar pelos itens de menu. Ele pode manifestar isso de várias formas, tal como dizer em voz alta ou colocar o dedo sobre um item do menu principal. Como resposta a essa ação, o avaliador responsável por simular o comportamento do sistema deve apresentar um pedaço de papel com os subitens do menu principal indicado. Quando o usuário escolher qual menu acionar, ele indica isso para o avaliador, que, por sua vez, modifica a interface com usuário simulando outro comportamento correspondente à ação do usuário, seja abrindo um diálogo sobre a tela atual, substituindo a tela atual por outra etc.

Sempre que necessário, os avaliadores podem modificar ligeiramente a interface para solucionar algum problema simples de usabilidade, como, por exemplo, colocar um botão de comando numa tela de modo a facilitar o acesso a uma operação. Se o problema de usabilidade for mais complexo, os avaliadores podem sugerir que o participante passe para a próxima tarefa solicitada.

Durante a simulação da interação, o observador deve ficar atento a diversos aspectos: partes da interface que funcionaram bem e que funcionaram mal, quais tarefas foram concluídas com sucesso, quais erros foram cometidos, quais comentários foram feitos e quaisquer outros dados que lhe auxiliem a apreciar a qualidade da interface. Como nos demais métodos de observação, depois de terminar a execução das

Capítulo 10 | Métodos de Avaliação de IHC 361

tarefas os avaliadores podem realizar uma entrevista pós-teste para colher a opinião do participante sobre o protótipo da interface e sugestões de melhorias.

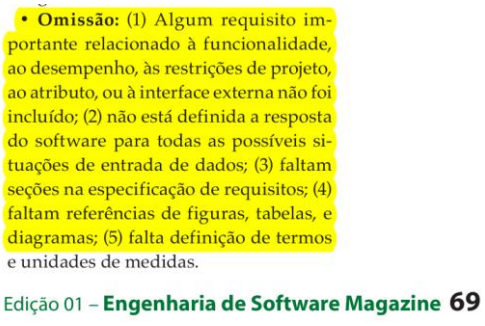
Após cada experiência de uso observada, os avaliadores se reúnem para realizar a atividade de *interpretação* (veja Seção 9.7.4). As anotações dos avaliadores sobre a experiência de uso, as entrevistas pré e pós-teste, e possivelmente o áudio e o vídeo gravados são analisados a fim de identificar problemas de usabilidade no protótipo de interface avaliado. O resultado dessa análise é uma lista de problemas na interface que

Resposta: ( ) Conforme ( ) Não conforme ( ) Não se aplica

Critério de Fagan: se a resposta for Não conforme, o defeito é do tipo Omissão: falta no artefato um elemento importante que o item exige.

Fonte: Página: 69, Definição dos Conceitos (taxonomia de defeitos). Artigo: MARCOS KALINOWSKI, RODRIGO OLIVEIRA SPÍNOLA, Introdução à Inspeção de Software, Engenharia de Software Magazine, edição 01, p. 68 a 74.

Foto:



• **Omissão:** (1) Algum requisito importante relacionado à funcionalidade, ao desempenho, às restrições de projeto, ao atributo, ou à interface externa não foi incluído; (2) não está definida a resposta do software para todas as possíveis situações de entrada de dados; (3) faltam seções na especificação de requisitos; (4) faltam referências de figuras, tabelas, e diagramas; (5) falta definição de termos e unidades de medidas.

Edição 01 – **Engenharia de Software Magazine** **69**

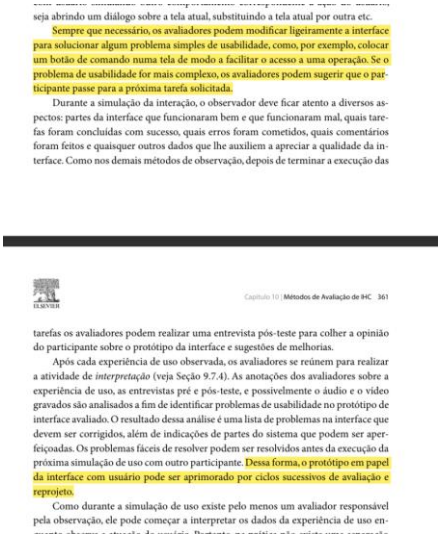
Severidade (Doolan, p. 178): ( ) Menor ( ) Maior ( ) Super-maior

Origem: Victor Andreozzi De La Rocque Couto, Lista de Verificação: Protótipo de Papel, item 8.

Item 73: Está previsto o ciclo de reprojeto do protótipo de papel, em que problemas simples de usabilidade são corrigidos e testados de novo entre as sessões?

Fonte: Página: 360 e 361, Capítulo 10. Métodos de Avaliação de IHC (Prototipação em Papel). Livro: SIMONE DINIZ JUNQUEIRA BARBOSA, BRUNO SANTANA DA SILVA, Interação Humano-Computador, 1ª ed., Editora Campus/Elsevier, 2010.

Foto:



seja abrindo um diálogo sobre a tela atual, substituindo a tela atual por outra etc.

Sempre que necessário, os avaliadores podem modificar ligeiramente a interface para solucionar algum problema simples de usabilidade, como, por exemplo, colocar um botão de comando numa tela de modo a facilitar o acesso a uma operação. Se o problema de usabilidade for mais complexo, os avaliadores podem sugerir que o participante passe para a próxima tarefa solicitada.

Durante a simulação da interação, o observador deve ficar atento a diversos aspectos: partes da interface que funcionaram bem e que funcionaram mal, quais tarefas foram concluídas com sucesso, quais erros foram cometidos, quais comentários foram feitos e quaisquer outros dados que lhe auxiliem a apreciar a qualidade da interface. Como nos demais métodos de observação, depois de terminar a execução das

Capítulo 10 | Métodos de Avaliação de IHC 361

tarefas os avaliadores podem realizar uma entrevista pós-teste para colher a opinião do participante sobre o protótipo da interface e sugestões de melhorias.

Após cada experiência de uso observada, os avaliadores se reúnem para realizar a atividade de *interpretação* (veja Seção 9.7.4). As anotações dos avaliadores sobre a experiência de uso, as entrevistas pré e pós-teste, e possivelmente o áudio e o vídeo gravados são analisados a fim de identificar problemas de usabilidade no protótipo de interface avaliado. O resultado dessa análise é uma lista de problemas na interface que devem ser corrigidos, além de indicações de partes do sistema que podem ser aperfeiçoadas. Os problemas fáceis de resolver podem ser resolvidos antes da execução da próxima simulação de uso com outro participante. Dessa forma, o protótipo em papel da interface com usuário pode ser aprimorado por ciclos sucessivos de avaliação e reprojeto.

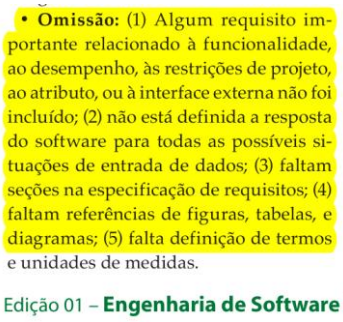
Como durante a simulação de uso existe pelo menos um avaliador responsável pela observação, ele pode começar a interpretar os dados da experiência de uso en-

Resposta: ( ) Conforme ( ) Não conforme ( ) Não se aplica

Critério de Fagan: se a resposta for Não conforme, o defeito é do tipo Omissão: falta no artefato um elemento importante que o item exige.

Fonte: Página: 69, Definição dos Conceitos (taxonomia de defeitos). Artigo: MARCOS KALINOWSKI, RODRIGO OLIVEIRA SPÍNOLA, Introdução à Inspeção de Software, Engenharia de Software Magazine, edição 01, p. 68 a 74.

Foto:



• **Omissão:** (1) Algum requisito importante relacionado à funcionalidade, ao desempenho, às restrições de projeto, ao atributo, ou à interface externa não foi incluído; (2) não está definida a resposta do software para todas as possíveis situações de entrada de dados; (3) faltam seções na especificação de requisitos; (4) faltam referências de figuras, tabelas, e diagramas; (5) falta definição de termos e unidades de medidas.

Edição 01 – **Engenharia de Software Magazine** **69**

Severidade (Doolan, p. 178): ( ) Menor ( ) Maior ( ) Super-maior

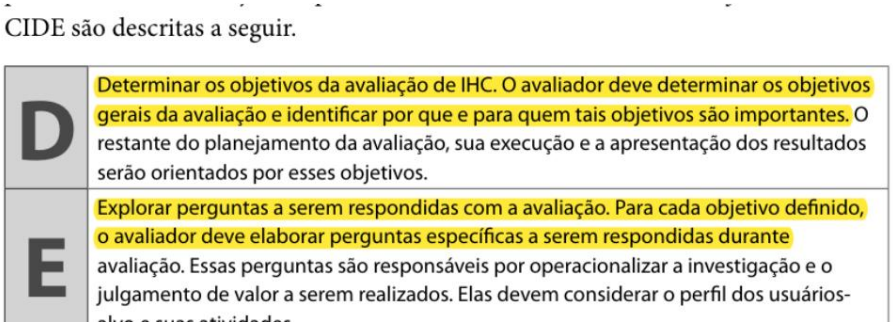
Origem: Victor Andreozzi De La Rocque Couto, Lista de Verificação: Protótipo de Papel, item 11.

4.8 Planejamento da avaliação (framework DECIDE)

Item 74: O planejamento determina os objetivos gerais da avaliação (por que e para quem são importantes) e explora, para cada objetivo, as perguntas a responder, considerando o perfil dos usuários-alvo (D e E do DECIDE)?

Fonte: Página: 313, Capítulo 9. Planejamento da Avaliação de IHC. Livro: SIMONE DINIZ JUNQUEIRA BARBOSA, BRUNO SANTANA DA SILVA, Interação Humano-Computador, 1ª ed., Editora Campus/Elsevier, 2010.

Foto:



CIDE são descritas a seguir.

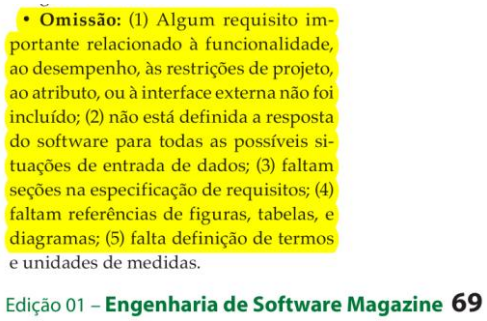
D	Determinar os objetivos da avaliação de IHC. O avaliador deve determinar os objetivos gerais da avaliação e identificar por que e para quem tais objetivos são importantes. O restante do planejamento da avaliação, sua execução e a apresentação dos resultados serão orientados por esses objetivos.
E	Explorar perguntas a serem respondidas com a avaliação. Para cada objetivo definido, o avaliador deve elaborar perguntas específicas a serem respondidas durante avaliação. Essas perguntas são responsáveis por operacionalizar a investigação e o julgamento de valor a serem realizados. Elas devem considerar o perfil dos usuários-

Resposta: ( ) Conforme ( ) Não conforme ( ) Não se aplica

Critério de Fagan: se a resposta for Não conforme, o defeito é do tipo Omissão: falta no artefato um elemento importante que o item exige.

Fonte: Página: 69, Definição dos Conceitos (taxonomia de defeitos). Artigo: MARCOS KALINOWSKI, RODRIGO OLIVEIRA SPÍNOLA, Introdução à Inspeção de Software, Engenharia de Software Magazine, edição 01, p. 68 a 74.

Foto:



• **Omissão:** (1) Algum requisito importante relacionado à funcionalidade, ao desempenho, às restrições de projeto, ao atributo, ou à interface externa não foi incluído; (2) não está definida a resposta do software para todas as possíveis situações de entrada de dados; (3) faltam seções na especificação de requisitos; (4) faltam referências de figuras, tabelas, e diagramas; (5) falta definição de termos e unidades de medidas.

Edição 01 – **Engenharia de Software Magazine** **69**

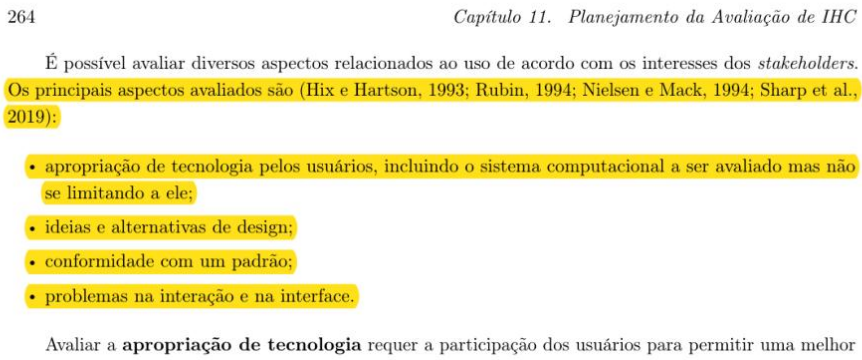
Severidade (Doolan, p. 178): ( ) Menor ( ) Maior ( ) Super-maior

Origem: Felipe Lopes Pedroza, Lista de Verificação: Entrega 6 (protótipo de papel, relato e planejamento da avaliação), item 7.

Item 75: Cada objetivo da avaliação está associado a pelo menos um dos aspectos que podem ser avaliados: apropriação de tecnologia pelos usuários, ideias e alternativas de design, conformidade com um padrão ou problemas na interação e na interface?

Fonte: Página: 264, Capítulo 11. Planejamento da Avaliação de IHC (Seção 11.2 O que Avaliar?). Livro: SIMONE DINIZ JUNQUEIRA BARBOSA, BRUNO SANTANA DA SILVA, MILENE SELBACH SILVEIRA, ISABELA GASPARINI, TICIANNE DARIN, GABRIEL DINIZ JUNQUEIRA BARBOSA, Interação Humano-Computador e Experiência do Usuário, 1ª ed., Rio de Janeiro, 2021.

Foto:



264 *Capítulo 11. Planejamento da Avaliação de IHC*

É possível avaliar diversos aspectos relacionados ao uso de acordo com os interesses dos *stakeholders*. Os principais aspectos avaliados são (Hix e Hartson, 1993; Rubin, 1994; Nielsen e Mack, 1994; Sharp et al., 2019):

- apropriação de tecnologia pelos usuários, incluindo o sistema computacional a ser avaliado mas não se limitando a ele;
- ideias e alternativas de design;
- conformidade com um padrão;
- problemas na interação e na interface.

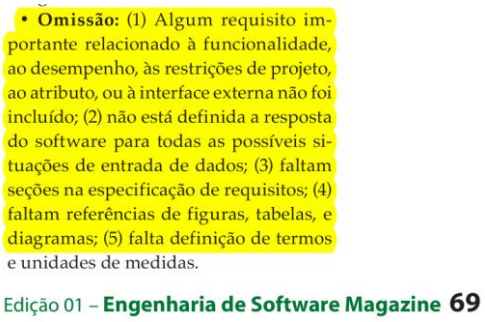
Avaliar a **apropriação de tecnologia** requer a participação dos usuários para permitir uma melhor

Resposta: ( ) Conforme ( ) Não conforme ( ) Não se aplica

Critério de Fagan: se a resposta for Não conforme, o defeito é do tipo Omissão: falta no artefato um elemento importante que o item exige.

Fonte: Página: 69, Definição dos Conceitos (taxonomia de defeitos). Artigo: MARCOS KALINOWSKI, RODRIGO OLIVEIRA SPÍNOLA, Introdução à Inspeção de Software, Engenharia de Software Magazine, edição 01, p. 68 a 74.

Foto:



- **Omissão:** (1) Algum requisito importante relacionado à funcionalidade, ao desempenho, às restrições de projeto, ao atributo, ou à interface externa não foi incluído; (2) não está definida a resposta do software para todas as possíveis situações de entrada de dados; (3) faltam seções na especificação de requisitos; (4) faltam referências de figuras, tabelas, e diagramas; (5) falta definição de termos e unidades de medidas.

Edição 01 – **Engenharia de Software Magazine** **69**

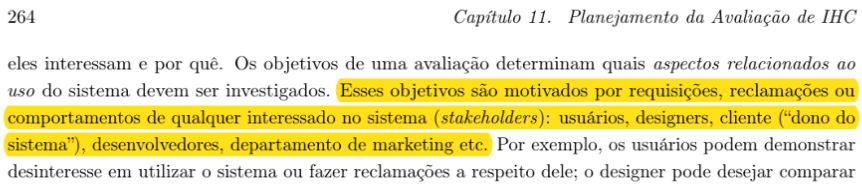
Severidade (Doolan, p. 178): ( ) Menor ( ) Maior ( ) Super-maior

Origem: Matheus do Vale Lameira, Lista de Verificação: Storyboard, item 21.

