

Disciplina: Requisitos de Software
Professor: André Barros de Sales
Matrícula: 221031256
Nome: Angélica da Costa Campos

Introdução

Esta lista de verificação foi elaborada para inspecionar artefato relacionado a Comparação em pares (Pairwise Comparison).

Lista de Verificação

1. Foi avaliado os requisitos comparando-os em pares?

E Pairwise comparison and rank ordering

People sometimes try to assign a unique priority sequence number to each requirement. Rank ordering a list of requirements involves making pairwise comparisons between all of them so you can judge which member of each pair has higher priority. Figure 14-1 in Chapter 14, "Beyond functionality," illustrated the use of a spreadsheet to perform just such a pairwise comparison of quality attributes; the same strategy could be applied to a set of features, user stories, or any other set of requirements of the same type. Performing such comparisons becomes unwieldy for more than a couple of dozen requirements. It could work at the granularity level of features, but not for all the functional requirements for a system as a whole.

In reality, rank ordering all of the requirements by priority is overkill. You won't be implementing all of these in individual releases; instead, you'll group them together in batches by release or development timebox. Grouping requirements into features, or into small sets of requirements that have similar priority or that otherwise must be implemented together, is sufficient.

WIEGERS, Karl; BEATTY, Joy. Software Requirements. 3. ed. Redmond: Microsoft Press, 2013. P.318.

2. Após a comparação em pares foi determinado qual tem maior prioridade de dentro da categoria?

E Pairwise comparison and rank ordering

People sometimes try to assign a unique priority sequence number to each requirement. Rank ordering a list of requirements involves making pairwise comparisons between all of them so you can judge which member of each pair has higher priority. Figure 14-1 in Chapter 14, "Beyond functionality," illustrated the use of a spreadsheet to perform just such a pairwise comparison of quality attributes; the same strategy could be applied to a set of features, user stories, or any other set of requirements of the same type. Performing such comparisons becomes unwieldy for more than a couple of dozen requirements. It could work at the granularity level of features, but not for all the functional requirements for a system as a whole.

In reality, rank ordering all of the requirements by priority is overkill. You won't be implementing all of these in individual releases; instead, you'll group them together in batches by release or development timebox. Grouping requirements into features, or into small sets of requirements that have similar priority or that otherwise must be implemented together, is sufficient.

WIEGERS, Karl; BEATTY, Joy. Software Requirements. 3. ed. Redmond: Microsoft Press, 2013. P.318.

3. Após a comparação em pares dentro de cada categoria foi determinado qual tem maior prioridade entre as categorias?

E Pairwise comparison and rank ordering

People sometimes try to assign a unique priority sequence number to each requirement. Rank ordering a list of requirements involves making pairwise comparisons between all of them so you can judge which member of each pair has higher priority. Figure 14-1 in Chapter 14, "Beyond functionality," illustrated the use of a spreadsheet to perform just such a pairwise comparison of quality attributes; the same strategy could be applied to a set of features, user stories, or any other set of requirements of the same type. Performing such comparisons becomes unwieldy for more than a couple of dozen requirements. It could work at the granularity level of features, but not for all the functional requirements for a system as a whole.

In reality, rank ordering all of the requirements by priority is overkill. You won't be implementing all of these in individual releases; instead, you'll group them together in batches by release or development timebox. Grouping requirements into features, or into small sets of requirements that have similar priority or that otherwise must be implemented together, is sufficient.

WIEGERS, Karl; BEATTY, Joy. Software Requirements. 3. ed. Redmond: Microsoft Press, 2013. P.318.

4. Foi usado uma planilha ou ferramenta semelhante para organizar as comparações?

E Pairwise comparison and rank ordering

People sometimes try to assign a unique priority sequence number to each requirement. Rank ordering a list of requirements involves making pairwise comparisons between all of them so you can judge which member of each pair has higher priority. Figure 14-1 in Chapter 14, "Beyond functionality," illustrated the use of a spreadsheet to perform just such a pairwise comparison of quality attributes; the same strategy could be applied to a set of features, user stories, or any other set of requirements of the same type. Performing such comparisons becomes unwieldy for more than a couple of dozen requirements. It could work at the granularity level of features, but not for all the functional requirements for a system as a whole.

In reality, rank ordering all of the requirements by priority is overkill. You won't be implementing all of these in individual releases; instead, you'll group them together in batches by release or development timebox. Grouping requirements into features, or into small sets of requirements that have similar priority or that otherwise must be implemented together, is sufficient.

WIEGERS, Karl; BEATTY, Joy. Software Requirements. 3. ed. Redmond: Microsoft Press, 2013. P.318.

5. A técnica foi aplicada em listas com até 12 requisitos?

E Pairwise comparison and rank ordering

People sometimes try to assign a unique priority sequence number to each requirement. Rank ordering a list of requirements involves making pairwise comparisons between all of them so you can judge which member of each pair has higher priority. Figure 14-1 in Chapter 14, "Beyond functionality," illustrated the use of a spreadsheet to perform just such a pairwise comparison of quality attributes; the same strategy could be applied to a set of features, user stories, or any other set of requirements of the same type. Performing such comparisons becomes unwieldy for more than a couple of dozen requirements. It could work at the granularity level of features, but not for all the functional requirements for a system as a whole.

In reality, rank ordering all of the requirements by priority is overkill. You won't be implementing all of these in individual releases; instead, you'll group them together in batches by release or development timebox. Grouping requirements into features, or into small sets of requirements that have similar priority or that otherwise must be implemented together, is sufficient.

