

Processo de Manutenção de Software

DroidFoundry

1) Propósito

O processo de manutenção de Software na DroidFoundry é importante para a melhoria contínua do sistema, esse processo atua com o intuito de reportar, classificar e gerenciar as anomalias encontradas e melhorias requeridas.

2) Definições

As definições deste processo foram definidas com base nas normas IEEE 1044-2008 e IEEE/ISO/IEC 14764-2006:

- *Engano*: Ação humana que produz um defeito.
- *Defeito*: Passo, processo ou definição de dados incorretos.
- *Erro*: Um estado inconsistente ou inesperado provocado por um defeito em um *software*.
- *Problema*: Dificuldade ou incerteza vivenciada por uma ou mais pessoas, podendo ser causado por uma ou mais falhas. Uma falha pode causar um ou mais problemas.
- *Falha*: Resultado obtido pela execução de um *software* que é diferente do esperado.
- *Solicitação de Mudança*: Se refere à artefatos que definem um conjunto de descrições para que uma mudança no *software* seja realizada.
- *Manutenções Evolutivas*: envolvem as manutenções responsáveis por inserir ou modificar os requisitos funcionais e não funcionais do software, assim como as manutenções executadas para melhorar a manutenibilidade do sistema, reduzindo o esforço necessário para realização das manutenções futuras. Nessa categoria incluem-se as alterações que almejam melhorar a compreensão sobre o software, reduzir a complexidade do código ou eliminar a degradação inserida por outras atividades de manutenção. De acordo com Sommerville (2007), as solicitações perfectivas representam em média 65% do esforço total da fase de manutenção.
- *Manutenções Adaptativas*: representam as manutenções executadas com o objetivo de adaptar o software às mudanças do ambiente, como por exemplo, alterações nas configurações de hardware e de sistema operacional. Segundo Sommerville(2007), esse tipo de solicitação representa em média 18% do esforço total da manutenção de software.
- *Manutenções Corretivas*: contemplam as mudanças acarretadas por falhas no processamento das funcionalidades ou devido a ausência de atributos especificados, como por exemplo, desempenho ou segurança e que comprometem o uso do sistema. Sommerville (2007) destaca que esse tipo de solicitação representa em média 17% do esforço demandado durante a fase de manutenção.
- *Manutenções Preventivas*: abrangem as correções diagnosticadas e concluídas antes que se tornem um problema real. Esse tipo de manutenção é parte de uma política de gerenciamento pró-ativo da fase de manutenção. Sommerville (2007) não apresenta uma estatística sobre o percentual do esforço demandado pela execução das solicitações preventivas.

3) Escopo de Aplicação

O processo de manutenção tem a função de dar suporte e acompanhamento a produtos já entregues à utilização, ou seja, que já estão em uso pelos usuários. Participará desta área todos os membros da organização, pois todas as etapas serão retomadas: planejamento, análise, implementação e testes.

O Processo de Manutenção da DroidFoundry foi projetado de modo a não ser específico para indústria e tecnologia, ele é modular e pode ser estendido de forma flexível; pode ser usado tanto em aplicações para um único usuário como em soluções complexas para múltiplos usuários.

4) Políticas

- Todas as anomalias encontradas devem ser reportadas através do Documento de Solicitação de Mudanças, por qualquer indivíduo.
- O Documento de Solicitação de Mudança será analisado pela Equipe de Manutenção, que avaliará a viabilidade do pedido. Caso este seja aprovado, um pedido de autorização será feito ao Gerente de Projetos para que a manutenção seja iniciada. Este fará um reunião com os envolvidos, onde será feita a análise de riscos, rastreabilidade e planejamento da mudança. Caso o pedido seja barrado em alguma dessas etapas, deverá ser feito um relatório explicando os motivos da não realização da mudança.

5) Métricas

Alguns indicadores de desempenho do Processo de Manutenção são a quantidade de manutenções corretivas aplicadas, análise de efeitos colaterais após a manutenção, quantidade de riscos previsíveis que ocorreram e se a manutenibilidade do software foi comprometida ou não após as modificações feitas.