Item 76: Os objetivos da avaliação indicam qual interessado (usuários, designers, cliente, desenvolvedores, marketing) os motivou, por requisições, reclamações ou comportamentos?

Fonte: Página: 264, Capítulo 11. Planejamento da Avaliação de IHC (Seção 11.2 O que Avaliar?). Livro: SIMONE DINIZ JUNQUEIRA BARBOSA, BRUNO SANTANA DA SILVA, MILENE SELBACH SILVEIRA, ISABELA GASPARINI, TICIANNE DARIN, GABRIEL DINIZ JUNQUEIRA BARBOSA, Interação Humano-Computador e Experiência do Usuário, 1ª ed., Rio de Janeiro, 2021.

Foto:



264 *Capítulo 11. Planejamento da Avaliação de IHC*

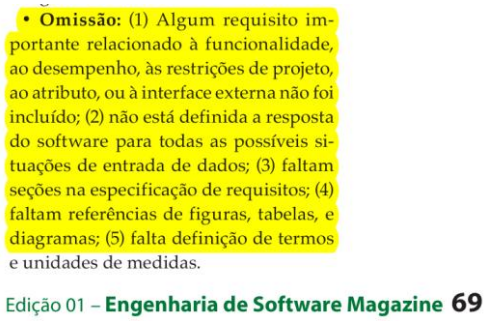
eles interessam e por quê. Os objetivos de uma avaliação determinam quais *aspectos relacionados ao uso* do sistema devem ser investigados. Esses objetivos são motivados por requisições, reclamações ou comportamentos de qualquer interessado no sistema (*stakeholders*): usuários, designers, cliente ("dono do sistema"), desenvolvedores, departamento de marketing etc. Por exemplo, os usuários podem demonstrar desinteresse em utilizar o sistema ou fazer reclamações a respeito dele; o designer pode desejar comparar

Resposta: ( ) Conforme ( ) Não conforme ( ) Não se aplica

Critério de Fagan: se a resposta for Não conforme, o defeito é do tipo Omissão: falta no artefato um elemento importante que o item exige.

Fonte: Página: 69, Definição dos Conceitos (taxonomia de defeitos). Artigo: MARCOS KALINOWSKI, RODRIGO OLIVEIRA SPÍNOLA, Introdução à Inspeção de Software, Engenharia de Software Magazine, edição 01, p. 68 a 74.

Foto:



• **Omissão:** (1) Algum requisito importante relacionado à funcionalidade, ao desempenho, às restrições de projeto, ao atributo, ou à interface externa não foi incluído; (2) não está definida a resposta do software para todas as possíveis situações de entrada de dados; (3) faltam seções na especificação de requisitos; (4) faltam referências de figuras, tabelas, e diagramas; (5) falta definição de termos e unidades de medidas.

Edição 01 – **Engenharia de Software Magazine** **69**

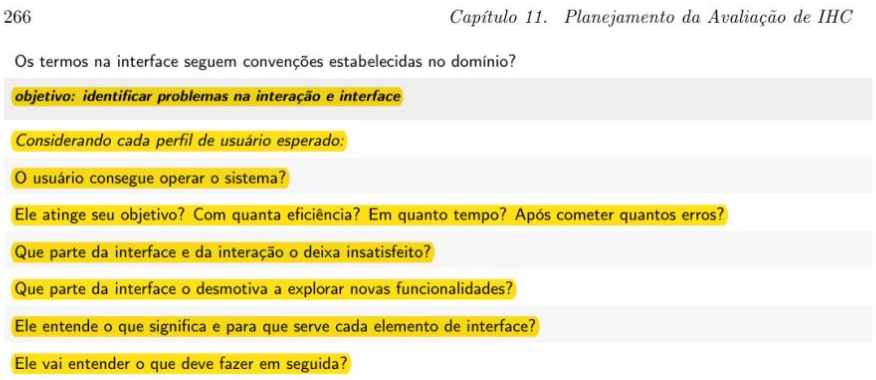
Severidade (Doolan, p. 178): ( ) Menor ( ) Maior ( ) Super-maior

Origem: Matheus do Vale Lameira, Lista de Verificação: Storyboard, item 23.

Item 77: Para o objetivo de identificar problemas na interação e na interface, o planejamento traz, para cada perfil de usuário, perguntas como: o usuário consegue operar o sistema? Atinge seu objetivo, com que eficiência e após quantos erros? Que parte o deixa insatisfeito? Entende o que deve fazer em seguida?

Fonte: Página: 266, Capítulo 11. Planejamento da Avaliação de IHC (Seção 11.2 O que Avaliar?, Tabela 11.1). Livro: SIMONE DINIZ JUNQUEIRA BARBOSA, BRUNO SANTANA DA SILVA, MILENE SELBACH SILVEIRA, ISABELA GASPARINI, TICIANNE DARIN, GABRIEL DINIZ JUNQUEIRA BARBOSA, Interação Humano-Computador e Experiência do Usuário, 1ª ed., Rio de Janeiro, 2021.

Foto:



266 *Capítulo 11. Planejamento da Avaliação de IHC*

Os termos na interface seguem convenções estabelecidas no domínio?

objetivo: identificar problemas na interação e interface

Considerando cada perfil de usuário esperado:

O usuário consegue operar o sistema?

Ele atinge seu objetivo? Com quanta eficiência? Em quanto tempo? Após cometer quantos erros?

Que parte da interface e da interação o deixa insatisfeito?

Que parte da interface o desmotiva a explorar novas funcionalidades?

Ele entende o que significa e para que serve cada elemento de interface?

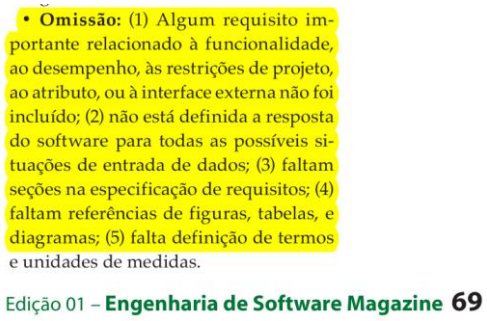
Ele vai entender o que deve fazer em seguida?

Resposta: ( ) Conforme ( ) Não conforme ( ) Não se aplica

Critério de Fagan: se a resposta for Não conforme, o defeito é do tipo Omissão: falta no artefato um elemento importante que o item exige.

Fonte: Página: 69, Definição dos Conceitos (taxonomia de defeitos). Artigo: MARCOS KALINOWSKI, RODRIGO OLIVEIRA SPÍNOLA, Introdução à Inspeção de Software, Engenharia de Software Magazine, edição 01, p. 68 a 74.

Foto:



• **Omissão:** (1) Algum requisito importante relacionado à funcionalidade, ao desempenho, às restrições de projeto, ao atributo, ou à interface externa não foi incluído; (2) não está definida a resposta do software para todas as possíveis situações de entrada de dados; (3) faltam seções na especificação de requisitos; (4) faltam referências de figuras, tabelas, e diagramas; (5) falta definição de termos e unidades de medidas.

Edição 01 – **Engenharia de Software Magazine** **69**

Severidade (Doolan, p. 178): ( ) Menor ( ) Maior ( ) Super-maior

Origem: Matheus do Vale Lameira, Lista de Verificação: Storyboard, item 25.

Item 78: Se um objetivo da avaliação é a conformidade com um padrão, o planejamento traz perguntas como: o sistema segue os padrões de acessibilidade? A interface segue o padrão do sistema operacional e da empresa? Os termos seguem as convenções do domínio?

Fonte: Página: 6 de 20 (PDF disponibilizado no Aprender), Capítulo Planejamento da Avaliação de IHC. Livro: SIMONE DINIZ JUNQUEIRA BARBOSA, BRUNO SANTANA DA SILVA, Interação Humano-Computador (PDF disponibilizado no Aprender).

Foto:

objetivo: verificar a conformidade com um padrão

O sistema está de acordo com os padrões de acessibilidade do W3C?

A interface segue o padrão do sistema operacional? E da empresa?

Os termos na interface seguem convenções estabelecidas no domínio?

Resposta: ( ) Conforme ( ) Não conforme ( ) Não se aplica

Critério de Fagan: se a resposta for Não conforme, o defeito é do tipo Omissão: falta no artefato um elemento importante que o item exige.

Fonte: Página: 69, Definição dos Conceitos (taxonomia de defeitos). Artigo: MARCOS KALINOWSKI, RODRIGO OLIVEIRA SPÍNOLA, Introdução à Inspeção de Software, Engenharia de Software Magazine, edição 01, p. 68 a 74.

Foto:

• **Omissão:** (1) Algum requisito importante relacionado à funcionalidade, ao desempenho, às restrições de projeto, ao atributo, ou à interface externa não foi incluído; (2) não está definida a resposta do software para todas as possíveis situações de entrada de dados; (3) faltam seções na especificação de requisitos; (4) faltam referências de figuras, tabelas, e diagramas; (5) falta definição de termos e unidades de medidas.

Edição 01 – **Engenharia de Software Magazine** **69**

Severidade (Doolan, p. 178): ( ) Menor ( ) Maior ( ) Super-maior

Origem: Victor Andreozzi De La Rocque Couto, Lista de Verificação: Planejamento da Avaliação (DECIDE), item 3.

Item 79: Se um objetivo é a apropriação da tecnologia, o planejamento traz perguntas como: de que maneira os usuários utilizam o sistema e em que difere do planejado? Quais são os pontos fortes e fracos, na opinião deles? Por que não incorporaram o sistema no cotidiano?

Fonte: Página: 6 de 20 (PDF disponibilizado no Aprender), Capítulo Planejamento da Avaliação de IHC. Livro: SIMONE DINIZ JUNQUEIRA BARBOSA, BRUNO SANTANA DA SILVA, Interação Humano-Computador (PDF disponibilizado no Aprender).

Foto:

objetivo: analisar a apropriação da tecnologia
De que maneira os usuários utilizam o sistema? Em que difere do planejado?
Como o sistema interativo afeta o modo de as pessoas se comunicarem e relacionarem?
Que variação houve no número de erros cometidos pelos usuários ao utilizarem o novo sistema? E no tempo que levam para atingir seus objetivos? E na sua satisfação com o sistema?
O quanto os usuários consideram o apoio computacional adequado para auxiliá-los na realização de suas atividades?
O quanto eles são motivados a explorar novas funcionalidades?
Quais são os pontos fortes e fracos do sistema, na opinião dos usuários?
Quais objetivos dos usuários podem ser alcançados através do sistema? E quais não podem? Quais necessidades e desejos foram ou não atendidos?
A tecnologia disponível pode oferecer maneiras mais interessantes ou eficientes de os usuários atingirem seus objetivos?
O que é possível modificar no sistema interativo para adequá-lo melhor ao ambiente de trabalho?
Por que os usuários não incorporaram o sistema no seu cotidiano?

Resposta: ( ) Conforme ( ) Não conforme ( ) Não se aplica

Critério de Fagan: se a resposta for Não conforme, o defeito é do tipo Omissão: falta no artefato um elemento importante que o item exige.

Fonte: Página: 69, Definição dos Conceitos (taxonomia de defeitos). Artigo: MARCOS KALINOWSKI, RODRIGO OLIVEIRA SPÍNOLA, Introdução à Inspeção de Software, Engenharia de Software Magazine, edição 01, p. 68 a 74.

Foto:

• **Omissão:** (1) Algum requisito importante relacionado à funcionalidade, ao desempenho, às restrições de projeto, ao atributo, ou à interface externa não foi incluído; (2) não está definida a resposta do software para todas as possíveis situações de entrada de dados; (3) faltam seções na especificação de requisitos; (4) faltam referências de figuras, tabelas, e diagramas; (5) falta definição de termos e unidades de medidas.

Edição 01 – **Engenharia de Software Magazine** **69**

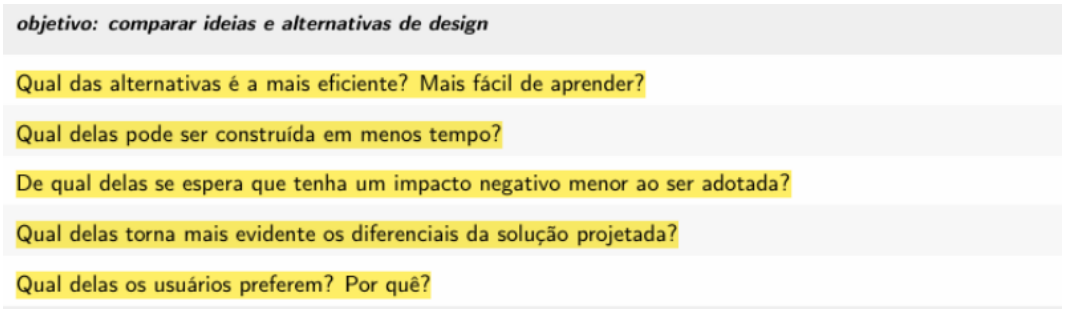
Severidade (Doolan, p. 178): ( ) Menor ( ) Maior ( ) Super-maior

Origem: Victor Andreozzi De La Rocque Couto, Lista de Verificação: Planejamento da Avaliação (DECIDE), item 4.

Item 80: Se um objetivo é comparar ideias e alternativas de design, o planejamento traz perguntas como: qual alternativa é mais eficiente e mais fácil de aprender? Qual pode ser construída em menos tempo? Qual os usuários preferem, e por quê?

Fonte: Página: 6 de 20 (PDF disponibilizado no Aprender), Capítulo Planejamento da Avaliação de IHC. Livro: SIMONE DINIZ JUNQUEIRA BARBOSA, BRUNO SANTANA DA SILVA, Interação Humano-Computador (PDF disponibilizado no Aprender).

Foto:

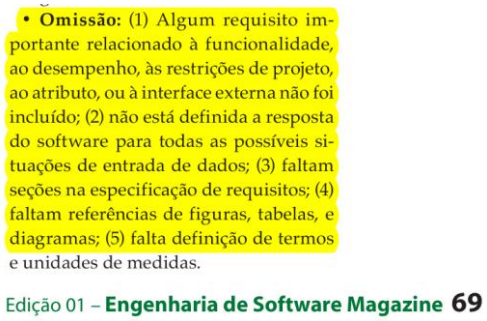


Resposta: ( ) Conforme ( ) Não conforme ( ) Não se aplica

Critério de Fagan: se a resposta for Não conforme, o defeito é do tipo Omissão: falta no artefato um elemento importante que o item exige.

Fonte: Página: 69, Definição dos Conceitos (taxonomia de defeitos). Artigo: MARCOS KALINOWSKI, RODRIGO OLIVEIRA SPÍNOLA, Introdução à Inspeção de Software, Engenharia de Software Magazine, edição 01, p. 68 a 74.

Foto:


• **Omissão:** (1) Algum requisito importante relacionado à funcionalidade, ao desempenho, às restrições de projeto, ao atributo, ou à interface externa não foi incluído; (2) não está definida a resposta do software para todas as possíveis situações de entrada de dados; (3) faltam seções na especificação de requisitos; (4) faltam referências de figuras, tabelas, e diagramas; (5) falta definição de termos e unidades de medidas.

Edição 01 – **Engenharia de Software Magazine** **69**

Severidade (Doolan, p. 178): ( ) Menor ( ) Maior ( ) Super-maior

Origem: Victor Andreozzi De La Rocque Couto, Lista de Verificação: Planejamento da Avaliação (DECIDE), item 5.

Item 81: O planejamento escolhe o método adequado ao prazo, ao orçamento, aos equipamentos e à experiência dos avaliadores e define as questões práticas: recrutamento, equipamentos, prazos, orçamento e mão de obra (C e I do DECIDE)?

Fonte: Página: 313, Capítulo 9. Planejamento da Avaliação de IHC. Livro: SIMONE DINIZ JUNQUEIRA BARBOSA, BRUNO SANTANA DA SILVA, Interação Humano-Computador, 1ª ed., Editora Campus/Elsevier, 2010.

