**Item 1 - (Teste 01) -**Funcionalidade Validar Venda - Uma venda financiada só pode ser aprovada caso o valor da parcela seja inferior a 30% do salário do cliente. Essa funcionalidade está validando de forma errada.  
**Item 2 - (Teste 02) -**Funcionalidade Imprimir Resumo Carro - Está funcionalidade está retornando informações incompletas do carro, no teste atual foram identificados apenas quilometragem e valor de venda; e as informações necessárias são de ID, modelo, quilometragem e venda  
**Item 3 - (Teste 03) -**Funcionalidade Validar Idade Cliente - Apenas clientes com idade superior a 18 anos podem ser cadastrados. Entretanto, nos testes, está sendo possível adicionar qualquer idade.  
**Item 4 - (Teste 04) -**Funcionalidade Atualizar Valor Desconto - O programa está retornando um valor errado no resultado de teste, trazendo um valor menos do que esperado.

Como analista e desenvolvedor de software, você recebeu a missão de identificar e descrever a solução para os problemas encontrados.  
  
Faça uma descrição da correção necessária para cada um dos resultados do teste. Não é necessário o envio do código implementado, apenas a descrição e elementos que indiquem o que deve ser feito para que os testes sejam aceitos pela equipe de qualidade.  
  
Após, descreva a resolução dos itens referentes à implantação da metodologia ágil Scrum, para o desenvolvimento do software apresentado.

RESPOSTAS:

Item 1: A validação da venda está incorreta, pois o cálculo considera que a parcela deve ser exatamente igual a 30% da renda do cliente, o correto é verificar se a parcela é **inferior a 30% da renda do cliente**.

Para a resolução do problema é necessário mudar a logica e o calulo do método “validarVenda”, linha 25; uma das possibilidades de resolução do problema é, na linha 27, substituir o calculo, “renda – 0.3”, por “renda \* 0.3” que representa o calculo correto para identificar 30% da renda do cliente; descoberto quanto vale a 30% da renda do cliente, é necessário mudar a logica da linha 28, com o operador relacional “<”, realizando a comparação entre as variáveis, “valorParcela” e “valorReferencia”, ficando da seguinte forma, “resultado = valorParcela < valorReferencia”, retornando um valor booleano; possibilitando a resolução deste problema de acordo com as necessidades e requerimento do cliente.

Item 2: Para a resolução do problema encontrado no teste 2, uma das possibilidades de resolução é, excluir as linhas 19, 20, 21, e adicionar as variáveis, informações requisitadas que estão faltando como “ID” e ”Modelo” com a função “this.” para retornar o conteúdo das variáveis ao resumo, diretamente na função “return” do método “imprimirResumoCarro”; simplificando o código e facilitando seu entendimento, desta forma:

( return "ID: " + this.id + "\n" + "Modelo: " + this.modelo + "\n" + "Quilometragem: " + this.quilometragem + "\n" + "Valor de Venda: " + this.valorVenda; ).

Outro forma de resolução deste problema é, apagar o “;” do final da linha 19 adicionar o operado aritmético “+” e apagar a “msg =” da linha 20, adicionando o conteúdo da linha 19 e vinte contemplando a resolução do teste 2.

Item 3: Para limitarmos o cadastro apenas à maiores de 18 anos, será necessário adicionar uma condição, se idade for menor que 18 interrompa o cadastro, ao método ”cadastrarCliente”.

Há diversas formas de realizar esta implementação, com “if”, apenas adicionado a condição após a linha 15, se “if” idade for menor “<” que 18, então retorne uma mensagem “JoptionPane” cadastro não permitido, interrompendo o cadastro com “return” dentro da condição. Já a outra forma de implementar esta condição é criar um laço de repetição com “while”; apagando a linha 15, criando uma variável booleana para controle do laço, na linha seguinte é introduzido o “while” com a variável de controle criada; dentro do laço, criamos o direcionamento para a variável “idade” com “this” e utilizamos o “integer” para inserir o valor “idade”, na linha seguinte criamos a condição com “if” se o conteúdo da variável ”this.idade” for maior ou igual a 18 a variável de controle utilizada no “while” recebe o valor booleano “true” permitindo o cadastro; caso a variável “this.idade” receber um valor menos que 18, através da condição “else” retorna uma mensagem com “JOptionPane” comunicando que o cadastro não foi permitido, finalizando o laço com “return” na linha seguinte.