WIEGERS, Karl; BEATTY, Joy. Software Requirements. 3. ed. Redmond: Microsoft Press, 2013. P.318.

6. Os requisitos foram agrupados em lotes, como categorias ou funcionalidades?

E

Pairwise comparison and rank ordering

People sometimes try to assign a unique priority sequence number to each requirement. Rank ordering a list of requirements involves making pairwise comparisons between all of them so you can judge which member of each pair has higher priority. Figure 14-1 in Chapter 14, "Beyond functionality," illustrated the use of a spreadsheet to perform just such a pairwise comparison of quality attributes; the same strategy could be applied to a set of features, user stories, or any other set of requirements of the same type. Performing such comparisons becomes unwieldy for more than a couple of dozen requirements. It could work at the granularity level of features, but not for all the functional requirements for a system as a whole.

In reality, rank ordering all of the requirements by priority is overkill. You won't be implementing all of these in individual releases; instead, you'll group them together in batches by release or development timebox. Grouping requirements into features, or into small sets of requirements that have similar priority or that otherwise must be implemented together, is sufficient.

WIEGERS, Karl; BEATTY, Joy. Software Requirements. 3. ed. Redmond: Microsoft Press, 2013. P.318.

7. Os requisitos agrupados tem prioridade semelhante?

E

Pairwise comparison and rank ordering

People sometimes try to assign a unique priority sequence number to each requirement. Rank ordering a list of requirements involves making pairwise comparisons between all of them so you can judge which member of each pair has higher priority. Figure 14-1 in Chapter 14, "Beyond functionality," illustrated the use of a spreadsheet to perform just such a pairwise comparison of quality attributes; the same strategy could be applied to a set of features, user stories, or any other set of requirements of the same type. Performing such comparisons becomes unwieldy for more than a couple of dozen requirements. It could work at the granularity level of features, but not for all the functional requirements for a system as a whole.

In reality, rank ordering all of the requirements by priority is overkill. You won't be implementing all of these in individual releases; instead, you'll group them together in batches by release or development timebox. Grouping requirements into features, or into small sets of requirements that have similar priority or that otherwise must be implemented together, is sufficient.

WIEGERS, Karl; BEATTY, Joy. Software Requirements. 3. ed. Redmond: Microsoft Press, 2013. P.318.

8. A técnica foi aplicada de forma exagerada avaliando priorizá-los todos em sequência única?

E

Pairwise comparison and rank ordering

People sometimes try to assign a unique priority sequence number to each requirement. Rank ordering a list of requirements involves making pairwise comparisons between all of them so you can judge which member of each pair has higher priority. Figure 14-1 in Chapter 14, "Beyond functionality," illustrated the use of a spreadsheet to perform just such a pairwise comparison of quality attributes; the same strategy could be applied to a set of features, user stories, or any other set of requirements of the same type. Performing such comparisons becomes unwieldy for more than a couple of dozen requirements. It could work at the granularity level of features, but not for all the functional requirements for a system as a whole.

In reality, rank ordering all of the requirements by priority is overkill. You won't be implementing all of these in individual releases; instead, you'll group them together in batches by release or development timebox. Grouping requirements into features, or into small sets of requirements that have similar priority or that otherwise must be implemented together, is sufficient.

WIEGERS, Karl; BEATTY, Joy. Software Requirements. 3. ed. Redmond: Microsoft Press, 2013. P.318.

9. Foi verificado se alguns requisitos precisam ser implementados juntos obrigatoriamente?

E Pairwise comparison and rank ordering

People sometimes try to assign a unique priority sequence number to each requirement. Rank ordering a list of requirements involves making pairwise comparisons between all of them so you can judge which member of each pair has higher priority. Figure 14-1 in Chapter 14, "Beyond functionality," illustrated the use of a spreadsheet to perform just such a pairwise comparison of quality attributes; the same strategy could be applied to a set of features, user stories, or any other set of requirements of the same type. Performing such comparisons becomes unwieldy for more than a couple of dozen requirements. It could work at the granularity level of features, but not for all the functional requirements for a system as a whole.

In reality, rank ordering all of the requirements by priority is overkill. You won't be implementing all of these in individual releases; instead, you'll group them together in batches by release or development timebox. Grouping requirements into features, or into small sets of requirements that have similar priority or that otherwise must be implemented together, is sufficient.

WIEGERS, Karl; BEATTY, Joy. Software Requirements. 3. ed. Redmond: Microsoft Press, 2013. P.318.

10. Os resultados da comparação em pares e da priorização final foram registrados?

E Pairwise comparison and rank ordering

People sometimes try to assign a unique priority sequence number to each requirement. Rank ordering a list of requirements involves making pairwise comparisons between all of them so you can judge which member of each pair has higher priority. Figure 14-1 in Chapter 14, "Beyond functionality," illustrated the use of a spreadsheet to perform just such a pairwise comparison of quality attributes; the same strategy could be applied to a set of features, user stories, or any other set of requirements of the same type. Performing such comparisons becomes unwieldy for more than a couple of dozen requirements. It could work at the granularity level of features, but not for all the functional requirements for a system as a whole.

In reality, rank ordering all of the requirements by priority is overkill. You won't be implementing all of these in individual releases; instead, you'll group them together in batches by release or development timebox. Grouping requirements into features, or into small sets of requirements that have similar priority or that otherwise must be implemented together, is sufficient.

WIEGERS, Karl; BEATTY, Joy. Software Requirements. 3. ed. Redmond: Microsoft Press, 2013. P.318.