Foto:

	julgamento de valor a serem realizados. Elas devem considerar o perfil dos usuários-alvo e suas atividades.
C	Escolher (*Choose*) os métodos de avaliação a serem utilizados. O avaliador deve escolher os métodos mais adequados para responder as perguntas e atingir os objetivos esperados, considerando também o prazo, o orçamento, os equipamentos disponíveis e o grau de conhecimento e experiência dos avaliadores.
I	Identificar e administrar as questões práticas da avaliação. Existem muitas questões práticas envolvidas numa avaliação de IHC, como, por exemplo, o recrutamento dos usuários que participarão da avaliação, a preparação e o uso dos equipamentos necessários, os prazos e o orçamento disponíveis, além da mão-de-obra necessária para conduzir a avaliação.
	Decidir como lidar com as questões éticas. Sempre que usuários são envolvidos numa

Resposta: ( ) Conforme ( ) Não conforme ( ) Não se aplica

Critério de Fagan: se a resposta for Não conforme, o defeito é do tipo Omissão: falta no artefato um elemento importante que o item exige.

Fonte: Página: 69, Definição dos Conceitos (taxonomia de defeitos). Artigo: MARCOS KALINOWSKI, RODRIGO OLIVEIRA SPÍNOLA, Introdução à Inspeção de Software, Engenharia de Software Magazine, edição 01, p. 68 a 74.

Foto:

• **Omissão:** (1) Algum requisito importante relacionado à funcionalidade, ao desempenho, às restrições de projeto, ao atributo, ou à interface externa não foi incluído; (2) não está definida a resposta do software para todas as possíveis situações de entrada de dados; (3) faltam seções na especificação de requisitos; (4) faltam referências de figuras, tabelas, e diagramas; (5) falta definição de termos e unidades de medidas.

Edição 01 – **Engenharia de Software Magazine** **69**

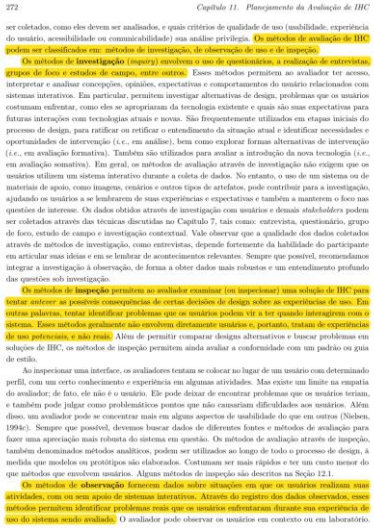
Severidade (Doolan, p. 178): ( ) Menor ( ) Maior ( ) Super-maior

Origem: Felipe Lopes Pedroza, Lista de Verificação: Entrega 6 (protótipo de papel, relato e planejamento da avaliação), item 8.

Item 82: O método de avaliação está identificado como de investigação (questionários, entrevistas, grupos de foco, estudos de campo), de inspeção (o avaliador examina a solução para antever problemas) ou de observação de uso?

Fonte: Página: 272, Capítulo 11. Planejamento da Avaliação de IHC (Seção 11.6 Qual Tipo de Método de Avaliação Escolher?). Livro: SIMONE DINIZ JUNQUEIRA BARBOSA, BRUNO SANTANA DA SILVA, MILENE SELBACH SILVEIRA, ISABELA GASPARINI, TICIANNE DARIN, GABRIEL DINIZ JUNQUEIRA BARBOSA, Interação Humano-Computador e Experiência do Usuário, 1ª ed., Rio de Janeiro, 2021.

Foto:



272 *Capítulo 11. Planejamento da Avaliação de IHC*

ser coletados, como eles devem ser analisados, e quais critérios de qualidade de uso (usabilidade, experiência do usuário, acessibilidade ou communicability) sua análise privilegia. Os métodos de avaliação de IHC podem ser classificados em: métodos de investigação, de observação de uso e de inspeção.

Os métodos de **investigação** (*inquiry*) envolvem o uso de questionários, a realização de entrevistas, grupos de foco e estudos de campo, entre outros. Esses métodos permitem ao avaliador ter acesso, interpretar e analisar concepções, opiniões, expectativas e comportamentos do usuário relacionados com sistemas interativos. Em particular, permitem investigar alternativas de design, problemas que os usuários costumam enfrentar, como eles se apropriaram da tecnologia existente e quais são suas expectativas para futuras interações com tecnologias atuais e novas. São frequentemente utilizados em etapas iniciais do processo de design, para ratificar ou retificar o entendimento da situação atual e identificar necessidades e oportunidades de intervenção (*i.e.*, em análise), bem como explorar formas alternativas de intervenção (*i.e.*, em avaliação formativa). Também são utilizados para avaliar a introdução da nova tecnologia (*i.e.*, em avaliação somativa). Em geral, os métodos de avaliação através de investigação não exigem que os usuários utilizem um sistema interativo durante a coleta de dados. No entanto, o uso de um sistema ou de materiais de apoio, como imagens, cenários e outros tipos de artefatos, pode contribuir para a investigação, ajudando os usuários a se lembrarem de suas experiências e expectativas e também a manterem o foco nas questões de interesse. Os dados obtidos através de investigação com usuários e demais *stakeholders* podem ser coletados através das técnicas discutidas no Capítulo 7, tais como: entrevista, questionário, grupo de foco, estudo de campo e investigação contextual. Vale observar que a qualidade dos dados coletados através de métodos de investigação, como entrevistas, depende fortemente da habilidade do participante em articular suas ideias e em se lembrar de acontecimentos relevantes. Sempre que possível, recomendamos integrar a investigação à observação, de forma a obter dados mais robustos e um entendimento profundo das *questões sob investigação*.

Os métodos de **inspeção** permitem ao avaliador examinar (ou inspecionar) uma solução de IHC para tentar *antever* as possíveis consequências de certas decisões de design sobre as experiências de uso. Em outras palavras, tentar identificar problemas que os usuários podem vir a ter quando interagirem com o sistema. Esses métodos geralmente não envolvem diretamente usuários e, portanto, tratam de experiências de uso preferenciais, e não reais. Além de permitir comparar designs alternativos e buscar problemas em soluções de IHC, os métodos de inspeção permitem ainda avaliar a conformidade com um padrão ou guia de estilo.

Ao inspecionar uma interface, os avaliadores tentam se colocar no lugar de um usuário com determinado perfil, com um certo conhecimento e experiência em algumas atividades. Mas existe um limite na empatia do avaliador; de fato, ele não é o usuário. Ele pode deixar de encontrar problemas que os usuários teriam, e também pode julgar como problemáticos pontos que não causariam dificuldades aos usuários. Além disso, um avaliador pode se concentrar mais em alguns aspectos de usabilidade do que em outros (Nielsen, 1994c). Sempre que possível, devemos buscar dados de diferentes fontes e métodos de avaliação para fazer uma apreciação mais robusta do sistema em questão. Os métodos de avaliação através de inspeção, também denominados métodos analíticos, podem ser utilizados ao longo de todo o processo de design, à medida que modelos ou protótipos são elaborados. Costumam ser mais rápidos e ter um custo menor do que métodos que envolvem usuários. Alguns métodos de inspeção são descritos na Seção 12.1.

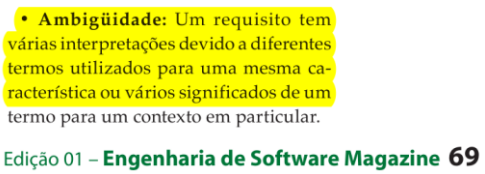
Os métodos de **observação** fornecem dados sobre situações em que os usuários realizam suas atividades, com ou sem apoio de sistemas interativos. Através do registro dos dados observados, esses métodos permitem identificar problemas reais que os usuários enfrentaram durante sua experiência de uso do sistema sendo avaliado. O avaliador pode observar os usuários em contexto ou em laboratório.

Resposta: ( ) Conforme ( ) Não conforme ( ) Não se aplica

Critério de Fagan: se a resposta for Não conforme, o defeito é do tipo Ambiguidade: o artefato admite mais de uma interpretação.

Fonte: Página: 69, Definição dos Conceitos (taxonomia de defeitos). Artigo: MARCOS KALINOWSKI, RODRIGO OLIVEIRA SPÍNOLA, Introdução à Inspeção de Software, Engenharia de Software Magazine, edição 01, p. 68 a 74.

Foto:



• **Ambigüidade:** Um requisito tem várias interpretações devido a diferentes termos utilizados para uma mesma característica ou vários significados de um termo para um contexto em particular.

Edição 01 – **Engenharia de Software Magazine** **69**

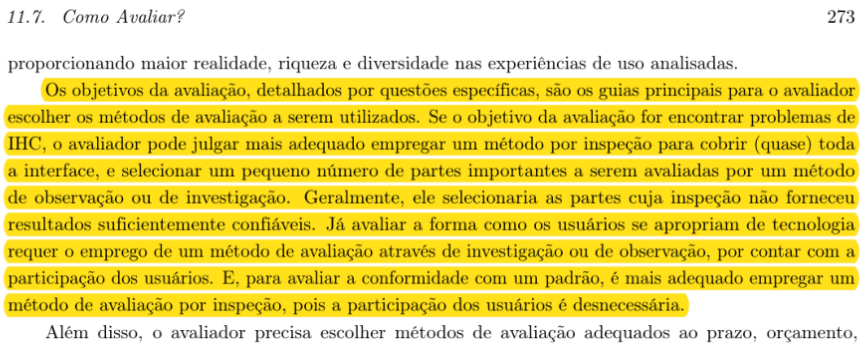
Severidade (Doolan, p. 178): ( ) Menor ( ) Maior ( ) Super-maior

Origem: Matheus do Vale Lameira, Lista de Verificação: Storyboard, item 29.

Item 83: O método escolhido é coerente com o objetivo: inspeção para cobrir quase toda a interface em busca de problemas; investigação ou observação, com usuários, para avaliar a apropriação de tecnologia; inspeção para a conformidade com um padrão?

Fonte: Página: 273, Capítulo 11. Planejamento da Avaliação de IHC (Seção 11.6 Qual Tipo de Método de Avaliação Escolher?). Livro: SIMONE DINIZ JUNQUEIRA BARBOSA, BRUNO SANTANA DA SILVA, MILENE SELBACH SILVEIRA, ISABELA GASPARINI, TICIANNE DARIN, GABRIEL DINIZ JUNQUEIRA BARBOSA, Interação Humano-Computador e Experiência do Usuário, 1ª ed., Rio de Janeiro, 2021.

Foto:



11.7. Como Avaliar? 273

proporcionando maior realidade, riqueza e diversidade nas experiências de uso analisadas.

Os objetivos da avaliação, detalhados por questões específicas, são os guias principais para o avaliador escolher os métodos de avaliação a serem utilizados. Se o objetivo da avaliação for encontrar problemas de IHC, o avaliador pode julgar mais adequado empregar um método por inspeção para cobrir (quase) toda a interface, e selecionar um pequeno número de partes importantes a serem avaliadas por um método de observação ou de investigação. Geralmente, ele selecionaria as partes cuja inspeção não forneceu resultados suficientemente confiáveis. Já avaliar a forma como os usuários se apropriam de tecnologia requer o emprego de um método de avaliação através de investigação ou de observação, por contar com a participação dos usuários. E, para avaliar a conformidade com um padrão, é mais adequado empregar um método de avaliação por inspeção, pois a participação dos usuários é desnecessária.

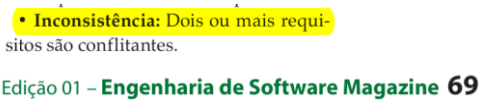
Além disso, o avaliador precisa escolher métodos de avaliação adequados ao prazo, orçamento,

Resposta: ( ) Conforme ( ) Não conforme ( ) Não se aplica

Critério de Fagan: se a resposta for Não conforme, o defeito é do tipo Inconsistência: partes do artefato, ou o artefato e outro artefato do projeto, são conflitantes.

Fonte: Página: 69, Definição dos Conceitos (taxonomia de defeitos). Artigo: MARCOS KALINOWSKI, RODRIGO OLIVEIRA SPÍNOLA, Introdução à Inspeção de Software, Engenharia de Software Magazine, edição 01, p. 68 a 74.

Foto:



• **Inconsistência:** Dois ou mais requisitos são conflitantes.

Edição 01 – **Engenharia de Software Magazine** **69**

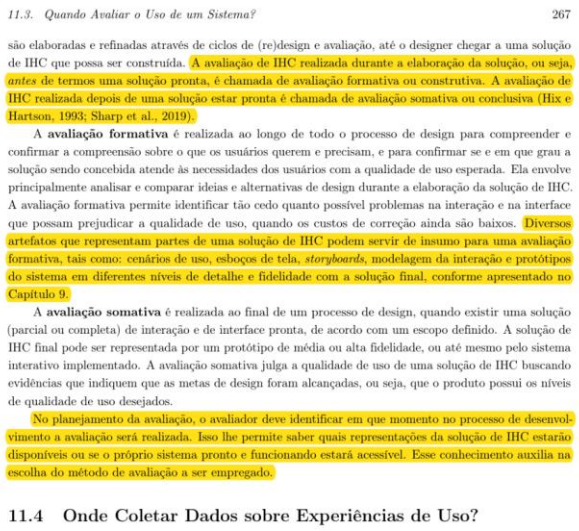
Severidade (Doolan, p. 178): ( ) Menor ( ) Maior ( ) Super-maior

Origem: Matheus do Vale Lameira, Lista de Verificação: Storyboard, item 30.

Item 84: O planejamento identifica em que momento do processo de design a avaliação ocorre: durante a elaboração da solução (avaliação formativa) ou depois de a solução estar pronta (avaliação somativa)?

Fonte: Página: 267, Capítulo 11. Planejamento da Avaliação de IHC (Seção 11.3 Quando Avaliar o Uso de um Sistema?). Livro: SIMONE DINIZ JUNQUEIRA BARBOSA, BRUNO SANTANA DA SILVA, MILENE SELBACH SILVEIRA, ISABELA GASPARINI, TICIANNE DARIN, GABRIEL DINIZ JUNQUEIRA BARBOSA, Interação Humano-Computador e Experiência do Usuário, 1ª ed., Rio de Janeiro, 2021.

Foto:



11.3. Quando Avaliar o Uso de um Sistema? 267

são elaboradas e refinadas através de ciclos de (re)design e avaliação, até o designer chegar a uma solução de IHC que possa ser construída. A avaliação de IHC realizada durante a elaboração da solução, ou seja, *antes* de termos uma solução pronta, é chamada de avaliação formativa ou construtiva. A avaliação de IHC realizada depois de uma solução estar pronta é chamada de avaliação somativa ou conclusiva (Hix e Hartson, 1993; Sharp et al., 2019).

A **avaliação formativa** é realizada ao longo de todo o processo de design para compreender e confirmar a compreensão sobre o que os usuários querem e precisam, e para confirmar se e em que grau a solução sendo concebida atende às necessidades dos usuários com a qualidade de uso esperada. Ela envolve principalmente analisar e comparar ideias e alternativas de design durante a elaboração da solução de IHC. A avaliação formativa permite identificar tão cedo quanto possível problemas na interação e na interface que possam prejudicar a qualidade de uso, quando os custos de correção ainda são baixos. Diversos artefatos que representam partes de uma solução de IHC podem servir de insumo para uma avaliação formativa, tais como: cenários de uso, esboços de tela, *storyboards*, modelagem da interação e protótipos do sistema em diferentes níveis de detalhe e fidelidade com a solução final, conforme apresentado no Capítulo 9.

A **avaliação somativa** é realizada ao final de um processo de design, quando existir uma solução (parcial ou completa) de interação e de interface pronta, de acordo com um escopo definido. A solução de IHC final pode ser representada por um protótipo de média ou alta fidelidade, ou até mesmo pelo sistema interativo implementado. A avaliação somativa julga a qualidade de uso de uma solução de IHC buscando evidências que indiquem que as metas de design foram alcançadas, ou seja, que o produto possui os níveis de qualidade de uso desejados.

No planejamento da avaliação, o avaliador deve identificar em que momento no processo de desenvolvimento a avaliação será realizada. Isso lhe permite saber quais representações da solução de IHC estarão disponíveis ou se o próprio sistema pronto e funcionando estará acessível. Esse conhecimento auxilia na escolha do método de avaliação a ser empregado.

