Fundação Bradesco

Engenharia de Requisitos

Módulo 02 - Entendendo os Requisitos

Parceria:

TATA

2.1 Requisitos Funcionais e Não Funcionais

Os requisitos podem ser classificados como:

a) Funcionais

b) Não funcionais

2.1 Requisitos Funcionais e Não Funcionais

Os requisitos funcionais especificam o que o sistema deve fazer. Por exemplo:

- Calcular os juros compostos à taxa de 14% ao ano, de um depósito fixo, por um período de três anos.
- Calcular o imposto à taxa de 30% em uma renda anual igual a e acima de R$ 2.000.000, mas não superior a R$ 3.000.000.
- Inverter uma matriz quadrada de números reais (tamanho máximo 100 x 100).

2.1 Requisitos Funcionais e Não Funcionais

Requisitos não funcionais especificam os atributos de qualidade gerais que o sistema deve satisfazer. Por exemplo:

- Portabilidade
- Confiabilidade
- Desempenho
- Testabilidade
- Modificabilidade
- Segurança
- Apresentação
- Reusabilidade
- Entendibilidade
- Aceitabilidade
- Interoperabilidade

2.1 Requisitos Funcionais e Não Funcionais

1.2162899989 = 1.216289999
1.06543226672 = 1.0654322667

Alguns exemplos de requisitos não funcionais são:

- O número significativo de dígitos para o qual a precisão deve ser mantida em todos os cálculos numéricos é 10.
- O tempo de resposta do sistema deve ser sempre inferior a 5 segundos.

2.1 Requisitos Funcionais e Não Funcionais

Figura: #incluir <stdio.h>
/* Ler uma linha de fp, * /
/* copiando-a para estrutura de linha (mas não mais que máx char). * /
/* Não colocar terminando em \n na estrutura de linha. */
/* Retorna comprimento linha, ou 0 para linha vazia ou EOF para fim de arquivo. */
/* int fgetline (FILE *fp, char line [], int max)
{
int nch = 0;
int c;
max = max -1 /* deixar espaço para '\0' */

enquanto ((c = getc(fp)) != EOF)
{
se(c == '\n');
divida;

se(nch < max)
{
linha[nch] = c;
nch = nch + 1;
}
}

- O software deve ser desenvolvido usando linguagem C em um sistema baseado no UNIX.
- Um livro pode ser deletado do Sistema de Gerenciamento de Biblioteca apenas pelo Administrador de Banco de Dados.

2.1 Requisitos Funcionais e Não Funcionais

$$\begin{vmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 2 & 6 & 4 & 8 \\ 3 & 1 & 1 & 2 \end{vmatrix}$$

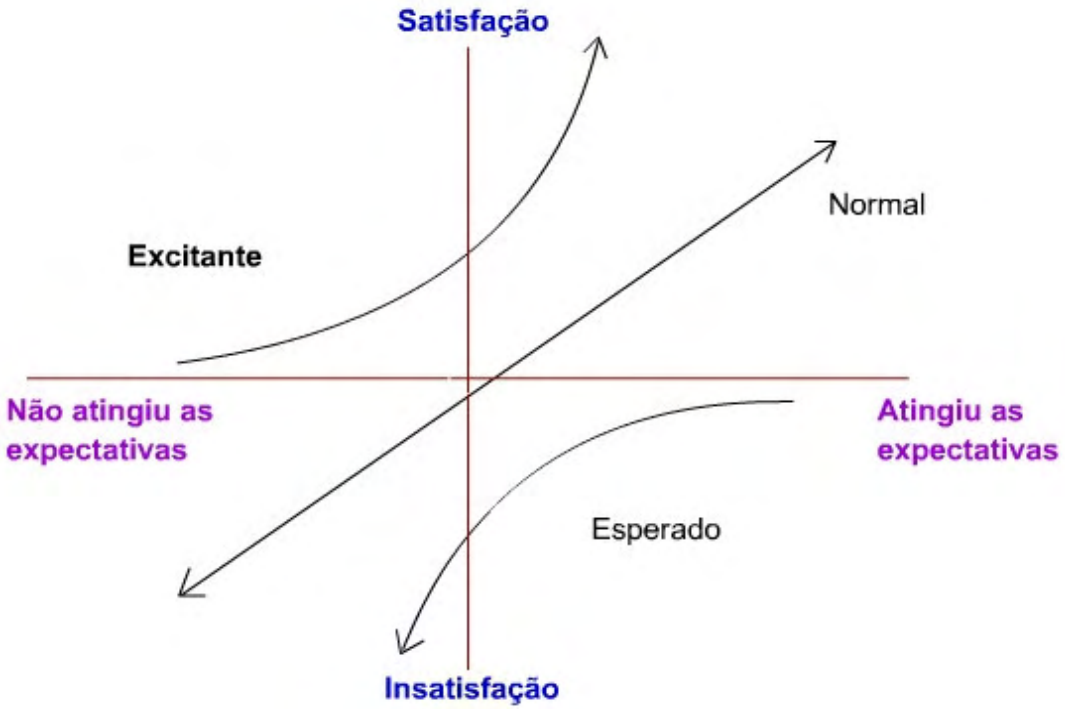
- A rotina de diagonalização da matriz deve zerar todos os elementos diagonais, que sejam iguais a ou inferiores a 10^{-3}.
- Executivos experientes devem ser capazes de usar todas as funções do sistema após um treinamento total de duas horas. Após esse treinamento, o número médio de erros cometidos por eles não deve exceder a dois por dia.