Uma métrica eficiente para avaliar a manutenção foi descoberta em um estudo conduzido por Banker e Slaughter (1997) para uma modelo de análise chamada *Data Envelopment Analysis* (DEA). DEA é um modelo utilizado para estimar a eficiência produtiva de unidades com base na relação entre os produtos ou serviços produzidos (*outputs*) e os recursos consumidos para produzi-los (*inputs*).

A eficiência dos projetos de manutenção é determinada com base na relação entre a quantidade de Pontos de Função (PF) produzidos e o número de horas demandadas para implementar os pontos de função. Em outras palavras, os projetos eficientes são aqueles em que é possível implementar a maior quantidade de pontos de função com o menor número de horas trabalhadas. A função é dada por:

$$\chi = f(y, z) * \theta_i$$

Onde χ representa a eficiência produtiva, mensurada por meio da quantidade de recursos (horas trabalhadas) necessários para concluir os projetos de manutenção; f denota a função de avaliação da produção do modelo DEA; y determina o volume de produção obtido com a execução das modificações realizadas para atender aos requisitos que compõem as solicitações de manutenção – mensurado através da métrica Pontos de Função (PF) –; z é o vetor de características dos projetos avaliados, composto pelas seguintes variáveis:

Experiência da equipe, número de plataformas de hardware envolvidas no projeto, número de

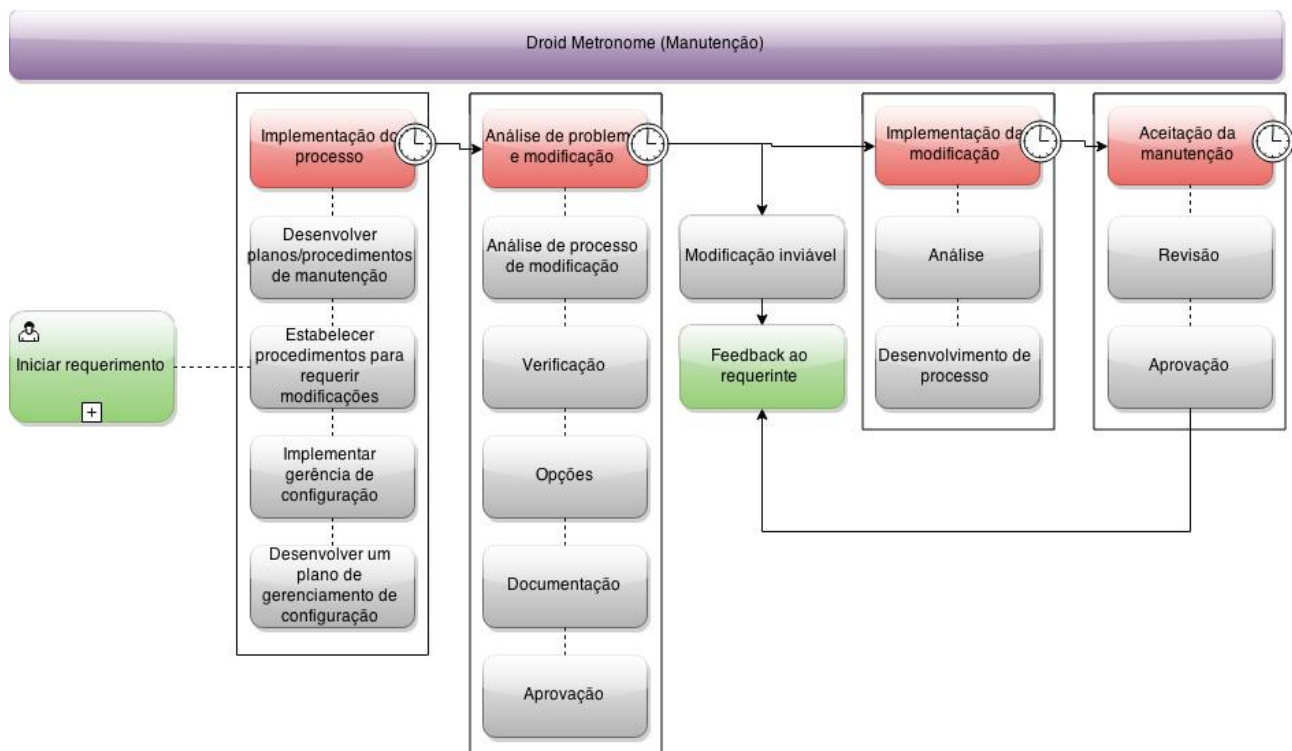
técnicas de design utilizadas durante a execução do projeto, quantidade de ferramentas de desenvolvimento requeridas e número de ferramentas e técnicas de testes requeridas; θ_i representa o índice de ineficiência do projeto i , dado pela quantidade de requisitos implementados divididos pelos requisitos compreendidos na solicitação de mudança:

$$\theta_i = R_i / R_c$$

Segundo Banker e Slaughter (1997), este modelo estima a eficiência das unidades identificando possíveis ganhos variáveis de escala.

Caberá ao responsável pela Garantia da Qualidade a escolha da melhor métrica de avaliação da manutenção baseado no projeto que está sendo desenvolvido com base nos indicadores de desempenho do processo. Esses indicadores devem mostrar a eficiência e a eficácia do processo.

6) Macro Fluxo



7) Definição de Atividades

Definir cada atividade do fluxo do processo de acordo com o modelo a seguir:

Atividade 1 Estabelecer plano do projeto de manutenção

Realização: Gerente de projeto.

Aprovação: Não se aplica.

Responsabilidades **Colaboração:** Equipe de desenvolvimento e testes ,devido a tarefa de implantação.

Informação: Equipe de desenvolvimento, testes e Gerente de projeto.

Tarefas

1.1. Desenvolver procedimentos de manutenção.

1.2. Estabelecer procedimentos para requerir modificações.

| | |
|---------------------------|---|
| | 1.3. Implementar gerência de configuração. |
| Pré-Condições | Nenhum critério específico. |
| Entradas | Baselines relevantes, documentação do sistema (se disponível) e requisição de modificações (se aplicáveis). |
| Critérios de Saída | Processo aprovado pela direção. |
| Produtos | Documento de solicitação de mudança. |
| Infraestrutura | Para desenvolver esta atividade é necessário o esforço conjunto de todos os integrantes do projeto além do uso de computadores conectados a internet para a comunicação entre a equipe. Softwares de vídeo-conferência como Skype ,hangout ou curse voice. |
| Ferramentas | Softwares de edição em conjunto de documentos tais como google drive, GitHub Issues. |
| Atividade 2 | Analisar problemas e modificações |
| Responsabilidades | Realização: Equipe de manutenção. |
| | Aprovação: Equipe de manutenção. |
| | Colaboração: Não se aplica. |
| | Informação: Equipe de desenvolvimento e testes. |
| | 2.1. Analisar processo de modificação. |
| Tarefas | 2.2. Verificar. |
| | 2.3. Encontrar possíveis opções alternativas. |
| | 2.4. Documentar. |
| | 2.5. Aprovar. |
| | |
| Pré-Condições | O plano de projeto de manutenção deverá ter sido implementado. |
| Entradas | Documento de solicitação de mudança. |
| Critérios de Saída | Processo deverá ser considerado viável pela equipe de manutenção. |
| Produtos | Documento de análise de viabilidade e riscos. |
| Infraestrutura | Para desenvolver esta atividade é necessário o esforço conjunto de todos os integrantes do projeto além do uso de computadores conectados a internet para a comunicação entre a equipe. Softwares de vídeo-conferência como Skype, hangout ou curse voice. |
| Ferramentas | Softwares de edição em conjunto de documentos tais como google drive, GitHub Issues. |
| Atividade 3 | Implementação da modificação |
| Responsabilidades | Realização: Equipe de manutenção. |
| | Aprovação: Equipe de manutenção. |
| | Colaboração: Equipe de desenvolvimento e testes. |
| | Informação: Requerintes da manutenção. |
| | 3.1. Analisar. |
| Tarefas | 3.2. Aplicar plano de projeto de manutenção. |
| | |