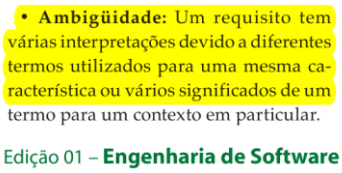
11.4 Onde Coletar Dados sobre Experiências de Uso?

Resposta: ( ) Conforme ( ) Não conforme ( ) Não se aplica

Critério de Fagan: se a resposta for Não conforme, o defeito é do tipo Ambiguidade: o artefato admite mais de uma interpretação.

Fonte: Página: 69, Definição dos Conceitos (taxonomia de defeitos). Artigo: MARCOS KALINOWSKI, RODRIGO OLIVEIRA SPÍNOLA, Introdução à Inspeção de Software, Engenharia de Software Magazine, edição 01, p. 68 a 74.

Foto:



• **Ambigüidade:** Um requisito tem várias interpretações devido a diferentes termos utilizados para uma mesma característica ou vários significados de um termo para um contexto em particular.

Edição 01 – **Engenharia de Software Magazine** **69**

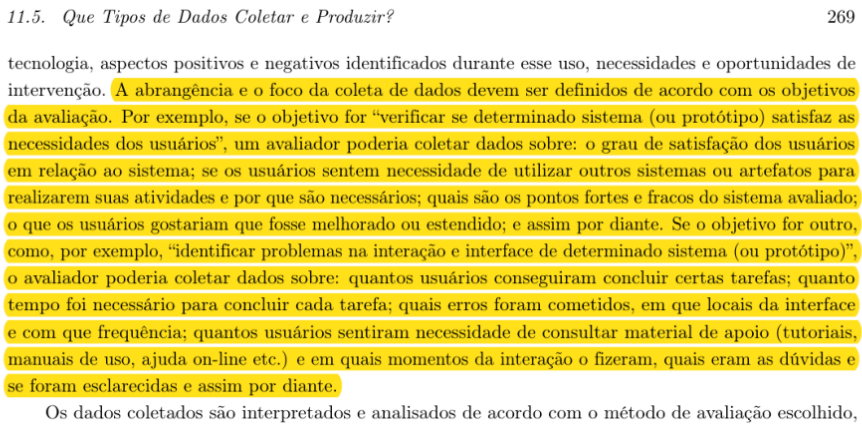
Severidade (Doolan, p. 178): ( ) Menor ( ) Maior ( ) Super-maior

Origem: Matheus do Vale Lameira, Lista de Verificação: Storyboard, item 31.

Item 85: O planejamento define os dados a coletar de acordo com os objetivos (satisfação, pontos fortes e fracos, tarefas concluídas, tempo, erros e consultas a material de apoio)?

Fonte: Página: 269, Capítulo 11. Planejamento da Avaliação de IHC (Seção 11.5 Que Tipos de Dados Coletar e Produzir?). Livro: SIMONE DINIZ JUNQUEIRA BARBOSA, BRUNO SANTANA DA SILVA, MILENE SELBACH SILVEIRA, ISABELA GASPARINI, TICIANNE DARIN, GABRIEL DINIZ JUNQUEIRA BARBOSA, Interação Humano-Computador e Experiência do Usuário, 1ª ed., Rio de Janeiro, 2021.

Foto:



11.5. Que Tipos de Dados Coletar e Produzir? 269

tecnologia, aspectos positivos e negativos identificados durante esse uso, necessidades e oportunidades de intervenção. A abrangência e o foco da coleta de dados devem ser definidos de acordo com os objetivos da avaliação. Por exemplo, se o objetivo for "verificar se determinado sistema (ou protótipo) satisfaz as necessidades dos usuários", um avaliador poderia coletar dados sobre: o grau de satisfação dos usuários em relação ao sistema; se os usuários sentem necessidade de utilizar outros sistemas ou artefatos para realizarem suas atividades e por que são necessários; quais são os pontos fortes e fracos do sistema avaliado; o que os usuários gostariam que fosse melhorado ou estendido; e assim por diante. Se o objetivo for outro, como, por exemplo, "identificar problemas na interação e interface de determinado sistema (ou protótipo)", o avaliador poderia coletar dados sobre: quantos usuários conseguiram concluir certas tarefas; quanto tempo foi necessário para concluir cada tarefa; quais erros foram cometidos, em que locais da interface e com que frequência; quantos usuários sentiram necessidade de consultar material de apoio (tutoriais, manuais de uso, ajuda on-line etc.) e em quais momentos da interação o fizeram, quais eram as dúvidas e se foram esclarecidas e assim por diante.

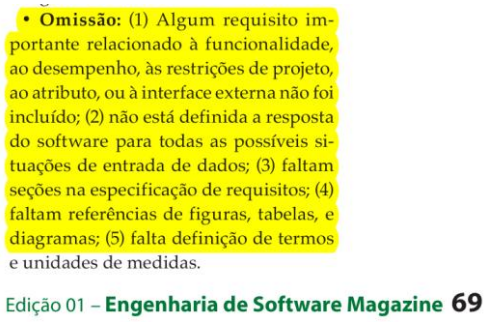
Os dados coletados são interpretados e analisados de acordo com o método de avaliação escolhido,

Resposta: ( ) Conforme ( ) Não conforme ( ) Não se aplica

Critério de Fagan: se a resposta for Não conforme, o defeito é do tipo Omissão: falta no artefato um elemento importante que o item exige.

Fonte: Página: 69, Definição dos Conceitos (taxonomia de defeitos). Artigo: MARCOS KALINOWSKI, RODRIGO OLIVEIRA SPÍNOLA, Introdução à Inspeção de Software, Engenharia de Software Magazine, edição 01, p. 68 a 74.

Foto:



• **Omissão:** (1) Algum requisito importante relacionado à funcionalidade, ao desempenho, às restrições de projeto, ao atributo, ou à interface externa não foi incluído; (2) não está definida a resposta do software para todas as possíveis situações de entrada de dados; (3) faltam seções na especificação de requisitos; (4) faltam referências de figuras, tabelas, e diagramas; (5) falta definição de termos e unidades de medidas.

Edição 01 – **Engenharia de Software Magazine** **69**

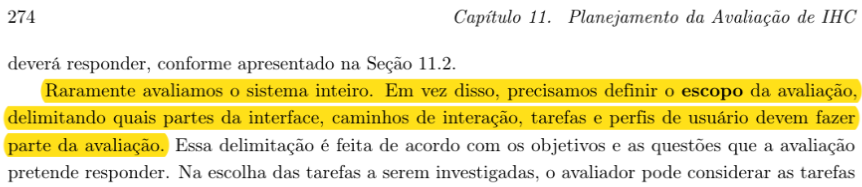
Severidade (Doolan, p. 178): ( ) Menor ( ) Maior ( ) Super-maior

Origem: Matheus do Vale Lameira, Lista de Verificação: Storyboard, item 27.

Item 86: O planejamento define o escopo da avaliação: quais partes da interface, caminhos de interação, tarefas e perfis de usuário serão avaliados, de acordo com os objetivos e as perguntas?

Fonte: Página: 274, Capítulo 11. Planejamento da Avaliação de IHC (Seção 11.7.2 Preparação). Livro: SIMONE DINIZ JUNQUEIRA BARBOSA, BRUNO SANTANA DA SILVA, MILENE SELBACH SILVEIRA, ISABELA GASPARINI, TICIANNE DARIN, GABRIEL DINIZ JUNQUEIRA BARBOSA, Interação Humano-Computador e Experiência do Usuário, 1ª ed., Rio de Janeiro, 2021.

Foto:



274 *Capítulo 11. Planejamento da Avaliação de IHC*

deverá responder, conforme apresentado na Seção 11.2.

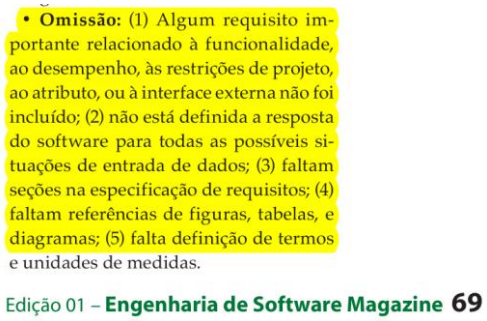
Raramente avaliamos o sistema inteiro. Em vez disso, precisamos definir o **escopo** da avaliação, delimitando quais partes da interface, caminhos de interação, tarefas e perfis de usuário devem fazer parte da avaliação. Essa delimitação é feita de acordo com os objetivos e as questões que a avaliação pretende responder. Na escolha das tarefas a serem investigadas, o avaliador pode considerar as tarefas

Resposta: ( ) Conforme ( ) Não conforme ( ) Não se aplica

Critério de Fagan: se a resposta for Não conforme, o defeito é do tipo Omissão: falta no artefato um elemento importante que o item exige.

Fonte: Página: 69, Definição dos Conceitos (taxonomia de defeitos). Artigo: MARCOS KALINOWSKI, RODRIGO OLIVEIRA SPÍNOLA, Introdução à Inspeção de Software, Engenharia de Software Magazine, edição 01, p. 68 a 74.

Foto:



• **Omissão:** (1) Algum requisito importante relacionado à funcionalidade, ao desempenho, às restrições de projeto, ao atributo, ou à interface externa não foi incluído; (2) não está definida a resposta do software para todas as possíveis situações de entrada de dados; (3) faltam seções na especificação de requisitos; (4) faltam referências de figuras, tabelas, e diagramas; (5) falta definição de termos e unidades de medidas.

Edição 01 – **Engenharia de Software Magazine** **69**

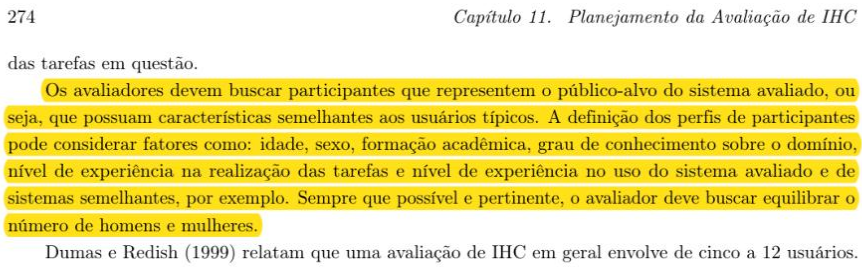
Severidade (Doolan, p. 178): ( ) Menor ( ) Maior ( ) Super-maior

Origem: Matheus do Vale Lameira, Lista de Verificação: Storyboard, item 26.

Item 87: O perfil dos participantes representa o público-alvo, considerando idade, sexo, formação, conhecimento do domínio e experiência nas tarefas e em sistemas semelhantes?

Fonte: Página: 274, Capítulo 11. Planejamento da Avaliação de IHC (Seção 11.7.2 Preparação). Livro: SIMONE DINIZ JUNQUEIRA BARBOSA, BRUNO SANTANA DA SILVA, MILENE SELBACH SILVEIRA, ISABELA GASPARINI, TICIANNE DARIN, GABRIEL DINIZ JUNQUEIRA BARBOSA, Interação Humano-Computador e Experiência do Usuário, 1ª ed., Rio de Janeiro, 2021.

Foto:



274 *Capítulo 11. Planejamento da Avaliação de IHC*

das tarefas em questão.

Os avaliadores devem buscar participantes que representem o público-alvo do sistema avaliado, ou seja, que possuam características semelhantes aos usuários típicos. A definição dos perfis de participantes pode considerar fatores como: idade, sexo, formação acadêmica, grau de conhecimento sobre o domínio, nível de experiência na realização das tarefas e nível de experiência no uso do sistema avaliado e de sistemas semelhantes, por exemplo. Sempre que possível e pertinente, o avaliador deve buscar equilibrar o número de homens e mulheres.

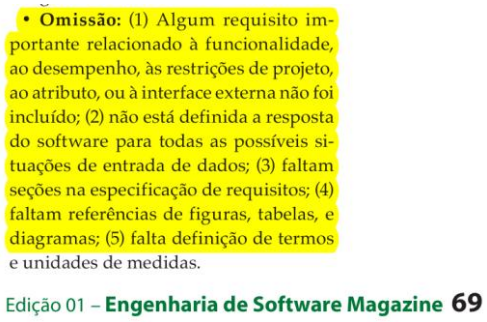
Dumas e Redish (1999) relatam que uma avaliação de IHC em geral envolve de cinco a 12 usuários.

Resposta: ( ) Conforme ( ) Não conforme ( ) Não se aplica

Critério de Fagan: se a resposta for Não conforme, o defeito é do tipo Omissão: falta no artefato um elemento importante que o item exige.

Fonte: Página: 69, Definição dos Conceitos (taxonomia de defeitos). Artigo: MARCOS KALINOWSKI, RODRIGO OLIVEIRA SPÍNOLA, Introdução à Inspeção de Software, Engenharia de Software Magazine, edição 01, p. 68 a 74.

Foto:



• **Omissão:** (1) Algum requisito importante relacionado à funcionalidade, ao desempenho, às restrições de projeto, ao atributo, ou à interface externa não foi incluído; (2) não está definida a resposta do software para todas as possíveis situações de entrada de dados; (3) faltam seções na especificação de requisitos; (4) faltam referências de figuras, tabelas, e diagramas; (5) falta definição de termos e unidades de medidas.

Edição 01 – **Engenharia de Software Magazine** **69**

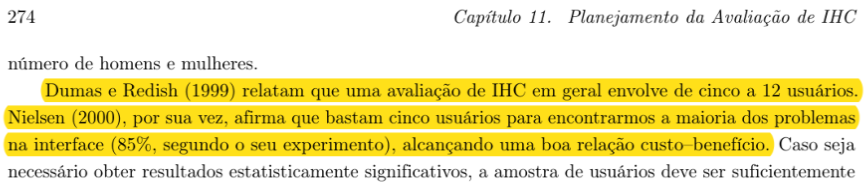
Severidade (Doolan, p. 178): ( ) Menor ( ) Maior ( ) Super-maior

Origem: Matheus do Vale Lameira, Lista de Verificação: Storyboard, item 33.

Item 88: O número de participantes está definido e justificado (por exemplo, de cinco a 12 usuários, ou cinco usuários para encontrar a maioria dos problemas da interface)?

Fonte: Página: 274, Capítulo 11. Planejamento da Avaliação de IHC (Seção 11.7.2 Preparação). Livro: SIMONE DINIZ JUNQUEIRA BARBOSA, BRUNO SANTANA DA SILVA, MILENE SELBACH SILVEIRA, ISABELA GASPARINI, TICIANNE DARIN, GABRIEL DINIZ JUNQUEIRA BARBOSA, Interação Humano-Computador e Experiência do Usuário, 1ª ed., Rio de Janeiro, 2021.

Foto:



274 *Capítulo 11. Planejamento da Avaliação de IHC*

número de homens e mulheres.

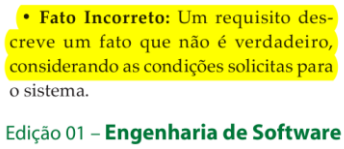
Dumas e Redish (1999) relatam que uma avaliação de IHC em geral envolve de cinco a 12 usuários. Nielsen (2000), por sua vez, afirma que bastam cinco usuários para encontrarmos a maioria dos problemas na interface (85%, segundo o seu experimento), alcançando uma boa relação custo–benefício. Caso seja necessário obter resultados estatisticamente significativos, a amostra de usuários deve ser suficientemente

Resposta: ( ) Conforme ( ) Não conforme ( ) Não se aplica

Critério de Fagan: se a resposta for Não conforme, o defeito é do tipo Fato incorreto: o artefato descreve algo que não é verdadeiro nas condições do projeto.

Fonte: Página: 69, Definição dos Conceitos (taxonomia de defeitos). Artigo: MARCOS KALINOWSKI, RODRIGO OLIVEIRA SPÍNOLA, Introdução à Inspeção de Software, Engenharia de Software Magazine, edição 01, p. 68 a 74.

Foto:



• **Fato Incorreto:** Um requisito descreve um fato que não é verdadeiro, considerando as condições solicitas para o sistema.

