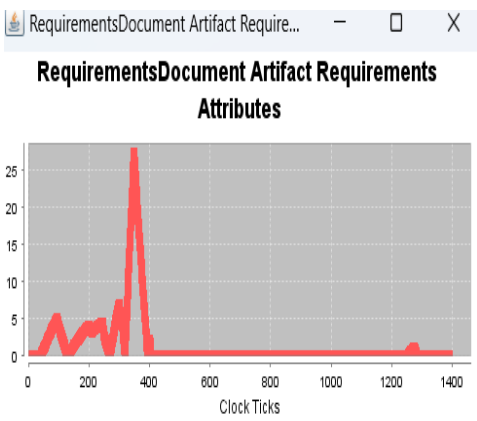
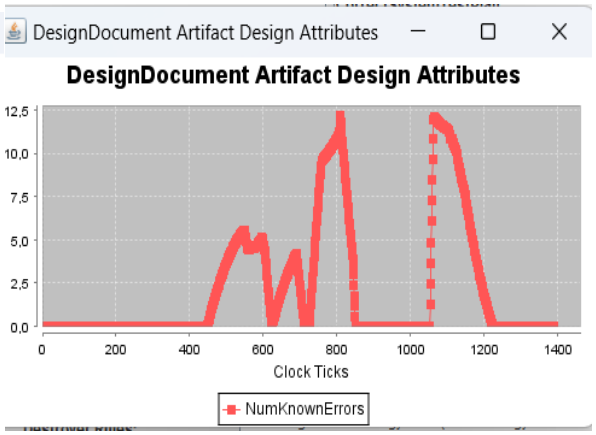
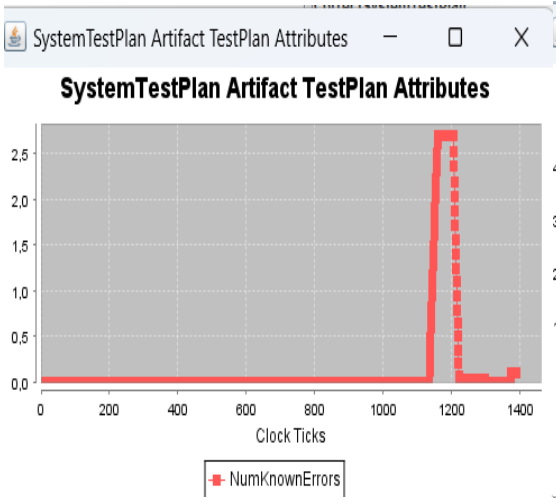
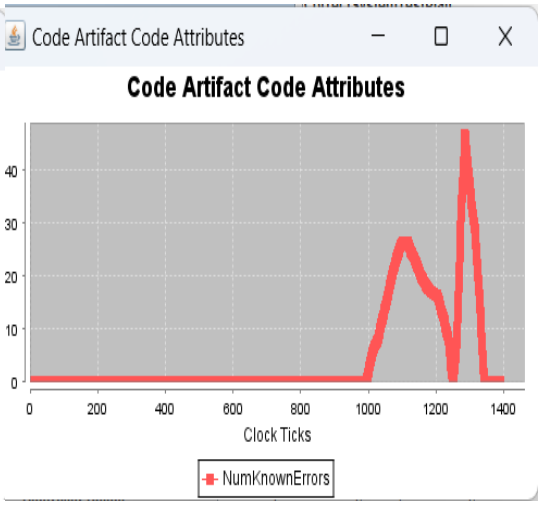
ALUNOS: Carlos Patrick Milak RA: 172320772 ; Lucas RA: 172316669 ; Natan RA: 172312946

MODELO CASCATA https://animaedu-my.sharepoint.com/personal/172320772_ulife_com_br/Documents/ALUNOS.docx?web=1









Pontos Positivos:

Estrutura Clara: O modelo cascata tem uma estrutura clara e fácil de entender. Cada fase tem seus objetivos específicos, o que simplifica o desenvolvimento.

Documentação Abrangente: Como cada fase requer documentação detalhada, isso pode ser benéfico para rastreamento, manutenção e compreensão do sistema.