2.2 Outras classificações

Os requisitos também podem ser classificados nas seguintes categorias:

- Nível de satisfação
- Prioridade
- Estabilidade
- Categorias de usuário

2.2 Outras classificações



Nível de satisfação:

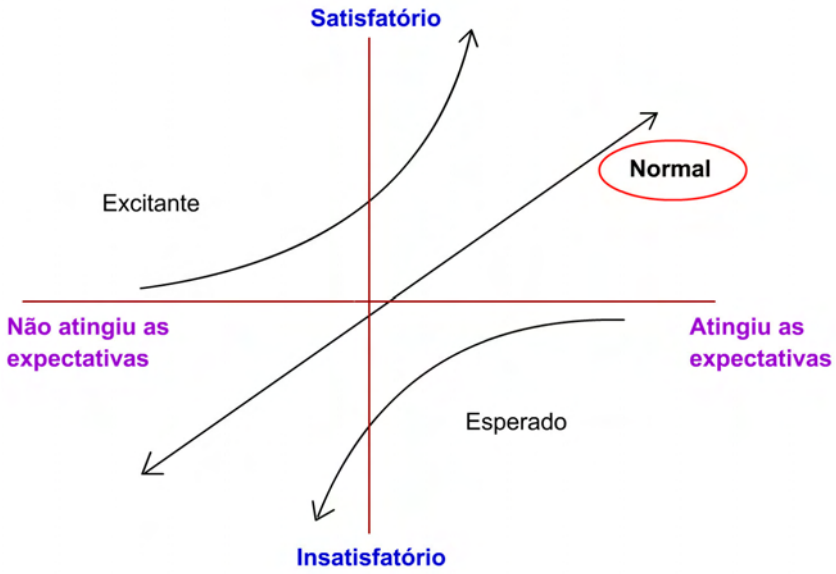
Existem três tipos:

a) Normal

b) Esperado

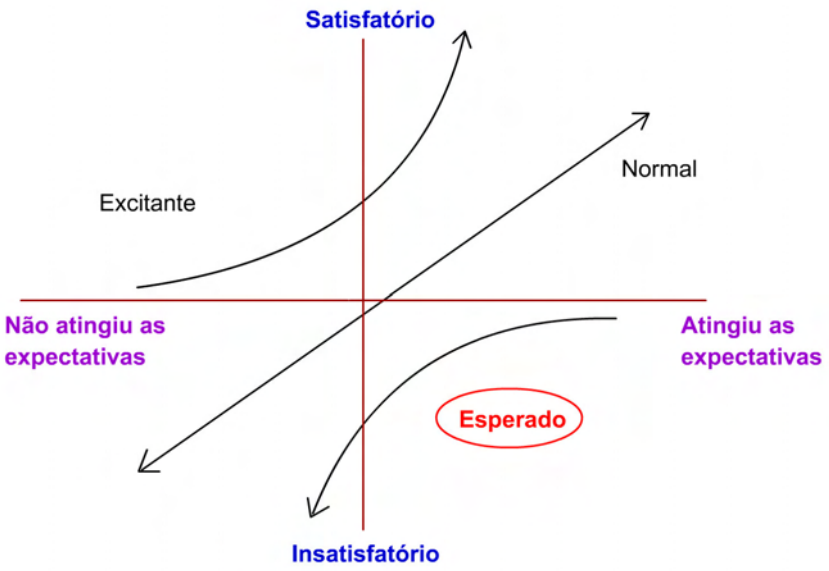
c) Além da expectativa

2.2 Outras classificações



Os requisitos normais são declarações específicas das necessidades do usuário. O nível de satisfação do usuário é diretamente proporcional ao grau em eles são satisfeitos pelo sistema.