Edição 01 – **Engenharia de Software Magazine** **69**

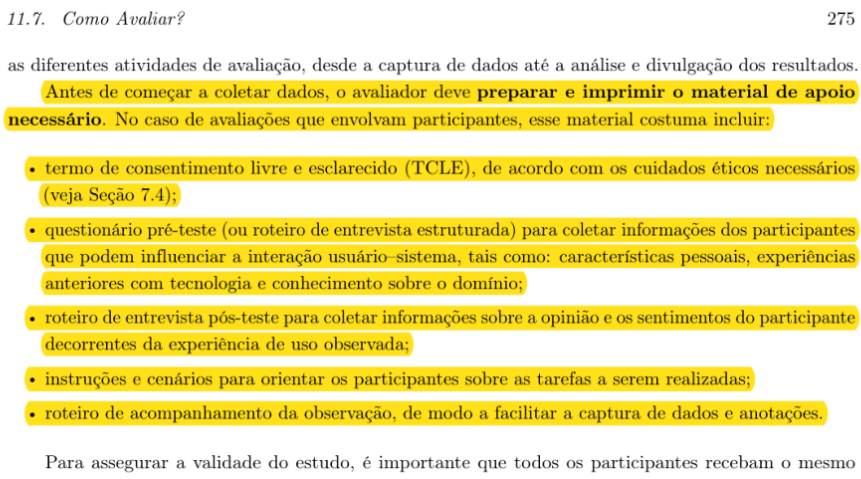
Severidade (Doolan, p. 178): ( ) Menor ( ) Maior ( ) Super-maior

Origem: Matheus do Vale Lameira, Lista de Verificação: Storyboard, item 34.

Item 89: O planejamento prevê o material de apoio: termo de consentimento livre e esclarecido, questionário pré-teste, roteiro de entrevista pós-teste, instruções e cenários das tarefas e roteiro de acompanhamento da observação?

Fonte: Página: 275, Capítulo 11. Planejamento da Avaliação de IHC (Seção 11.7.2 Preparação). Livro: SIMONE DINIZ JUNQUEIRA BARBOSA, BRUNO SANTANA DA SILVA, MILENE SELBACH SILVEIRA, ISABELA GASPARINI, TICIANNE DARIN, GABRIEL DINIZ JUNQUEIRA BARBOSA, Interação Humano-Computador e Experiência do Usuário, 1ª ed., Rio de Janeiro, 2021.

Foto:



11.7. Como Avaliar? 275

as diferentes atividades de avaliação, desde a captura de dados até a análise e divulgação dos resultados.

Antes de começar a coletar dados, o avaliador deve **preparar e imprimir o material de apoio necessário**. No caso de avaliações que envolvam participantes, esse material costuma incluir:

- termo de consentimento livre e esclarecido (TCLE), de acordo com os cuidados éticos necessários (veja Seção 7.4);
- questionário pré-teste (ou roteiro de entrevista estruturada) para coletar informações dos participantes que podem influenciar a interação usuário–sistema, tais como: características pessoais, experiências anteriores com tecnologia e conhecimento sobre o domínio;
- roteiro de entrevista pós-teste para coletar informações sobre a opinião e os sentimentos do participante decorrentes da experiência de uso observada;
- instruções e cenários para orientar os participantes sobre as tarefas a serem realizadas;
- roteiro de acompanhamento da observação, de modo a facilitar a captura de dados e anotações.

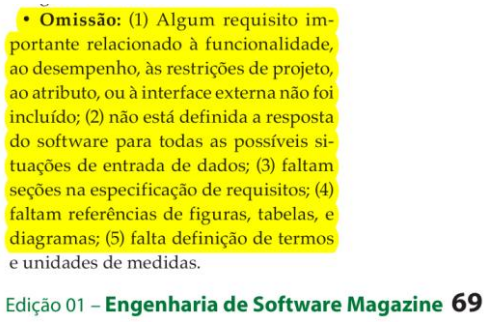
Para assegurar a validade do estudo, é importante que todos os participantes recebam o mesmo

Resposta: ( ) Conforme ( ) Não conforme ( ) Não se aplica

Critério de Fagan: se a resposta for Não conforme, o defeito é do tipo Omissão: falta no artefato um elemento importante que o item exige.

Fonte: Página: 69, Definição dos Conceitos (taxonomia de defeitos). Artigo: MARCOS KALINOWSKI, RODRIGO OLIVEIRA SPÍNOLA, Introdução à Inspeção de Software, Engenharia de Software Magazine, edição 01, p. 68 a 74.

Foto:



- **Omissão:** (1) Algum requisito importante relacionado à funcionalidade, ao desempenho, às restrições de projeto, ao atributo, ou à interface externa não foi incluído; (2) não está definida a resposta do software para todas as possíveis situações de entrada de dados; (3) faltam seções na especificação de requisitos; (4) faltam referências de figuras, tabelas, e diagramas; (5) falta definição de termos e unidades de medidas.

Edição 01 – **Engenharia de Software Magazine** **69**

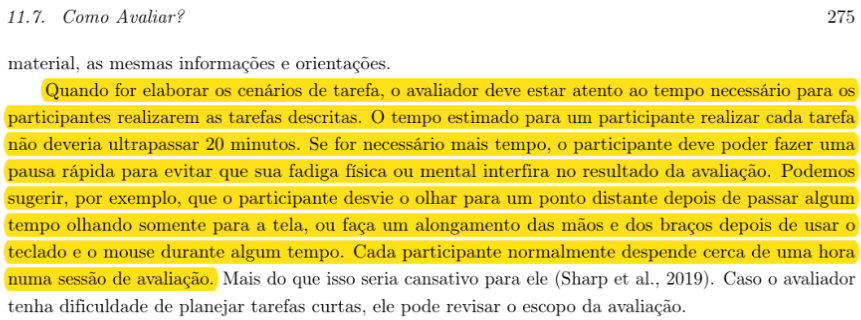
Severidade (Doolan, p. 178): ( ) Menor ( ) Maior ( ) Super-maior

Origem: Matheus do Vale Lameira, Lista de Verificação: Storyboard, item 35.

Item 90: O tempo planejado é compatível com o recomendado: cada tarefa com no máximo 20 minutos e cerca de uma hora por participante em cada sessão?

Fonte: Página: 275, Capítulo 11. Planejamento da Avaliação de IHC (Seção 11.7.2 Preparação). Livro: SIMONE DINIZ JUNQUEIRA BARBOSA, BRUNO SANTANA DA SILVA, MILENE SELBACH SILVEIRA, ISABELA GASPARINI, TICIANNE DARIN, GABRIEL DINIZ JUNQUEIRA BARBOSA, Interação Humano-Computador e Experiência do Usuário, 1ª ed., Rio de Janeiro, 2021.

Foto:



11.7. Como Avaliar? 275

material, as mesmas informações e orientações.

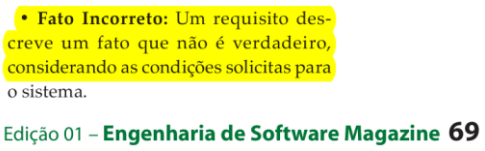
Quando for elaborar os cenários de tarefa, o avaliador deve estar atento ao tempo necessário para os participantes realizarem as tarefas descritas. O tempo estimado para um participante realizar cada tarefa não deveria ultrapassar 20 minutos. Se for necessário mais tempo, o participante deve poder fazer uma pausa rápida para evitar que sua fadiga física ou mental interfira no resultado da avaliação. Podemos sugerir, por exemplo, que o participante desvie o olhar para um ponto distante depois de passar algum tempo olhando somente para a tela, ou faça um alongamento das mãos e dos braços depois de usar o teclado e o mouse durante algum tempo. Cada participante normalmente despende cerca de uma hora numa sessão de avaliação. Mais do que isso seria cansativo para ele (Sharp et al., 2019). Caso o avaliador tenha dificuldade de planejar tarefas curtas, ele pode revisar o escopo da avaliação.

Resposta: ( ) Conforme ( ) Não conforme ( ) Não se aplica

Critério de Fagan: se a resposta for Não conforme, o defeito é do tipo Fato incorreto: o artefato descreve algo que não é verdadeiro nas condições do projeto.

Fonte: Página: 69, Definição dos Conceitos (taxonomia de defeitos). Artigo: MARCOS KALINOWSKI, RODRIGO OLIVEIRA SPÍNOLA, Introdução à Inspeção de Software, Engenharia de Software Magazine, edição 01, p. 68 a 74.

Foto:



• **Fato Incorreto:** Um requisito descreve um fato que não é verdadeiro, considerando as condições solicitas para o sistema.

Edição 01 – **Engenharia de Software Magazine** **69**

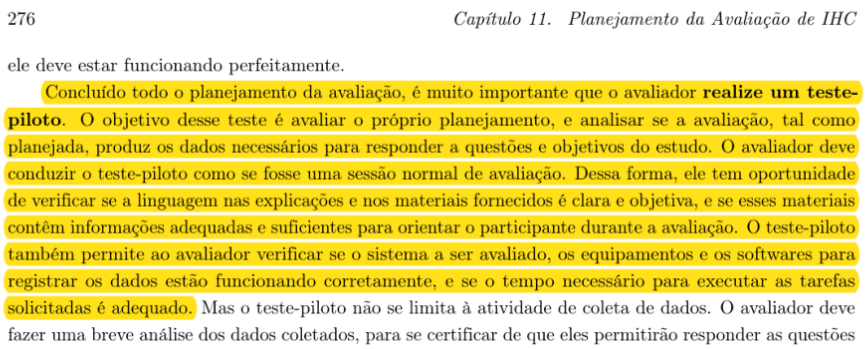
Severidade (Doolan, p. 178): ( ) Menor ( ) Maior ( ) Super-maior

Origem: Matheus do Vale Lameira, Lista de Verificação: Storyboard, item 36.

Item 91: Está previsto um teste-piloto, conduzido como uma sessão normal, para verificar se a linguagem e os materiais são claros, se o sistema e os equipamentos funcionam e se o tempo das tarefas é adequado?

Fonte: Página: 276, Capítulo 11. Planejamento da Avaliação de IHC (Seção 11.7.2 Preparação). Livro: SIMONE DINIZ JUNQUEIRA BARBOSA, BRUNO SANTANA DA SILVA, MILENE SELBACH SILVEIRA, ISABELA GASPARINI, TICIANNE DARIN, GABRIEL DINIZ JUNQUEIRA BARBOSA, Interação Humano-Computador e Experiência do Usuário, 1ª ed., Rio de Janeiro, 2021.

Foto:



276 *Capítulo 11. Planejamento da Avaliação de IHC*

ele deve estar funcionando perfeitamente.

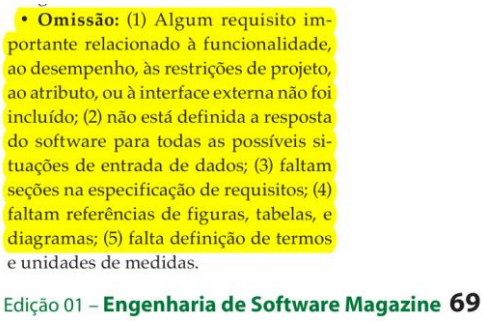
Concluído todo o planejamento da avaliação, é muito importante que o avaliador **realize um teste-piloto**. O objetivo desse teste é avaliar o próprio planejamento, e analisar se a avaliação, tal como planejada, produz os dados necessários para responder a questões e objetivos do estudo. O avaliador deve conduzir o teste-piloto como se fosse uma sessão normal de avaliação. Dessa forma, ele tem oportunidade de verificar se a linguagem nas explicações e nos materiais fornecidos é clara e objetiva, e se esses materiais contêm informações adequadas e suficientes para orientar o participante durante a avaliação. O teste-piloto também permite ao avaliador verificar se o sistema a ser avaliado, os equipamentos e os softwares para registrar os dados estão funcionando corretamente, e se o tempo necessário para executar as tarefas solicitadas é adequado. Mas o teste-piloto não se limita à atividade de coleta de dados. O avaliador deve fazer uma breve análise dos dados coletados, para se certificar de que eles permitirão responder as questões

Resposta: ( ) Conforme ( ) Não conforme ( ) Não se aplica

Critério de Fagan: se a resposta for Não conforme, o defeito é do tipo Omissão: falta no artefato um elemento importante que o item exige.

Fonte: Página: 69, Definição dos Conceitos (taxonomia de defeitos). Artigo: MARCOS KALINOWSKI, RODRIGO OLIVEIRA SPÍNOLA, Introdução à Inspeção de Software, Engenharia de Software Magazine, edição 01, p. 68 a 74.

Foto:



• **Omissão:** (1) Algum requisito importante relacionado à funcionalidade, ao desempenho, às restrições de projeto, ao atributo, ou à interface externa não foi incluído; (2) não está definida a resposta do software para todas as possíveis situações de entrada de dados; (3) faltam seções na especificação de requisitos; (4) faltam referências de figuras, tabelas, e diagramas; (5) falta definição de termos e unidades de medidas.

Edição 01 – **Engenharia de Software Magazine** **69**

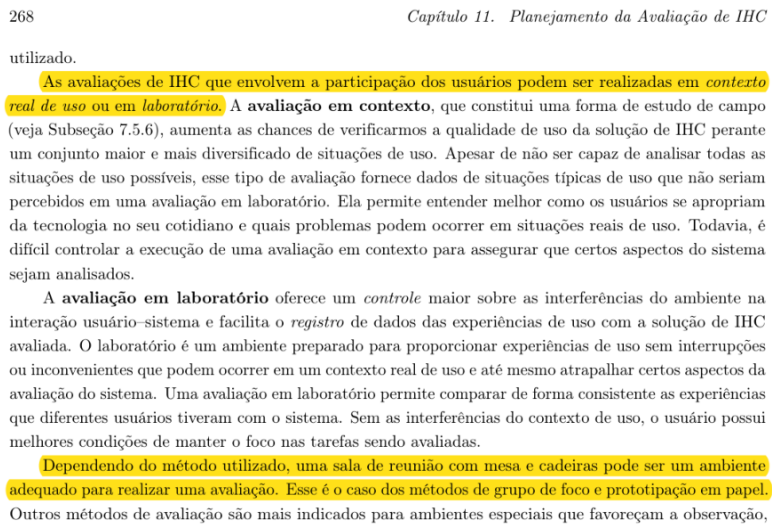
Severidade (Doolan, p. 178): ( ) Menor ( ) Maior ( ) Super-maior

Origem: Matheus do Vale Lameira, Lista de Verificação: Storyboard, item 37.

Item 92: O planejamento define o local da avaliação, em contexto real de uso ou em laboratório, adequado ao método escolhido?

Fonte: Página: 268, Capítulo 11. Planejamento da Avaliação de IHC (Seção 11.4 Onde Coletar Dados sobre Experiências de Uso?). Livro: SIMONE DINIZ JUNQUEIRA BARBOSA, BRUNO SANTANA DA SILVA, MILENE SELBACH SILVEIRA, ISABELA GASPARINI, TICIANNE DARIN, GABRIEL DINIZ JUNQUEIRA BARBOSA, Interação Humano-Computador e Experiência do Usuário, 1ª ed., Rio de Janeiro, 2021.

Foto:



268 *Capítulo 11. Planejamento da Avaliação de IHC*

utilizado.

As avaliações de IHC que envolvem a participação dos usuários podem ser realizadas em *contexto real de uso* ou em *laboratório*. A **avaliação em contexto**, que constitui uma forma de estudo de campo (veja Subseção 7.5.6), aumenta as chances de verificarmos a qualidade de uso da solução de IHC perante um conjunto maior e mais diversificado de situações de uso. Apesar de não ser capaz de analisar todas as situações de uso possíveis, esse tipo de avaliação fornece dados de situações típicas de uso que não seriam percebidos em uma avaliação em laboratório. Ela permite entender melhor como os usuários se apropriam da tecnologia no seu cotidiano e quais problemas podem ocorrer em situações reais de uso. Todavia, é difícil controlar a execução de uma avaliação em contexto para assegurar que certos aspectos do sistema sejam analisados.

A **avaliação em laboratório** oferece um *controle* maior sobre as interferências do ambiente na interação usuário–sistema e facilita o *registro* de dados das experiências de uso com a solução de IHC avaliada. O laboratório é um ambiente preparado para proporcionar experiências de uso sem interrupções ou inconvenientes que podem ocorrer em um contexto real de uso e até mesmo atrapalhar certos aspectos da avaliação do sistema. Uma avaliação em laboratório permite comparar de forma consistente as experiências que diferentes usuários tiveram com o sistema. Sem as interferências do contexto de uso, o usuário possui melhores condições de manter o foco nas tarefas sendo avaliadas.