Fases Bem Definidas: As fases (requisitos, design, implementação, testes, manutenção) são bem definidas, facilitando a gestão do projeto.

Adequado para Projetos Pequenos e Bem Compreendidos: Pode funcionar bem quando os requisitos são claros desde o início e não se espera muita mudança ao longo do projeto.

Pontos Negativos:

Inflexibilidade: O modelo cascata é inflexível em relação a mudanças nos requisitos. Uma vez que uma fase é concluída, é difícil voltar atrás sem começar do zero.

Feedback Limitado: Os clientes geralmente só veem o produto final, o que pode levar a feedback tardio e dificuldades na adaptação às necessidades do cliente.

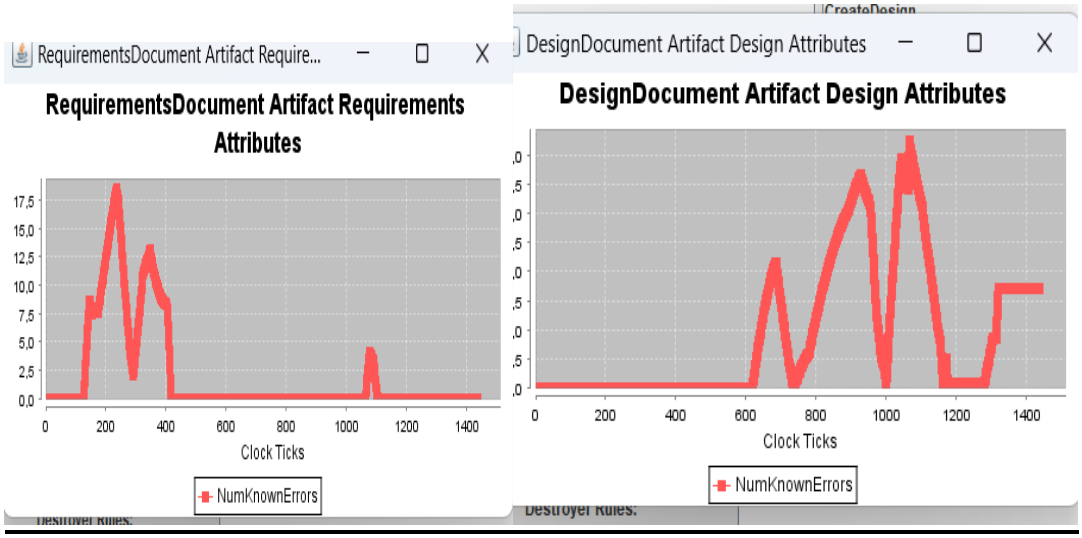
Riscos no Final do Projeto: Os testes são realizados no final do ciclo, o que significa que falhas podem não ser identificadas até que o produto esteja quase pronto para o lançamento.

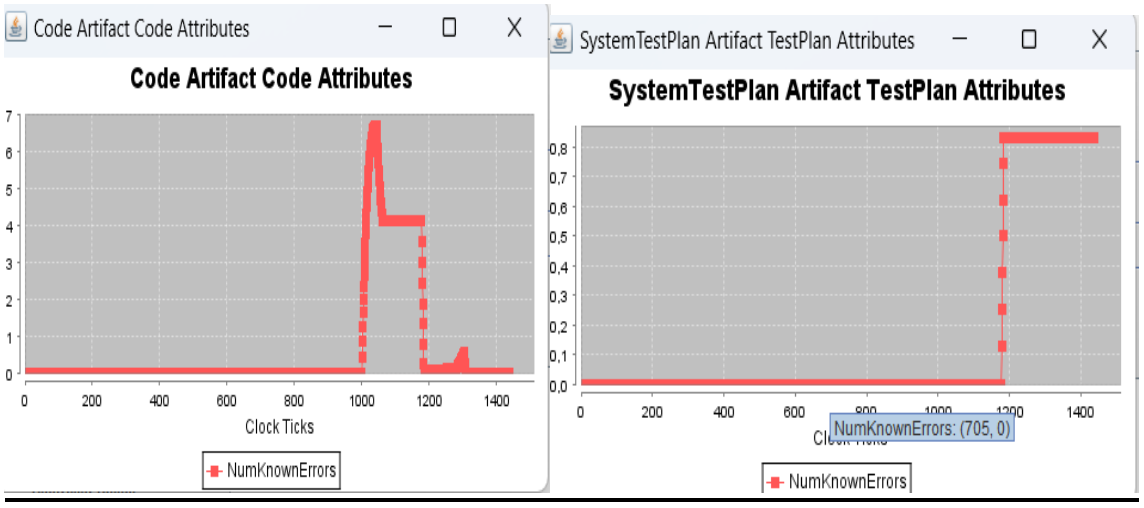
Longo Tempo de Entrega: Pode resultar em um longo tempo de entrega do produto final, especialmente se houver atrasos em qualquer fase do projeto.