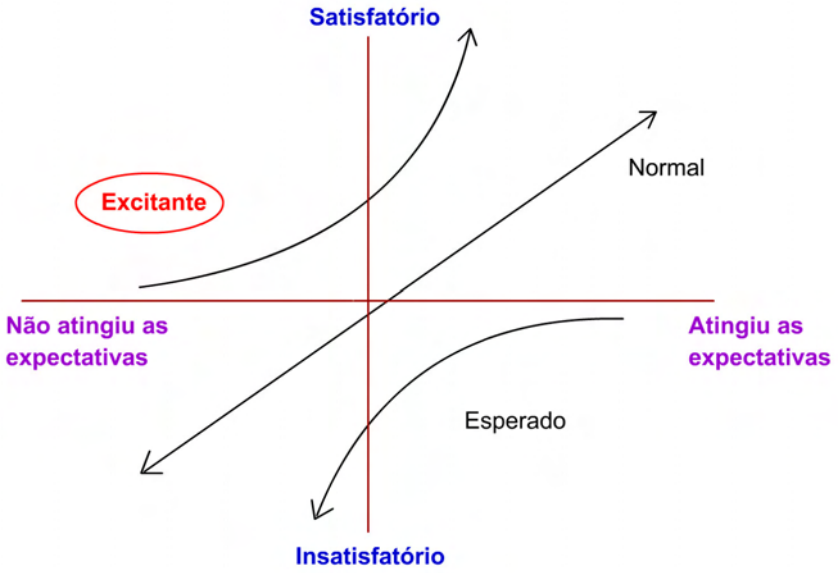
2.2 Outras classificações



Os requisitos esperados podem não ser declarados pelos usuários, mas espera-se que o desenvolvedor os atenda. Se os requisitos forem atendidos, o nível de satisfação dos usuários poderá não aumentar, mas se não forem, eles poderão ficar extremamente insatisfeitos.

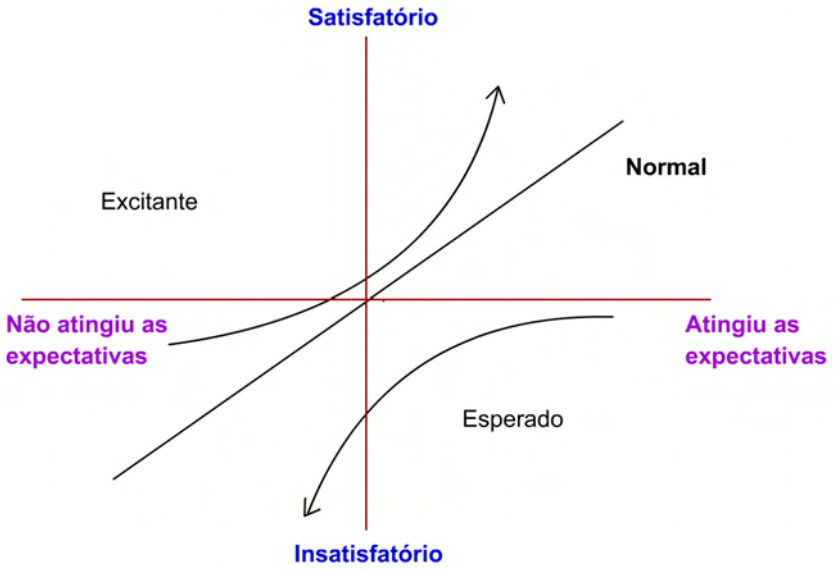
Os requisitos esperados são muito importantes do ponto de vista do desenvolvedor.