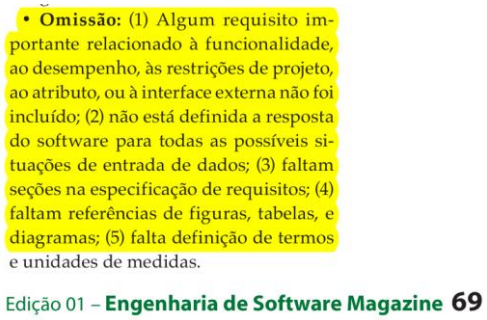
Dependendo do método utilizado, uma sala de reunião com mesa e cadeiras pode ser um ambiente adequado para realizar uma avaliação. Esse é o caso dos métodos de grupo de foco e prototipação em papel. Outros métodos de avaliação são mais indicados para ambientes especiais que favoreçam a observação,

Resposta: ( ) Conforme ( ) Não conforme ( ) Não se aplica

Critério de Fagan: se a resposta for Não conforme, o defeito é do tipo Omissão: falta no artefato um elemento importante que o item exige.

Fonte: Página: 69, Definição dos Conceitos (taxonomia de defeitos). Artigo: MARCOS KALINOWSKI, RODRIGO OLIVEIRA SPÍNOLA, Introdução à Inspeção de Software, Engenharia de Software Magazine, edição 01, p. 68 a 74.

Foto:



• **Omissão:** (1) Algum requisito importante relacionado à funcionalidade, ao desempenho, às restrições de projeto, ao atributo, ou à interface externa não foi incluído; (2) não está definida a resposta do software para todas as possíveis situações de entrada de dados; (3) faltam seções na especificação de requisitos; (4) faltam referências de figuras, tabelas, e diagramas; (5) falta definição de termos e unidades de medidas.

Edição 01 – **Engenharia de Software Magazine** **69**

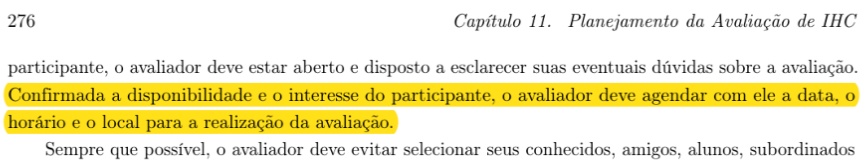
Severidade (Doolan, p. 178): ( ) Menor ( ) Maior ( ) Super-maior

Origem: Matheus do Vale Lameira, Lista de Verificação: Storyboard, item 38.

Item 93: O planejamento prevê o agendamento da data, do horário e do local com cada participante?

Fonte: Página: 276, Capítulo 11. Planejamento da Avaliação de IHC (Seção 11.7.2 Preparação). Livro: SIMONE DINIZ JUNQUEIRA BARBOSA, BRUNO SANTANA DA SILVA, MILENE SELBACH SILVEIRA, ISABELA GASPARINI, TICIANNE DARIN, GABRIEL DINIZ JUNQUEIRA BARBOSA, Interação Humano-Computador e Experiência do Usuário, 1ª ed., Rio de Janeiro, 2021.

Foto:



276 *Capítulo 11. Planejamento da Avaliação de IHC*

participante, o avaliador deve estar aberto e disposto a esclarecer suas eventuais dúvidas sobre a avaliação. Confirmada a disponibilidade e o interesse do participante, o avaliador deve agendar com ele a data, o horário e o local para a realização da avaliação.

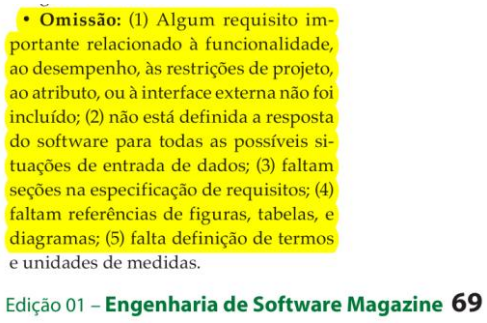
Sempre que possível, o avaliador deve evitar selecionar seus conhecidos, amigos, alunos, subordinados

Resposta: ( ) Conforme ( ) Não conforme ( ) Não se aplica

Critério de Fagan: se a resposta for Não conforme, o defeito é do tipo Omissão: falta no artefato um elemento importante que o item exige.

Fonte: Página: 69, Definição dos Conceitos (taxonomia de defeitos). Artigo: MARCOS KALINOWSKI, RODRIGO OLIVEIRA SPÍNOLA, Introdução à Inspeção de Software, Engenharia de Software Magazine, edição 01, p. 68 a 74.

Foto:



• **Omissão:** (1) Algum requisito importante relacionado à funcionalidade, ao desempenho, às restrições de projeto, ao atributo, ou à interface externa não foi incluído; (2) não está definida a resposta do software para todas as possíveis situações de entrada de dados; (3) faltam seções na especificação de requisitos; (4) faltam referências de figuras, tabelas, e diagramas; (5) falta definição de termos e unidades de medidas.

Edição 01 – **Engenharia de Software Magazine** **69**

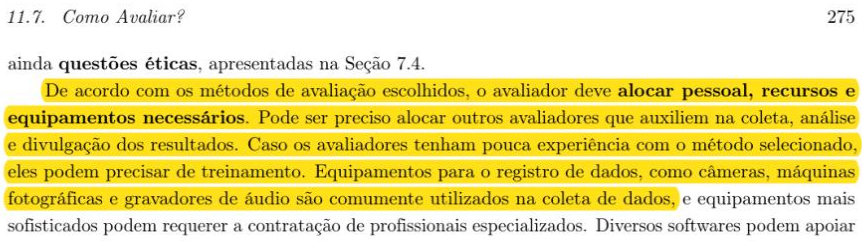
Severidade (Doolan, p. 178): ( ) Menor ( ) Maior ( ) Super-maior

Origem: Matheus do Vale Lameira, Lista de Verificação: Storyboard, item 39.

Item 94: O planejamento aloca o pessoal, os recursos e os equipamentos de registro de dados (câmeras, gravadores) de acordo com os métodos escolhidos?

Fonte: Página: 275, Capítulo 11. Planejamento da Avaliação de IHC (Seção 11.7.2 Preparação). Livro: SIMONE DINIZ JUNQUEIRA BARBOSA, BRUNO SANTANA DA SILVA, MILENE SELBACH SILVEIRA, ISABELA GASPARINI, TICIANNE DARIN, GABRIEL DINIZ JUNQUEIRA BARBOSA, Interação Humano-Computador e Experiência do Usuário, 1ª ed., Rio de Janeiro, 2021.

Foto:



11.7. Como Avaliar? 275

ainda **questões éticas**, apresentadas na Seção 7.4.

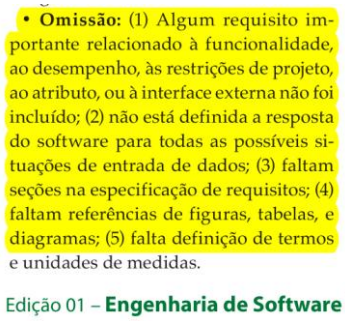
De acordo com os métodos de avaliação escolhidos, o avaliador deve **alocar pessoal, recursos e equipamentos necessários**. Pode ser preciso alocar outros avaliadores que auxiliem na coleta, análise e divulgação dos resultados. Caso os avaliadores tenham pouca experiência com o método selecionado, eles podem precisar de treinamento. Equipamentos para o registro de dados, como câmeras, máquinas fotográficas e gravadores de áudio são comumente utilizados na coleta de dados, e equipamentos mais sofisticados podem requerer a contratação de profissionais especializados. Diversos softwares podem apoiar

Resposta: ( ) Conforme ( ) Não conforme ( ) Não se aplica

Critério de Fagan: se a resposta for Não conforme, o defeito é do tipo Omissão: falta no artefato um elemento importante que o item exige.

Fonte: Página: 69, Definição dos Conceitos (taxonomia de defeitos). Artigo: MARCOS KALINOWSKI, RODRIGO OLIVEIRA SPÍNOLA, Introdução à Inspeção de Software, Engenharia de Software Magazine, edição 01, p. 68 a 74.

Foto:



• **Omissão:** (1) Algum requisito importante relacionado à funcionalidade, ao desempenho, às restrições de projeto, ao atributo, ou à interface externa não foi incluído; (2) não está definida a resposta do software para todas as possíveis situações de entrada de dados; (3) faltam seções na especificação de requisitos; (4) faltam referências de figuras, tabelas, e diagramas; (5) falta definição de termos e unidades de medidas.

Edição 01 – **Engenharia de Software Magazine** **69**

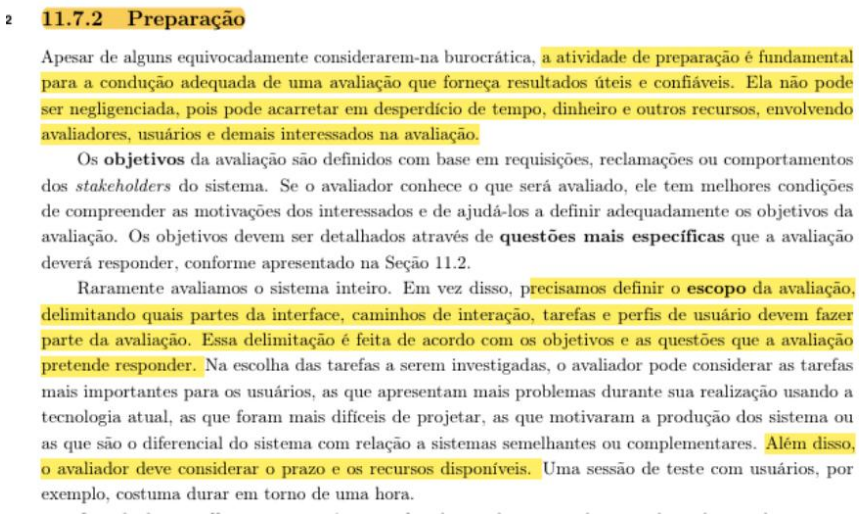
Severidade (Doolan, p. 178): ( ) Menor ( ) Maior ( ) Super-maior

Origem: Matheus do Vale Lameira, Lista de Verificação: Storyboard, item 40.

Item 95: O planejamento da avaliação tem um cronograma?

Fonte: Página: 14 de 20 (PDF disponibilizado no Aprender), Capítulo Planejamento da Avaliação de IHC. Livro: SIMONE DINIZ JUNQUEIRA BARBOSA, BRUNO SANTANA DA SILVA, Interação Humano-Computador (PDF disponibilizado no Aprender).

Foto:



11.7.2 Preparação

Apesar de alguns equivocadamente considerarem-na burocrática, a atividade de preparação é fundamental para a condução adequada de uma avaliação que forneça resultados úteis e confiáveis. Ela não pode ser negligenciada, pois pode acarretar em desperdício de tempo, dinheiro e outros recursos, envolvendo avaliadores, usuários e demais interessados na avaliação.

Os **objetivos** da avaliação são definidos com base em requisições, reclamações ou comportamentos dos *stakeholders* do sistema. Se o avaliador conhece o que será avaliado, ele tem melhores condições de compreender as motivações dos interessados e de ajudá-los a definir adequadamente os objetivos da avaliação. Os objetivos devem ser detalhados através de **questões mais específicas** que a avaliação deverá responder, conforme apresentado na Seção 11.2.

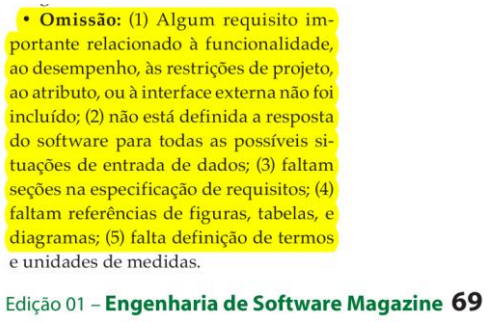
Raramente avaliamos o sistema inteiro. Em vez disso, precisamos definir o **escopo** da avaliação, delimitando quais partes da interface, caminhos de interação, tarefas e perfis de usuário devem fazer parte da avaliação. Essa delimitação é feita de acordo com os objetivos e as questões que a avaliação pretende responder. Na escolha das tarefas a serem investigadas, o avaliador pode considerar as tarefas mais importantes para os usuários, as que apresentam mais problemas durante sua realização usando a tecnologia atual, as que foram mais difíceis de projetar, as que motivaram a produção dos sistema ou as que são o diferencial do sistema com relação a sistemas semelhantes ou complementares. Além disso, o avaliador deve considerar o prazo e os recursos disponíveis. Uma sessão de teste com usuários, por exemplo, costuma durar em torno de uma hora.

Resposta: ( ) Conforme ( ) Não conforme ( ) Não se aplica

Critério de Fagan: se a resposta for Não conforme, o defeito é do tipo Omissão: falta no artefato um elemento importante que o item exige.

Fonte: Página: 69, Definição dos Conceitos (taxonomia de defeitos). Artigo: MARCOS KALINOWSKI, RODRIGO OLIVEIRA SPÍNOLA, Introdução à Inspeção de Software, Engenharia de Software Magazine, edição 01, p. 68 a 74.

Foto:



- **Omissão:** (1) Algum requisito importante relacionado à funcionalidade, ao desempenho, às restrições de projeto, ao atributo, ou à interface externa não foi incluído; (2) não está definida a resposta do software para todas as possíveis situações de entrada de dados; (3) faltam seções na especificação de requisitos; (4) faltam referências de figuras, tabelas, e diagramas; (5) falta definição de termos e unidades de medidas.

Edição 01 – **Engenharia de Software Magazine** **69**

Severidade (Doolan, p. 178): ( ) Menor ( ) Maior ( ) Super-maior

Origem: Victor Andreozzi De La Rocque Couto, Lista de Verificação: Planejamento da Avaliação (DECIDE), item 7.

Item 96: O planejamento decide como lidar com as questões éticas e prevê avaliar e interpretar os dados considerando a confiabilidade e as validades interna, externa e ecológica do estudo (D e E do DECIDE)?

Fonte: Página: 313, Capítulo 9. Planejamento da Avaliação de IHC. Livro: SIMONE DINIZ JUNQUEIRA BARBOSA, BRUNO SANTANA DA SILVA, Interação Humano-Computador, 1ª ed., Editora Campus/Elsevier, 2010.

Foto:

	necessários, os prazos e o orçamento disponíveis, além da mão-de-obra necessária para conduzir a avaliação.
D	Decidir como lidar com as questões éticas. Sempre que usuários são envolvidos numa avaliação, o avaliador deve tomar os cuidados éticos necessários (veja Seção 5.4). Os participantes da avaliação devem ser respeitados e não podem ser prejudicados direta ou indiretamente, nem durante os experimentos, nem após a divulgação dos resultados da avaliação.
E	Avaliar (*Evaluate*), interpretar e apresentar os dados. O avaliador precisa estar atento a alguns aspectos da avaliação realizada antes de tirar conclusões e divulgar resultados. Ele deve considerar: o grau de *confiabilidade* dos dados (i.e., semelhança dos resultados obtidos quando emprega mais de uma vez o mesmo método de avaliação nas mesmas circunstâncias; a *validade interna* do estudo (i.e., se o método de avaliação mede o que deveria medir, se o faz com rigor e evita que os dados sejam distorcidos); a *validade externa* do estudo (i.e., até que ponto os resultados podem ser generalizados ou transferidos a um outro contexto semelhante); e a *validade ecológica* do estudo (i.e., o quanto os materiais, métodos e ambiente de estudo se assemelham à situação real investigada).

Resposta: ( ) Conforme ( ) Não conforme ( ) Não se aplica

Critério de Fagan: se a resposta for Não conforme, o defeito é do tipo Omissão: falta no artefato um elemento importante que o item exige.

Fonte: Página: 69, Definição dos Conceitos (taxonomia de defeitos). Artigo: MARCOS KALINOWSKI, RODRIGO OLIVEIRA SPÍNOLA, Introdução à Inspeção de Software, Engenharia de Software Magazine, edição 01, p. 68 a 74.

Foto:

• **Omissão:** (1) Algum requisito importante relacionado à funcionalidade, ao desempenho, às restrições de projeto, ao atributo, ou à interface externa não foi incluído; (2) não está definida a resposta do software para todas as possíveis situações de entrada de dados; (3) faltam seções na especificação de requisitos; (4) faltam referências de figuras, tabelas, e diagramas; (5) falta definição de termos e unidades de medidas.