Não Acomoda Mudanças de Requisitos: Mudanças nos requisitos durante o desenvolvimento podem ser difíceis de serem incorporadas sem retroceder às fases anteriores.

Não é Adequado para Projetos Complexos e de Grande Escala: Pode não ser a escolha ideal para projetos complexos, onde os requisitos estão sujeitos a mudanças significativas ao longo do tempo.

MODELO INCREMENTAL





Pontos Positivos:

Entrega Rápida de Funcionalidades: O modelo incremental permite a entrega rápida e contínua de partes funcionais do software, proporcionando benefícios tangíveis ao cliente ao longo do tempo.

Feedback Contínuo: Os clientes podem fornecer feedback ao longo do desenvolvimento, o que facilita a adaptação às mudanças nos requisitos e melhora a satisfação do cliente.

Maior Flexibilidade: O modelo é mais flexível para acomodar mudanças nos requisitos, já que cada incremento pode ser ajustado com base no feedback do cliente ou nas necessidades emergentes.

Redução de Riscos: Os riscos são mitigados ao distribuir o desenvolvimento ao longo de várias iterações. Falhas podem ser identificadas e corrigidas mais cedo no processo.

Melhoria Contínua: Cada incremento é uma oportunidade de aprendizado. As lições aprendidas em uma iteração podem ser aplicadas nas próximas, melhorando continuamente o processo e o produto.

Pontos Negativos:

Complexidade de Gerenciamento: Gerenciar vários incrementos simultaneamente pode ser complexo, especialmente quando as dependências entre os incrementos não são claras.

Possibilidade de Integração Difícil: Se não houver uma estratégia adequada de integração entre os incrementos, pode haver desafios na fusão e teste de todas as funcionalidades juntas.

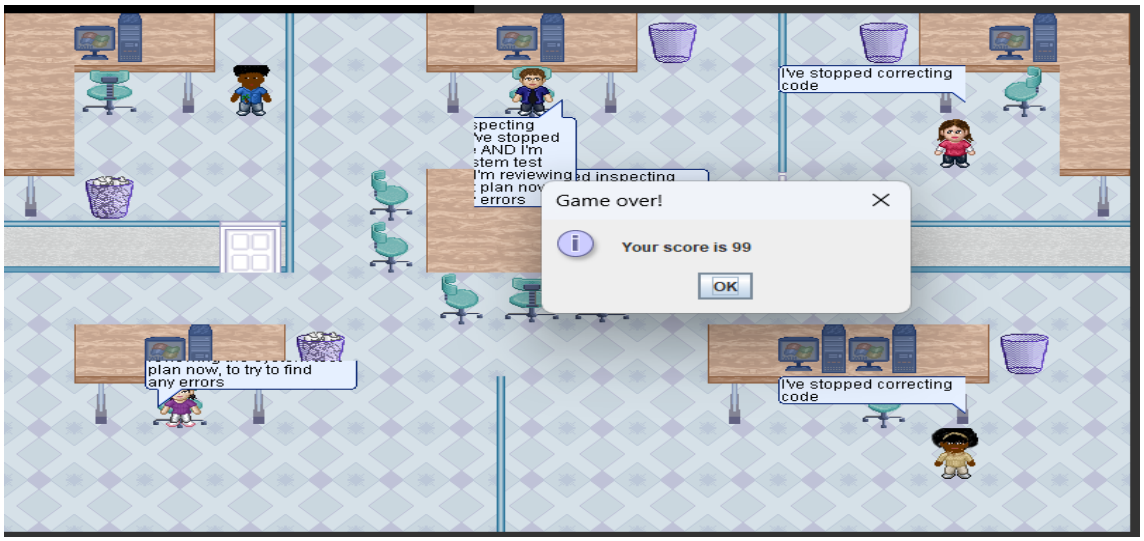
Necessidade de Planejamento Detalhado: O sucesso do modelo incremental depende de um planejamento detalhado para garantir que os incrementos sejam compatíveis e que o produto final atenda aos requisitos.