Essas são algumas das forma de resolução dos problemas encontrados no teste 3.

**Item 4:** Para corrigir o problema encontrado, pelo “QA”, será necessário fazer a substituição do operador de atribuição por um operador de decremento, de “\*=” para “-=”, e ajustar a logica do calculo utilizado. O calculo usado para encontrar o percentual de desconto esta correto, “percentualDesconto / 100”, porem para alcançarmos o resultado esperado com a logica utilizada, será necessário multiplicarmos o resultado deste calculo ao valor da venda, “this.valorVenda \* (percentualDesconto / 100)”; após a realização deste calculo, será necessário decrescer o resultado encontrado do valor total da venda representado pela variável “this.valorVenda”,esse decrescimento ocorre através do operador de atribuição substituído, como instruído a cima; resultando esta logica, this.valorVenda -= (this.valorVenda \* (percentualDesconto / 100)).

Há a possibilidade de utilizar outras logicas para encontrar o mesmo resultado requerido e esperado pelo cliente, porem para manter a coerência e logica já utilizada visando um código claro e legível, estes ajustes contemplam o resultado esperado.

**Item 5:** Entre as vantagens da metodologia “Scrum” podemos citar a flexibilidade e adaptabilidade que permiti se adequar a necessidades dos “stakehoulders” de forma abrangente e sem burocracia, possibilitando uma comunicação efetiva entre as equipes envolvidas e os “stakehoulders”, através de reuniões diárias e pontuais; entrega continua de valor a cada final de “ciclo de vida” do projeto, “sprints”, afim de contemplar as necessidades do cliente e sua satisfação; permite também a melhoria continua através das retrospectivas que ocorrem ao final de cada “sprint”, identificando os pontos fortes e fracos de cada ciclo possibilitando a implementação de melhorias continuas.

Entre as desvantagens desta metodologia está na dependência e comprometimento da equipe envolvida, que deve estar engajada de forma autônoma e independente no projeto do inicio ao fim; outra desvantagem é a implementação em projetos de grande escala devido a dificuldade de coordenação de diversos times, que detêm características autônomas, multidisciplinar e independentes; outra grande desvantagem é a falta de uma documentação formal, por se tratar de uma metodologia flexível iterativa e incremental, há uma dificuldade em manter uma documentação robusta e abrangente.

**Item 6:** Entre as funcionalidades da metodologia ágil “scrum” são:

****Sprint:** são ciclos de trabalho, projetos, com uma duração fixa de 1 a 4 semanas; ao inicio de cada “sprint” é realizada uma reunião de planejamento que define as prioridades seja ela uma funcionalidade ou incremento de uma nova funcionalidade.**

****Product Backlog:** É a lista de prioridades, compostas por funcionalidades e requisitos do projeto, geralmente é o “Product Owner” que gerencia as prioridades do projeto em cada “sprint”.**

****Daily Scrum:** São reuniões diárias de 15 minutos, que tem o objetivo de alinhar a equipe afim de identificar bloqueios e planejar o trabalho do dia; com uma estrutura de perguntas e respostas, onde cada membro da equipe responde, oque foi feito ontem?, o que será feito hoje?, e se, há algum impedimento?.**

****Sprint Review e Retrospective:** As “review” são realizadas ao final de cada “sprint” afim de apresentar os resultados e incrementos realizados aos “stakeholders”; e a “restrospective” é uma reunião realizada também ao final de cada “sprint” com a finalidade de identificar os pontos de melhoria nos processos e na colaboração das equipes, afim de manter a equipe engajada e alinhada.**