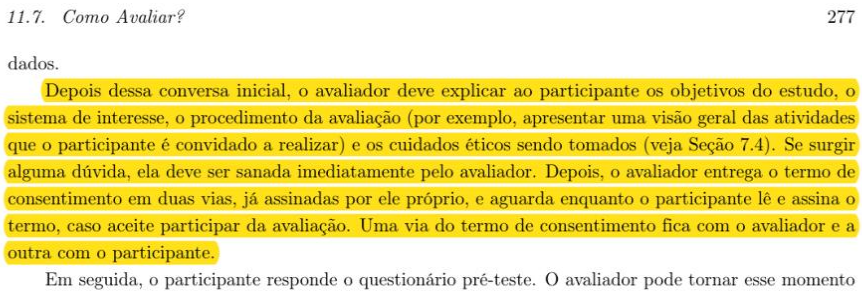
Severidade (Doolan, p. 178): ( ) Menor ( ) Maior ( ) Super-maior

Origem: Felipe Lopes Pedroza, Lista de Verificação: Entrega 6 (protótipo de papel, relato e planejamento da avaliação), item 9.

Item 97: Antes da sessão, o avaliador explica ao participante os objetivos, o sistema, o procedimento e os cuidados éticos e entrega o termo de consentimento em duas vias já assinadas por ele?

Fonte: Página: 277, Capítulo 11. Planejamento da Avaliação de IHC (Seção 11.7.3 Coleta de Dados). Livro: SIMONE DINIZ JUNQUEIRA BARBOSA, BRUNO SANTANA DA SILVA, MILENE SELBACH SILVEIRA, ISABELA GASPARINI, TICIANNE DARIN, GABRIEL DINIZ JUNQUEIRA BARBOSA, Interação Humano-Computador e Experiência do Usuário, 1ª ed., Rio de Janeiro, 2021.

Foto:



11.7. Como Avaliar? 277

dados.

Depois dessa conversa inicial, o avaliador deve explicar ao participante os objetivos do estudo, o sistema de interesse, o procedimento da avaliação (por exemplo, apresentar uma visão geral das atividades que o participante é convidado a realizar) e os cuidados éticos sendo tomados (veja Seção 7.4). Se surgir alguma dúvida, ela deve ser sanada imediatamente pelo avaliador. Depois, o avaliador entrega o termo de consentimento em duas vias, já assinadas por ele próprio, e aguarda enquanto o participante lê e assina o termo, caso aceite participar da avaliação. Uma via do termo de consentimento fica com o avaliador e a outra com o participante.

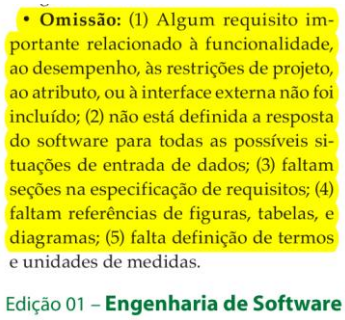
Em seguida, o participante responde o questionário pré-teste. O avaliador pode tornar esse momento

Resposta: ( ) Conforme ( ) Não conforme ( ) Não se aplica

Critério de Fagan: se a resposta for Não conforme, o defeito é do tipo Omissão: falta no artefato um elemento importante que o item exige.

Fonte: Página: 69, Definição dos Conceitos (taxonomia de defeitos). Artigo: MARCOS KALINOWSKI, RODRIGO OLIVEIRA SPÍNOLA, Introdução à Inspeção de Software, Engenharia de Software Magazine, edição 01, p. 68 a 74.

Foto:



• **Omissão:** (1) Algum requisito importante relacionado à funcionalidade, ao desempenho, às restrições de projeto, ao atributo, ou à interface externa não foi incluído; (2) não está definida a resposta do software para todas as possíveis situações de entrada de dados; (3) faltam seções na especificação de requisitos; (4) faltam referências de figuras, tabelas, e diagramas; (5) falta definição de termos e unidades de medidas.

Edição 01 – **Engenharia de Software Magazine** **69**

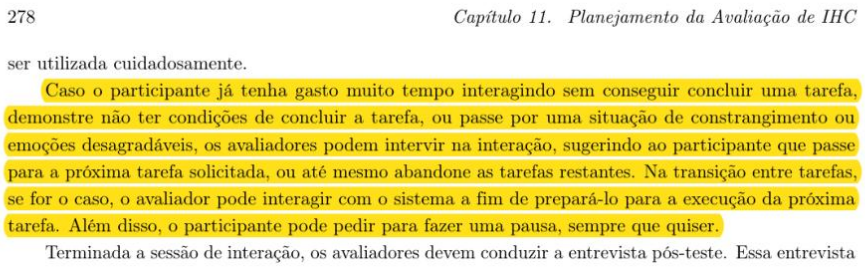
Severidade (Doolan, p. 178): ( ) Menor ( ) Maior ( ) Super-maior

Origem: Matheus do Vale Lameira, Lista de Verificação: Storyboard, item 42.

Item 98: O planejamento prevê que o participante pode pedir pausa sempre que quiser e que, se não conseguir concluir uma tarefa ou se sentir constrangido, pode passar para a próxima ou abandonar as restantes?

Fonte: Página: 278, Capítulo 11. Planejamento da Avaliação de IHC (Seção 11.7.3 Coleta de Dados). Livro: SIMONE DINIZ JUNQUEIRA BARBOSA, BRUNO SANTANA DA SILVA, MILENE SELBACH SILVEIRA, ISABELA GASPARINI, TICIANNE DARIN, GABRIEL DINIZ JUNQUEIRA BARBOSA, Interação Humano-Computador e Experiência do Usuário, 1ª ed., Rio de Janeiro, 2021.

Foto:



278 *Capítulo 11. Planejamento da Avaliação de IHC*

ser utilizada cuidadosamente.

Caso o participante já tenha gasto muito tempo interagindo sem conseguir concluir uma tarefa, demonstre não ter condições de concluir a tarefa, ou passe por uma situação de constrangimento ou emoções desagradáveis, os avaliadores podem intervir na interação, sugerindo ao participante que passe para a próxima tarefa solicitada, ou até mesmo abandone as tarefas restantes. Na transição entre tarefas, se for o caso, o avaliador pode interagir com o sistema a fim de prepará-lo para a execução da próxima tarefa. Além disso, o participante pode pedir para fazer uma pausa, sempre que quiser.

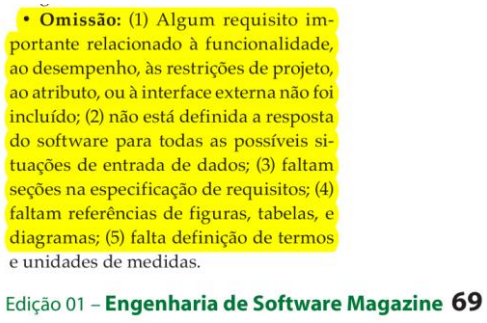
Terminada a sessão de interação, os avaliadores devem conduzir a entrevista pós-teste. Essa entrevista

Resposta: ( ) Conforme ( ) Não conforme ( ) Não se aplica

Critério de Fagan: se a resposta for Não conforme, o defeito é do tipo Omissão: falta no artefato um elemento importante que o item exige.

Fonte: Página: 69, Definição dos Conceitos (taxonomia de defeitos). Artigo: MARCOS KALINOWSKI, RODRIGO OLIVEIRA SPÍNOLA, Introdução à Inspeção de Software, Engenharia de Software Magazine, edição 01, p. 68 a 74.

Foto:



• **Omissão:** (1) Algum requisito importante relacionado à funcionalidade, ao desempenho, às restrições de projeto, ao atributo, ou à interface externa não foi incluído; (2) não está definida a resposta do software para todas as possíveis situações de entrada de dados; (3) faltam seções na especificação de requisitos; (4) faltam referências de figuras, tabelas, e diagramas; (5) falta definição de termos e unidades de medidas.

Edição 01 – **Engenharia de Software Magazine** **69**

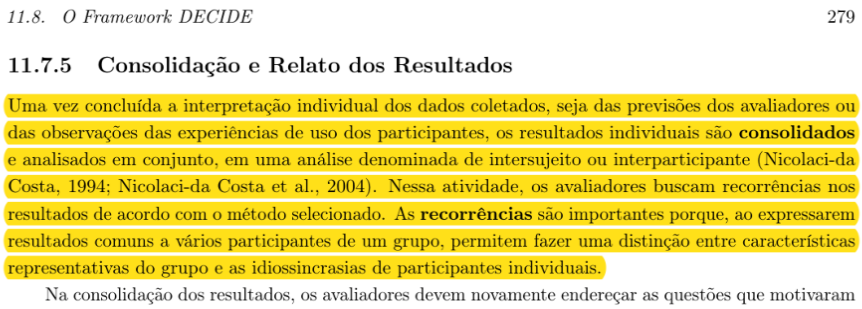
Severidade (Doolan, p. 178): ( ) Menor ( ) Maior ( ) Super-maior

Origem: Matheus do Vale Lameira, Lista de Verificação: Storyboard, item 43.

Item 99: Os dados de cada participante são interpretados individualmente e depois consolidados, buscando recorrências para distinguir o que é do grupo do que é de um participante só?

Fonte: Página: 279, Capítulo 11. Planejamento da Avaliação de IHC (Seção 11.7.5 Consolidação e Relato dos Resultados). Livro: SIMONE DINIZ JUNQUEIRA BARBOSA, BRUNO SANTANA DA SILVA, MILENE SELBACH SILVEIRA, ISABELA GASPARINI, TICIANNE DARIN, GABRIEL DINIZ JUNQUEIRA BARBOSA, Interação Humano-Computador e Experiência do Usuário, 1ª ed., Rio de Janeiro, 2021.

Foto:



11.8. O Framework DECIDE 279

11.7.5 Consolidação e Relato dos Resultados

Uma vez concluída a interpretação individual dos dados coletados, seja das previsões dos avaliadores ou das observações das experiências de uso dos participantes, os resultados individuais são **consolidados** e analisados em conjunto, em uma análise denominada de intersujeito ou interparticipante (Nicolaci-da Costa, 1994; Nicolaci-da Costa et al., 2004). Nessa atividade, os avaliadores buscam recorrências nos resultados de acordo com o método selecionado. As **recorrências** são importantes porque, ao expressarem resultados comuns a vários participantes de um grupo, permitem fazer uma distinção entre características representativas do grupo e as idiossincrasias de participantes individuais.

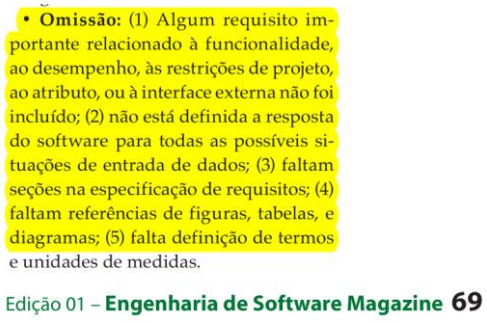
Na consolidação dos resultados, os avaliadores devem novamente endereçar as questões que motivaram

Resposta: ( ) Conforme ( ) Não conforme ( ) Não se aplica

Critério de Fagan: se a resposta for Não conforme, o defeito é do tipo Omissão: falta no artefato um elemento importante que o item exige.

Fonte: Página: 69, Definição dos Conceitos (taxonomia de defeitos). Artigo: MARCOS KALINOWSKI, RODRIGO OLIVEIRA SPÍNOLA, Introdução à Inspeção de Software, Engenharia de Software Magazine, edição 01, p. 68 a 74.

Foto:



• **Omissão:** (1) Algum requisito importante relacionado à funcionalidade, ao desempenho, às restrições de projeto, ao atributo, ou à interface externa não foi incluído; (2) não está definida a resposta do software para todas as possíveis situações de entrada de dados; (3) faltam seções na especificação de requisitos; (4) faltam referências de figuras, tabelas, e diagramas; (5) falta definição de termos e unidades de medidas.

Edição 01 – **Engenharia de Software Magazine** **69**

Severidade (Doolan, p. 178): ( ) Menor ( ) Maior ( ) Super-maior

Origem: Matheus do Vale Lameira, Lista de Verificação: Storyboard, item 45.

4.9 Relato dos resultados da avaliação

Item 100: O relato da avaliação do protótipo de papel comunica os objetivos, uma breve descrição do método, o número e o perfil dos avaliadores e dos participantes e as tarefas executadas?

Fonte: Página: 361, Capítulo 10. Métodos de Avaliação de IHC. Livro: SIMONE DINIZ JUNQUEIRA BARBOSA, BRUNO SANTANA DA SILVA, Interação Humano-Computador, 1ª ed., Editora Campus/Elsevier, 2010.

Foto:

dores sugerem propostas de correção desses problemas ou de caminhos que podem ser explorados para melhorar a interface.

No *relato dos resultados*, os avaliadores devem comunicar aos interessados:

- os objetivos da avaliação;
- uma breve descrição do método de prototipação em papel;
- o número e o perfil de avaliadores e dos participantes;
- as tarefas executadas pelos participantes;
- uma lista de problemas de usabilidade corrigidos durante os ciclos de avaliação e reprojeto, indicando:

Resposta: ( ) Conforme ( ) Não conforme ( ) Não se aplica

Critério de Fagan: se a resposta for Não conforme, o defeito é do tipo Omissão: falta no artefato um elemento importante que o item exige.

Fonte: Página: 69, Definição dos Conceitos (taxonomia de defeitos). Artigo: MARCOS KALINOWSKI, RODRIGO OLIVEIRA SPÍNOLA, Introdução à Inspeção de Software, Engenharia de Software Magazine, edição 01, p. 68 a 74.

Foto:

• **Omissão:** (1) Algum requisito importante relacionado à funcionalidade, ao desempenho, às restrições de projeto, ao atributo, ou à interface externa não foi incluído; (2) não está definida a resposta do software para todas as possíveis situações de entrada de dados; (3) faltam seções na especificação de requisitos; (4) faltam referências de figuras, tabelas, e diagramas; (5) falta definição de termos e unidades de medidas.

Severidade (Doolan, p. 178): ( ) Menor ( ) Maior ( ) Super-maior

Origem: Felipe Lopes Pedroza, Lista de Verificação: Entrega 6 (protótipo de papel, relato e planejamento da avaliação), item 4.

Item 101: O relato traz a lista de problemas de usabilidade corrigidos durante os ciclos de avaliação e reprojeto, com o local, os fatores de usabilidade prejudicados, a descrição, a correção feita e se o problema voltou a ocorrer?

Fonte: Página: 361, Capítulo 10. Métodos de Avaliação de IHC. Livro: SIMONE DINIZ JUNQUEIRA BARBOSA, BRUNO SANTANA DA SILVA, Interação Humano-Computador, 1ª ed., Editora Campus/Elsevier, 2010.

Foto:

- o número e o perfil de avaliadores e dos participantes;
- as tarefas executadas pelos participantes;
- uma lista de problemas de usabilidade corrigidos durante os ciclos de avaliação e reprojeto, indicando:
  - local onde ocorreu;
  - fatores de usabilidade prejudicados;
  - descrição e justificativa do problema;
  - correção realizada no protótipo em papel;
  - indicação se o problema voltou a ocorrer depois da correção;

Resposta: ( ) Conforme ( ) Não conforme ( ) Não se aplica

Critério de Fagan: se a resposta for Não conforme, o defeito é do tipo Omissão: falta no artefato um elemento importante que o item exige.

Fonte: Página: 69, Definição dos Conceitos (taxonomia de defeitos). Artigo: MARCOS KALINOWSKI, RODRIGO OLIVEIRA SPÍNOLA, Introdução à Inspeção de Software, Engenharia de Software Magazine, edição 01, p. 68 a 74.

Foto:

• **Omissão:** (1) Algum requisito importante relacionado à funcionalidade, ao desempenho, às restrições de projeto, ao atributo, ou à interface externa não foi incluído; (2) não está definida a resposta do software para todas as possíveis situações de entrada de dados; (3) faltam seções na especificação de requisitos; (4) faltam referências de figuras, tabelas, e diagramas; (5) falta definição de termos e unidades de medidas.

Severidade (Doolan, p. 178): ( ) Menor ( ) Maior ( ) Super-maior

Origem: Felipe Lopes Pedroza, Lista de Verificação: Entrega 6 (protótipo de papel, relato e planejamento da avaliação), item 5.