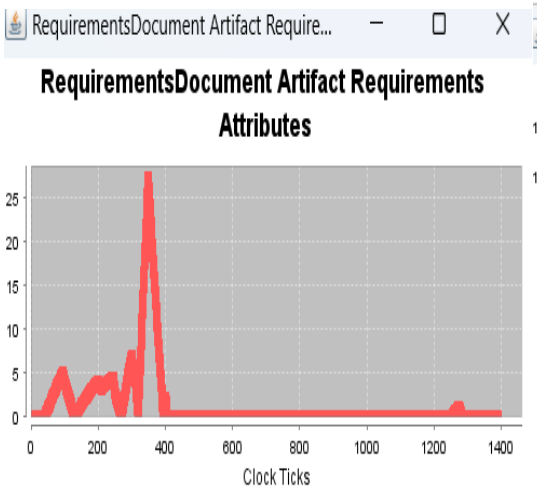
Custo Incremental: O desenvolvimento incremental pode resultar em custos adicionais, pois cada incremento pode exigir esforço adicional de desenvolvimento, teste e documentação.

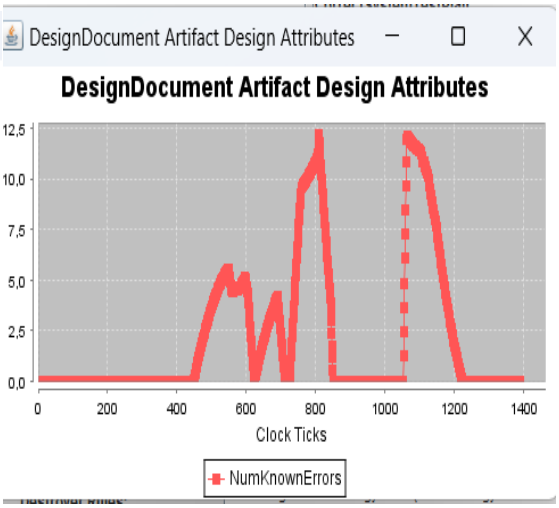
Requisitos Iniciais Essenciais: Para ser eficaz, o modelo incremental requer uma compreensão clara dos requisitos iniciais. Se os requisitos básicos mudarem drasticamente, o modelo pode não ser tão eficiente.

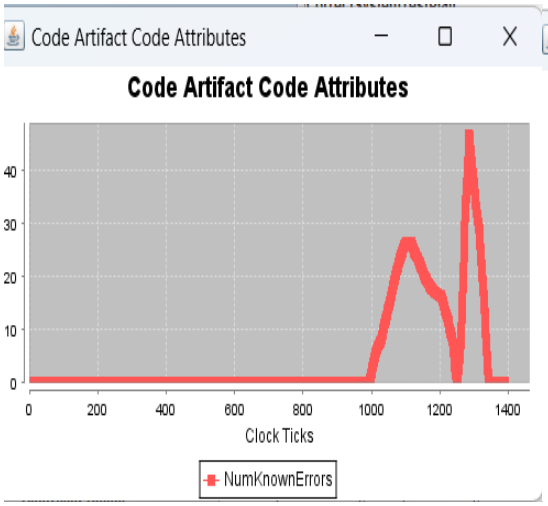
Dificuldade em Prever o Escopo Total: Pode ser desafiador estimar o esforço total necessário, pois o escopo exato pode evoluir com base nos feedbacks e nas mudanças nos requisitos ao longo do tempo.

