

# Engenharia de Software - Engenharia de Requisitos

## Verificação usando Checklist

### Introdução

Elicitar e representar os requisitos de um produto de *software* são atividades críticas no processo de desenvolvimento. Se falhas forem cometidas nesta etapa, irão se propagar para o restante das fases e podem ser percebidas apenas quando já for tarde demais e o produto estiver entregue.

Erros sempre serão cometidos, pois as atividades de desenvolvimento de *software* são de natureza cognitiva e, portanto, dependem de pessoas e pessoas cometem erros.

Já que podemos minimizar, mas não eliminar a introdução de erros, podemos investir em atividades que evitem a sua propagação. Estas são as atividades de verificação de requisitos.

### DESAFIO

Você está se tornando um analista de requisitos experiente e seu gerente pediu a sua ajuda para apoiar um colega que está iniciando.

O colega definiu a lista de requisitos funcionais e não funcionais para um *software* que apoiará reuniões remotas, uma demanda que cresceu no mercado devido ao trabalho remoto ocasionado pela COVID-19.

Foram repassados os seguintes requisitos funcionais e não funcionais:

1. O *software* deverá ter sistema de login
2. Todos os participantes precisam se logar para entrar
3. O *software* deverá usar o sistema de login único da empresa
4. O *software* deverá permitir gerar diversos convites para participantes
5. O organizador poderá colocar todos os microfones no mudo
6. O organizador poderá desabilitar todas as câmeras dos participantes
7. O organizador poderá delegar a função de organizador
8. O *software* deverá possuir um chat
9. O organizador poderá compartilhar a tela com os usuários
10. Um participante só poderá compartilhar a sua tela se o organizador autorizar
11. O *software* deverá permitir dividir os participantes em grupos de trabalho
12. O *software* deverá permitir que o organizador transite entre os grupos de trabalho
13. O *software* deverá permitir que a reunião seja gravada
14. O *software* deverá interoperar com o sistema de login da universidade.
15. O *software* deverá ser fácil de aprender a usar, pois não haverá treinamento na empresa.
16. O *software* deverá se adaptar a diversos tipos de tela, porque tem funcionários que não tem computador em casa.
17. O *software* deverá ser otimizado em relação ao uso de recursos de rede, pois a Internet de alguns funcionários não é boa.
18. O sistema deverá ter mecanismos para evitar erros de operação dos usuários, uma vez que ele pode ser acessado por pessoas menos familiarizadas com a tecnologia.

## Engenharia de Software - Engenharia de Requisitos

### Verificação usando Checklist

Seu gerente lhe pediu que conduza uma revisão por pares usando um checklist de requisitos:

Atributo	Definição	Sim	Não	N/A
<b>Completo</b>	O requisito está especificado de forma completa e que possibilita que o desenvolvedor o implemente?			
<b>Correto</b>	O requisito reflete o que o usuário, cliente ou seus representantes desejam?			
<b>Único</b>	O requisito descreve uma única capacidade, característica, restrição ou atributo de qualidade?			
<b>Viável</b>	O requisito é viável técnica e financeiramente de ser implementado, de acordo com as restrições do projeto?			
<b>Necessário</b>	O requisito tem um motivo de existir, que é representado pelo seu relacionamento a uma fonte de informação e a um objetivo de negócio?			
<b>Priorizado</b>	O requisito possui uma prioridade atribuída para que possa ser alocado a uma versão do software?			
<b>Não-ambíguo</b>	O requisito não contém ambiguidades que levem os stakeholders a interpretá-lo de forma diferente?			
<b>Verificável</b>	O requisito é possível de ser verificado posteriormente quanto à sua implementação?			
<b>Conforme</b>	O requisito está em conformidade com os padrões de especificação estabelecidos, se houver?			