Item 102: O relato traz a lista dos problemas ainda não corrigidos, com o local, os fatores prejudicados, a descrição, a prioridade e sugestões de correção, além das partes que podem ser mais bem elaboradas?

Fonte: Página: 362, Capítulo 10. Métodos de Avaliação de IHC. Livro: SIMONE DINIZ JUNQUEIRA BARBOSA, BRUNO SANTANA DA SILVA, Interação Humano-Computador, 1ª ed., Editora Campus/Elsevier, 2010.

Foto:

- uma lista dos problemas de usabilidade ainda não corrigidos, indicando:
  - local onde ocorreu;
  - fatores de usabilidade prejudicados;
  - descrição e justificativa do problema;
  - prioridade para correção;
  - sugestões de correção;
- indicações de partes do sistema que podem ser mais bem elaboradas.

10.3 Resumo Comparativo dos Métodos de Avaliação

Resposta: ( ) Conforme ( ) Não conforme ( ) Não se aplica

Critério de Fagan: se a resposta for Não conforme, o defeito é do tipo Omissão: falta no artefato um elemento importante que o item exige.

Fonte: Página: 69, Definição dos Conceitos (taxonomia de defeitos). Artigo: MARCOS KALINOWSKI, RODRIGO OLIVEIRA SPÍNOLA, Introdução à Inspeção de Software, Engenharia de Software Magazine, edição 01, p. 68 a 74.

Foto:

• **Omissão:** (1) Algum requisito importante relacionado à funcionalidade, ao desempenho, às restrições de projeto, ao atributo, ou à interface externa não foi incluído; (2) não está definida a resposta do software para todas as possíveis situações de entrada de dados; (3) faltam seções na especificação de requisitos; (4) faltam referências de figuras, tabelas, e diagramas; (5) falta definição de termos e unidades de medidas.

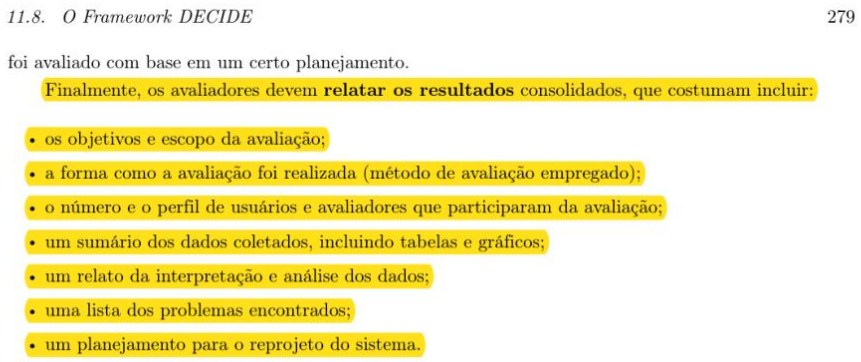
Severidade (Doolan, p. 178): ( ) Menor ( ) Maior ( ) Super-maior

Origem: Felipe Lopes Pedroza, Lista de Verificação: Entrega 6 (protótipo de papel, relato e planejamento da avaliação), item 6.

Item 103: O planejamento do relato prevê os objetivos e o escopo, o método empregado, o número e o perfil de usuários e avaliadores, o sumário dos dados, a interpretação e a análise, a lista de problemas e o planejamento do reprojeto?

Fonte: Página: 279, Capítulo 11. Planejamento da Avaliação de IHC (Seção 11.7.5 Consolidação e Relato dos Resultados). Livro: SIMONE DINIZ JUNQUEIRA BARBOSA, BRUNO SANTANA DA SILVA, MILENE SELBACH SILVEIRA, ISABELA GASPARINI, TICIANNE DARIN, GABRIEL DINIZ JUNQUEIRA BARBOSA, Interação Humano-Computador e Experiência do Usuário, 1ª ed., Rio de Janeiro, 2021.

Foto:



11.8. O Framework DECIDE 279

foi avaliado com base em um certo planejamento.

Finalmente, os avaliadores devem **relatar os resultados** consolidados, que costumam incluir:

- os objetivos e escopo da avaliação;
- a forma como a avaliação foi realizada (método de avaliação empregado);
- o número e o perfil de usuários e avaliadores que participaram da avaliação;
- um sumário dos dados coletados, incluindo tabelas e gráficos;
- um relato da interpretação e análise dos dados;
- uma lista dos problemas encontrados;
- um planejamento para o reprojeto do sistema.

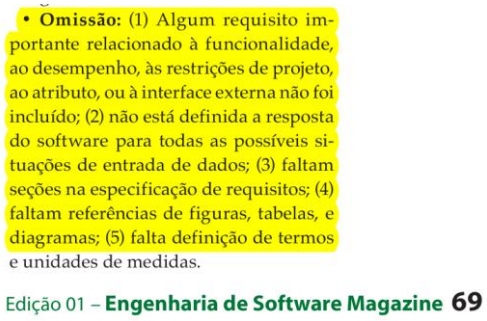
11.8 O Framework DECIDE

Resposta: ( ) Conforme ( ) Não conforme ( ) Não se aplica

Critério de Fagan: se a resposta for Não conforme, o defeito é do tipo Omissão: falta no artefato um elemento importante que o item exige.

Fonte: Página: 69, Definição dos Conceitos (taxonomia de defeitos). Artigo: MARCOS KALINOWSKI, RODRIGO OLIVEIRA SPÍNOLA, Introdução à Inspeção de Software, Engenharia de Software Magazine, edição 01, p. 68 a 74.

Foto:



- **Omissão:** (1) Algum requisito importante relacionado à funcionalidade, ao desempenho, às restrições de projeto, ao atributo, ou à interface externa não foi incluído; (2) não está definida a resposta do software para todas as possíveis situações de entrada de dados; (3) faltam seções na especificação de requisitos; (4) faltam referências de figuras, tabelas, e diagramas; (5) falta definição de termos e unidades de medidas.

Edição 01 – **Engenharia de Software Magazine** **69**

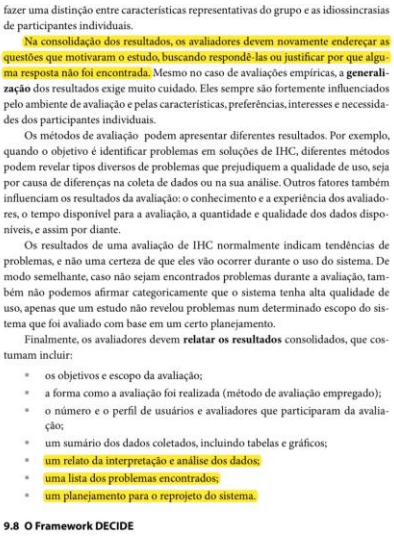
Severidade (Doolan, p. 178): ( ) Menor ( ) Maior ( ) Super-maior

Origem: Matheus do Vale Lameira, Lista de Verificação: Storyboard, item 46.

Item 104: O planejamento do relato prevê responder às questões que motivaram o estudo, ou justificar por que alguma resposta não foi encontrada, além da interpretação e da análise, da lista de problemas e do planejamento do reprojeto?

Fonte: Página: 312, Capítulo 9. Planejamento da Avaliação de IHC. Livro: SIMONE DINIZ JUNQUEIRA BARBOSA, BRUNO SANTANA DA SILVA, Interação Humano-Computador, 1ª ed., Editora Campus/Elsevier, 2010.

Foto:



fazer uma distinção entre características representativas do grupo e as idiossincrasias de participantes individuais.

Na consolidação dos resultados, os avaliadores devem novamente endereçar as questões que motivaram o estudo, buscando respondê-las ou justificar por que alguma resposta não foi encontrada. Mesmo no caso de avaliações empíricas, a **generalização** dos resultados exige muito cuidado. Eles sempre são fortemente influenciados pelo ambiente de avaliação e pelas características, preferências, interesses e necessidades dos participantes individuais.

Os métodos de avaliação podem apresentar diferentes resultados. Por exemplo, quando o objetivo é identificar problemas em soluções de IHC, diferentes métodos podem revelar tipos diversos de problemas que prejudiquem a qualidade de uso, seja por causa de diferenças na coleta de dados ou na sua análise. Outros fatores também influenciam os resultados da avaliação: o conhecimento e a experiência dos avaliadores, o tempo disponível para a avaliação, a quantidade e qualidade dos dados disponíveis, e assim por diante.

Os resultados de uma avaliação de IHC normalmente indicam tendências de problemas, e não uma certeza de que eles vão ocorrer durante o uso do sistema. De modo semelhante, caso não sejam encontrados problemas durante a avaliação, também não podemos afirmar categoricamente que o sistema tenha alta qualidade de uso, apenas que um estudo não revelou problemas num determinado escopo do sistema que foi avaliado com base em um certo planejamento.

Finalmente, os avaliadores devem **relatar os resultados** consolidados, que costumam incluir:

- os objetivos e escopo da avaliação;
- a forma como a avaliação foi realizada (método de avaliação empregado);
- o número e o perfil de usuários e avaliadores que participaram da avaliação;
- um sumário dos dados coletados, incluindo tabelas e gráficos;
- um relato da interpretação e análise dos dados;
- uma lista dos problemas encontrados;
- um planejamento para o reprojeto do sistema.

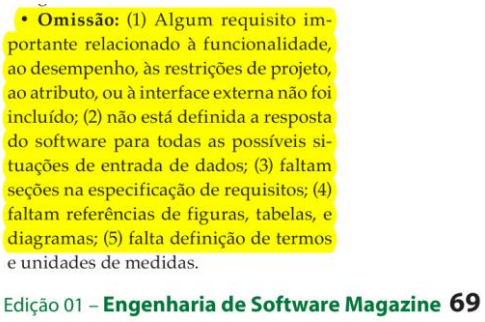
9.8 O Framework DECIDE

Resposta: ( ) Conforme ( ) Não conforme ( ) Não se aplica

Critério de Fagan: se a resposta for Não conforme, o defeito é do tipo Omissão: falta no artefato um elemento importante que o item exige.

Fonte: Página: 69, Definição dos Conceitos (taxonomia de defeitos). Artigo: MARCOS KALINOWSKI, RODRIGO OLIVEIRA SPÍNOLA, Introdução à Inspeção de Software, Engenharia de Software Magazine, edição 01, p. 68 a 74.

Foto:



• **Omissão:** (1) Algum requisito importante relacionado à funcionalidade, ao desempenho, às restrições de projeto, ao atributo, ou à interface externa não foi incluído; (2) não está definida a resposta do software para todas as possíveis situações de entrada de dados; (3) faltam seções na especificação de requisitos; (4) faltam referências de figuras, tabelas, e diagramas; (5) falta definição de termos e unidades de medidas.

Edição 01 – **Engenharia de Software Magazine** **69**

Severidade (Doolan, p. 178): ( ) Menor ( ) Maior ( ) Super-maior

Origem: Felipe Lopes Pedroza, Lista de Verificação: Entrega 6 (protótipo de papel, relato e planejamento da avaliação), item 11.

Item 105: No teste de usabilidade, o relato prevê tabelas e gráficos que sumarizam as medições e a lista de problemas, com local, descrição e justificativa, fatores de usabilidade prejudicados e sugestões de solução?

Fonte: Página: 343, Capítulo 10. Métodos de Avaliação de IHC. Livro: SIMONE DINIZ JUNQUEIRA BARBOSA, BRUNO SANTANA DA SILVA, Interação Humano-Computador, 1ª ed., Editora Campus/Elsevier, 2010.

Foto:

de problemas com base nas suas interpretações do que ocorreu e fornecer sugestões de melhorias na interface e interação.

O *relato dos resultados* do teste de usabilidade deve descrever (veja Seção 9.7.5):

- os objetivos e escopo da avaliação;
- uma breve descrição do método de teste de usabilidade;
- o número e o perfil dos avaliadores e dos participantes;
- as tarefas executadas pelos participantes;
- tabelas e gráficos que sumarizam as medições realizadas;
- uma lista de problemas encontrados, indicando, para cada problema:
  - local onde ocorreu;
  - descrição e justificativa;
  - discussão, indicando os fatores de usabilidade prejudicados;
  - sugestões de solução.

Resposta: ( ) Conforme ( ) Não conforme ( ) Não se aplica

Critério de Fagan: se a resposta for Não conforme, o defeito é do tipo Omissão: falta no artefato um elemento importante que o item exige.

Fonte: Página: 69, Definição dos Conceitos (taxonomia de defeitos). Artigo: MARCOS KALINOWSKI, RODRIGO OLIVEIRA SPÍNOLA, Introdução à Inspeção de Software, Engenharia de Software Magazine, edição 01, p. 68 a 74.

Foto:

• **Omissão:** (1) Algum requisito importante relacionado à funcionalidade, ao desempenho, às restrições de projeto, ao atributo, ou à interface externa não foi incluído; (2) não está definida a resposta do software para todas as possíveis situações de entrada de dados; (3) faltam seções na especificação de requisitos; (4) faltam referências de figuras, tabelas, e diagramas; (5) falta definição de termos e unidades de medidas.

Severidade (Doolan, p. 178): ( ) Menor ( ) Maior ( ) Super-maior

Origem: Felipe Lopes Pedroza, Lista de Verificação: Entrega 6 (protótipo de papel, relato e planejamento da avaliação), item 12.

5. Ficha de registro da inspeção

Campo	Registro
Projeto e artefato inspecionado	
Autor do artefato	
Moderador	
Inspetores	
Seções da lista usadas	
Data da apresentação e da reunião	
Tempo de preparação de cada inspetor	
Duração da reunião (até 2 horas)	

Tabela 7: dados da inspeção.

Nº	Item	Local no artefato	Descrição da discrepância	Tipo	Sev.	Defeito ou falso positivo	Corrigido?
1							
2							
3							
4							
5							
6							
7							
8							
9							
10							
11							
12							

Tabela 8: registro dos defeitos (preparação, reunião e retrabalho).

Decisão do moderador na continuação: ( ) artefato aceito ( ) aceito após o retrabalho ( ) nova inspeção

Totais: defeitos menores ____ maiores ____ super-maiores ____ falsos positivos ____

6. Declaração sobre o uso de IA Generativa

Usamos a ferramenta de IA Generativa Claude (Anthropic) como apoio para reunir e padronizar os itens das listas de origem e para montar este documento. Os itens, as fontes e as fotos vêm das listas elaboradas por cada integrante. O conteúdo e a revisão final são de responsabilidade do grupo.

7. Referências

BARBOSA, Simone Diniz Junqueira; SILVA, Bruno Santana da. Interação Humano-Computador. 1ª ed. Rio de Janeiro: Elsevier/Campus, 2010.

BARBOSA, Simone Diniz Junqueira; SILVA, Bruno Santana da; SILVEIRA, Milene Selbach; GASPARINI, Isabela; DARIN, Ticianne; BARBOSA, Gabriel Diniz Junqueira. Interação Humano-Computador e Experiência do Usuário. 1ª ed. Rio de Janeiro, 2021.

DOOLAN, E. P. Experience with Fagan's Inspection Method. Software: Practice and Experience, vol. 22, n. 2, p. 173 a 182, fev. 1992.

FAGAN, M. E. Design and Code Inspection to Reduce Errors in Program Development. IBM Systems Journal, vol. 15, n. 3, p. 182 a 211, 1976 (citado por Kalinowski e Spínola).

KALINOWSKI, Marcos; SPÍNOLA, Rodrigo Oliveira. Introdução à Inspeção de Software. Engenharia de Software Magazine, ed. 01, p. 68 a 74.

ROGERS, Yvonne; SHARP, Helen; PREECE, Jennifer. Design de Interação: além da interação humano-computador. 3ª ed. Porto Alegre: Bookman, 2013.

VIANNA, Maurício; VIANNA, Ysmar; ADLER, Isabel K.; LUCENA, Brenda; RUSSO, Beatriz. Design Thinking: Inovação em Negócios. Rio de Janeiro: MJV Press, 2012.

PREECE, Jennifer; ROGERS, Yvonne; SHARP, Helen. Design de Interação: além da interação homem-computador. Porto Alegre: Bookman, 2005.

Listas de verificação de origem: Felipe Lopes Pedroza (CTT; Storyboard; Entrega 6); Matheus do Vale Lameira (Perfil do Usuário; Storyboard); Victor Andreozzi De La Rocque Couto (Planejamento da Avaliação; Personas; Cenários; HTA; Protótipo de Papel). Universidade de Brasília, 2026.