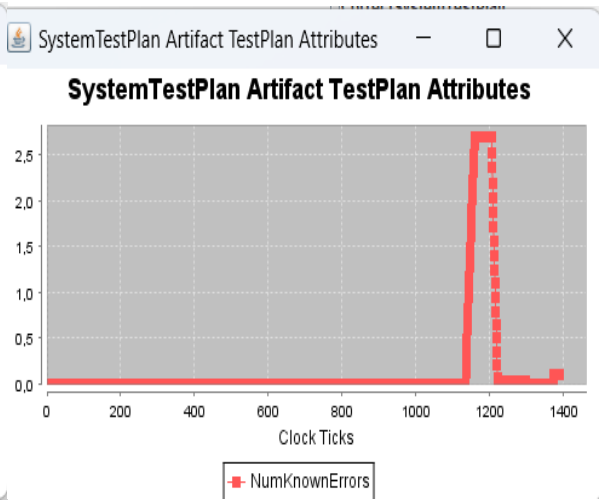
MODELO RUB


I've stopped correcting code
Game over!
Your score is 99
OK
plan now, to try to find any errors
I've stopped correcting code


RequirementsDocument Artifact Require...
RequirementsDocument Artifact Requirements Attributes
Clock Ticks


DesignDocument Artifact Design Attributes
DesignDocument Artifact Design Attributes
Clock Ticks
NumKnownErrors


Code Artifact Code Attributes
Code Artifact Code Attributes
Clock Ticks
NumKnownErrors


SystemTestPlan Artifact TestPlan Attributes
SystemTestPlan Artifact TestPlan Attributes
Clock Ticks
NumKnownErrors

Pontos Positivos:

Abordagem Iterativa e Incremental: O RUP utiliza uma abordagem iterativa e incremental, permitindo que o desenvolvimento seja realizado em pequenas iterações, o que facilita o gerenciamento de mudanças nos requisitos.

Foco na Qualidade e na Arquitetura: Coloca ênfase na qualidade do software e na definição e refinamento contínuo da arquitetura do sistema, resultando em produtos mais robustos e de alta qualidade.

Adaptabilidade: Pode ser adaptado para se adequar a diferentes tipos de projetos, desde pequenos a grandes, simples a complexos.

Gestão de Riscos: O RUP incorpora a gestão de riscos ao longo do processo, identificando e mitigando riscos desde o início do projeto.

Documentação Abundante: O processo RUP enfatiza a criação de documentação detalhada, o que pode ser valioso para rastrear decisões, requisitos e o estado do projeto.

Envolvimento do Cliente: Inclui mecanismos para a participação contínua do cliente, promovendo uma comunicação mais próxima e feedback regular.

Pontos Negativos:

Complexidade: O RUP pode ser percebido como complexo, especialmente para projetos pequenos. A quantidade de documentação e a formalidade podem ser excessivas para algumas equipes.

Custo e Tempo: O processo RUP pode ser intensivo em termos de custo e tempo, especialmente durante as fases iniciais, devido à necessidade de planejamento, definição da arquitetura e documentação.