2.2 Outras classificações



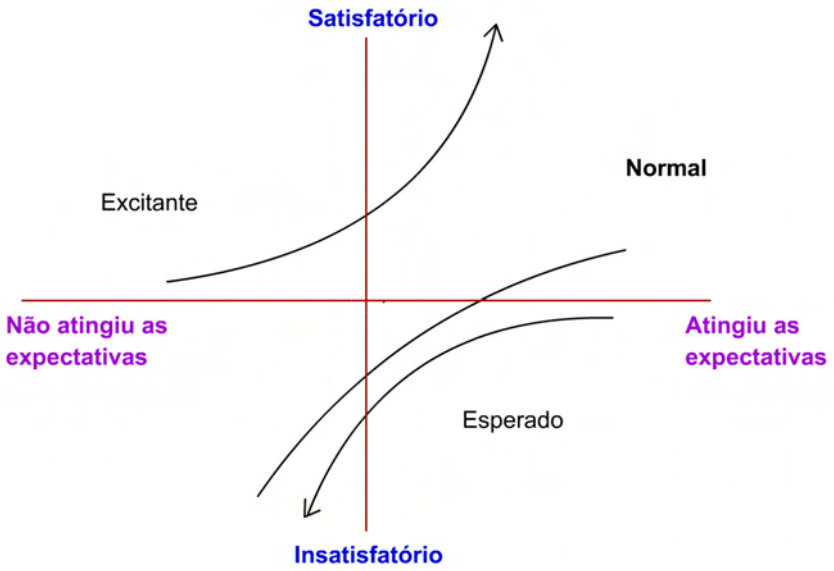
Requisitos acima da expectativa, além de não serem declarados pelos usuários, sequer são esperados por eles. Mas se o desenvolvedor os fornece ao sistema, o nível de satisfação do usuário será muito alto.

2.2 Outras classificações



A tendência ao longo dos anos tem sido a de que os requisitos acima da expectativa se tornem normais e que ...

2.2 Outras classificações



Alguns dos requisitos normais se tornem esperados.

Observe o exemplo na próxima tela:

2.2 Outras classificações

O recurso de auxílio online foi utilizado pela primeira vez no sistema UNIX, na forma de páginas gerenciadas pelo usuário.

Naquela época, era um recurso acima da expectativa. Mais tarde, outros usuários começaram a exigir esse recurso como parte de seus sistemas.

Hoje em dia, os usuários não pedem por isso, mas espera-se que o desenvolvedor forneça esse recurso.

2.2 Outras classificações

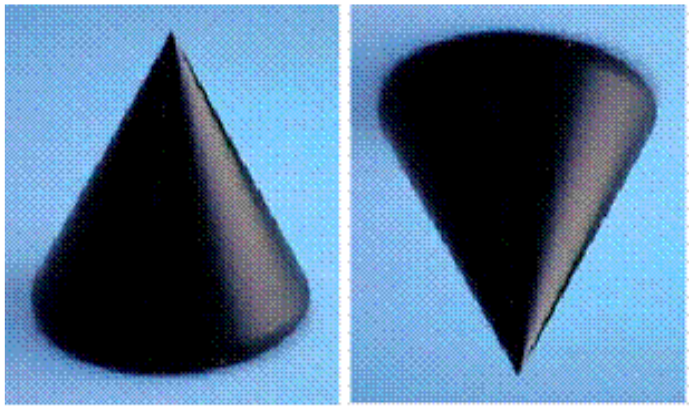
Prioridade:

Forma de priorizar os requisitos. Eles podem ser classificados como:

- Necessários
- Desejáveis
- Não essenciais

Essa classificação é feita em conjunto com os usuários e ajuda na determinação de foco em um modelo de desenvolvimento interativo.

2.2 Outras classificações



Estabilidade:

Os requisitos podem ser estáveis ou instáveis. Requisitos estáveis não mudam freqüentemente, ou pelo menos o período de tempo para mudança é longo.

No entanto, alguns requisitos podem mudar freqüentemente (instáveis). Por exemplo, se estiver ocorrendo alterações no processo do negócio juntamente com o desenvolvimento do software, os requisitos correspondentes podem mudar até que o processo se estabilize.

2.2 Outras classificações

Categorias de usuários:

Como definimos na introdução, haverá muitos interessados em um sistema. De modo geral, eles são de dois tipos:

a) Aqueles que definem as políticas para o sistema

b) Aqueles que utilizam o sistema

Eles podem ainda ser captados entre essas classes dependendo das necessidades de informação e serviços necessários. É importante que todos os interessados sejam identificados e suas exigências sejam captadas.

Conclusão

Parabéns!

Você chegou ao final do **Módulo 2**!

Agora você está apto(a) a continuar este curso.

Siga para o **Módulo 3** e tenha um excelente aprendizado!