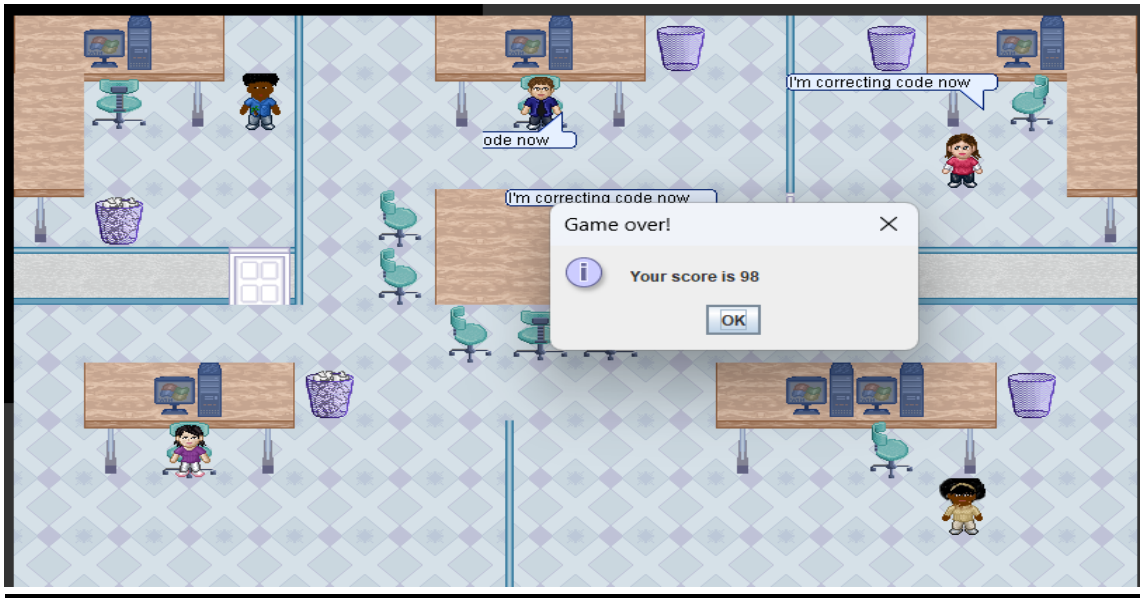
Necessidade de Treinamento: A implementação efetiva do RUP pode exigir que a equipe seja treinada nas práticas e princípios do processo, o que pode aumentar o tempo e os custos iniciais.

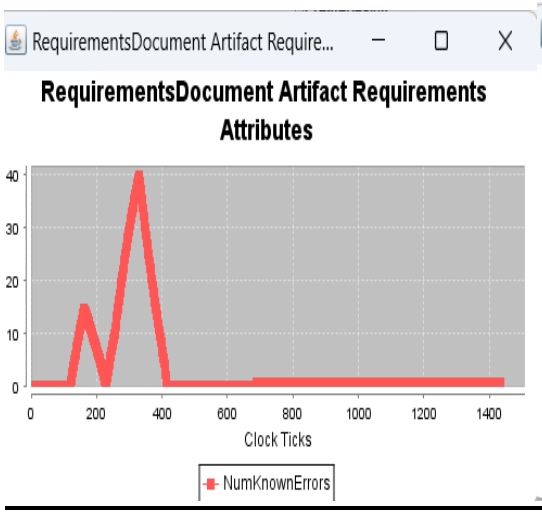
Adoção Pode Ser Desafiadora: Algumas equipes podem achar difícil adotar completamente o RUP, especialmente se estiverem acostumadas com modelos de desenvolvimento menos formalizados.

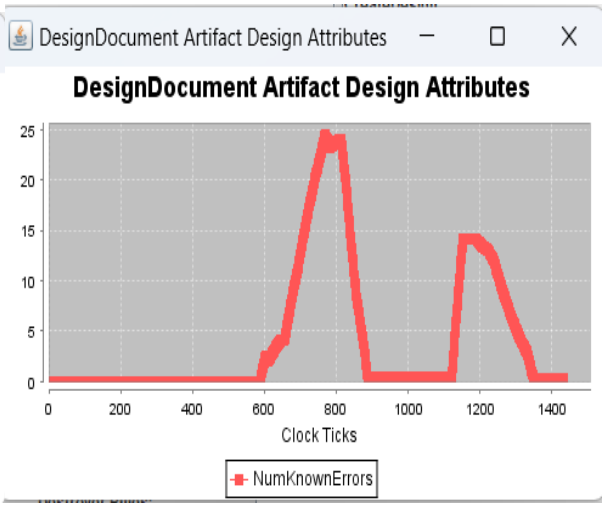
Fase de Iniciação Pode Ser Longa: A fase inicial do RUP, dedicada ao planejamento, definição da arquitetura e elaboração, pode ser extensa, o que pode ser visto como um desafio em projetos que exigem entrega rápida.

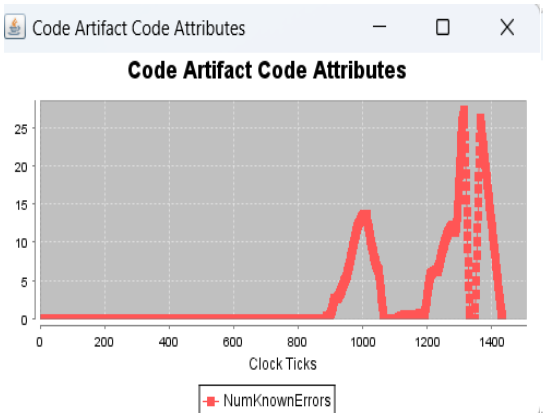
Customização Necessária: Para se adequar a um projeto específico, o RUP frequentemente requer customização, o que pode adicionar complexidade ao processo.

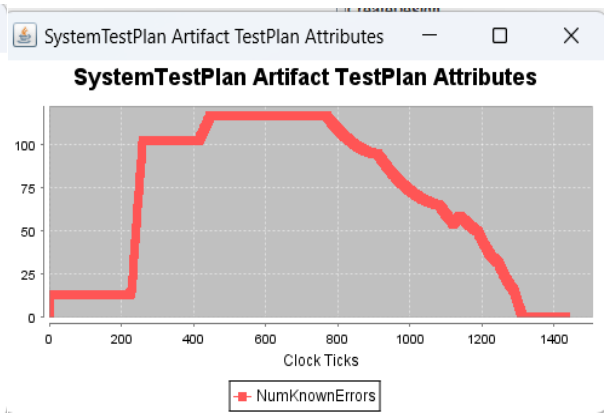
MODELO PROTOTIPAÇÃO











Pontos Positivos:

Feedback do Cliente: A prototipação permite obter feedback contínuo do cliente durante o desenvolvimento, o que ajuda a garantir que o produto final atenda às expectativas.

Compreensão Aprofundada dos Requisitos: Ao criar protótipos, os desenvolvedores e clientes podem ter uma compreensão mais clara dos requisitos, ajudando a evitar mal-entendidos e melhorando a precisão na definição do escopo.

Adaptação a Mudanças de Requisitos: A natureza iterativa da prototipação facilita a incorporação de mudanças nos requisitos ao longo do desenvolvimento, já que o protótipo pode ser ajustado conforme necessário.

Minimização de Riscos: A prototipação permite identificar riscos e desafios no início do processo, reduzindo a probabilidade de erros significativos no produto final.

Aumento da Satisfação do Cliente: A interação contínua com protótipos ajuda a garantir que o cliente esteja envolvido no processo de desenvolvimento, aumentando a satisfação e a aceitação do produto.

Pontos Negativos:

Custo e Tempo: A criação de protótipos adiciona tempo e custos ao processo de desenvolvimento. Dependendo da complexidade do sistema, a elaboração de protótipos pode exigir recursos significativos.

Possível Falta de Foco na Arquitetura: Em alguns casos, a ênfase na rápida produção de protótipos pode levar a uma falta de atenção à arquitetura do sistema, resultando em desafios de escalabilidade e manutenção.

Dificuldades na Transição para o Produto Final: Às vezes, os protótipos podem ser construídos com tecnologias diferentes das planejadas para o produto final, o que pode levar a desafios na transição.

Risco de Protótipo Ser Aceito como Produto Final: Existe o risco de que o protótipo seja aceito como o produto final, mesmo que não seja completamente representativo das capacidades e requisitos necessários.

Necessidade de Boa Comunicação: A eficácia da prototipação depende de uma comunicação clara entre desenvolvedores e clientes. Se a comunicação falhar, o protótipo pode não atender às expectativas.

Requisitos Não Funcionais Podem Ser Desconsiderados: Em um foco excessivo nos aspectos funcionais, os requisitos não funcionais, como desempenho e segurança, podem ser negligenciados.