| | |
|---------------------------|--|
| Pré-Condições | O documento de análise de viabilidade e riscos deverá ter sido gerado e aprovado. |
| Entradas | Documento de análise de viabilidade e riscos. |
| Critérios de Saída | A modificação deverá ter sido implementada e testada. |
| Produtos | Documento de controle e correção. |
| Infraestrutura | Para desenvolver esta atividade é necessário o esforço conjunto de todos os integrantes de desenvolvimento, manutenção e testes além do uso de computadores conectados a internet para a comunicação entre a equipe. Softwares de vídeo-conferência como Skype, hangout ou curse voice. |
| Ferramentas | Softwares de controle de versão como git. IDE's para manutenção de código fonte. |
| Atividade 4 | Aceitação da manutenção Realização: «Equipes de manutenção ,desenvolvimento e testes. |
| Responsabilidades | Aprovação: «Não se aplica. Colaboração: « Não se aplica. Informação: «Requerintes da manutenção. |
| Tarefas | 4.1. Revisar manutenção. 4.2. Aprovar manutenção. |
| Pré-Condições | Uma modificação no projeto de software feita. |
| Entradas | Documento de controle e correção. |
| Critérios de Saída | Todas as modificações efetuadas no sistema deverão ser reportadas. |
| Produtos | Notas de release. |
| Infraestrutura | Para desenvolver esta atividade é necessário o esforço conjunto de todos os integrantes de desenvolvimento, manutenção e testes além do uso de computadores conectados a internet para a comunicação entre a equipe. Softwares de vídeo-conferência como Skype, hangout ou curse voice. |
| Ferramentas | Softwares de edição em conjunto de documentos tais como google drive. Softwares de controle de versão como git. |
| Atividade 5 | Migração do software Realização: Equipe de desenvolvimento. |
| Responsabilidades | Aprovação: Gerente do projeto. Colaboração: Stakeholders. Informação: Stakeholders do projeto. |
| Tarefas | 5.1. Analisar requisitos de migração e seu impacto. 5.2. Estabelecer um cronograma para realização da migração. 5.3. Identificar, desenvolver e/ou adquirir ferramentas para migração. 5.4. Executar operações de conversão e migração dos produtos de software e |

seus dados para o novo ambiente.

5.5. Verificar a migração através de testes.

5.6. Identificar e prover suporte à ambientes antigos.

| | |
|---------------------------|--|
| Pré-Condições | Um motivo para o projeto ser migrado. |
| Entradas | Documento de solicitação de mudança para migração do produto de software. |
| Critérios de Saída | A compatibilidade de funcionamento do produto no novo ambiente deve ser verificada. Plano de migração (ISO/IEC 12207); |
| Produtos | Produto de software migrado. |
| Infraestrutura | Para desenvolver esta atividade é necessário o esforço conjunto de todos os integrantes da equipe responsáveis pelo sistema a ser migrado. Softwares de vídeo-conferência como Skype, hangout ou curse voice. Softwares de edição em conjunto de documentos tais como google drive, GitHub Issues, Wiki. |
| Ferramentas | Softwares de controle de versão como git. IDE's para migração do código fonte. |
| Atividade 6 | Aposentadoria de software |
| Responsabilidades | Realização: Analista do sistema. Aprovação: Gerente do projeto. Colaboração: Não se aplica. Informação: Requerintes da manutenção. |
| Tarefas | 6.1. Gerar plano de aposentadoria. 6.2. Notificar os interessados. 6.3. Notificar a realização da aposentadoria. 6.4. Arquivar os dados do sistema aposentado. |
| Pré-Condições | Um motivo para o projeto ser descontinuado. |
| Entradas | O sistema que será aposentado. |
| Critérios de Saída | Todas as atividades referente ao sistema aposentado devem ser encerradas e seus dados mantidos para fins de pesquisa. |
| Produtos | Nota de aposentadoria do software. |
| Infraestrutura | Para desenvolver esta atividade é necessário o esforço conjunto de todos os integrantes da equipe responsáveis pelo sistema a ser aposentado. |
| Ferramentas | Softwares de vídeo-conferência como Skype, hangout ou curse voice. |

8) Templates dos Artefatos

Documento de análise de viabilidade e

riscos: <https://github.com/gabrielaimeeg/DroidMetronome/wiki/TEMPLATE-Documento-de->

Análise-de-Viabilidade-e-Riscos

Documento de controle e

correção: <https://github.com/gabrielaimeeg/DroidMetronome/wiki/TEMPLATE-Documento-de-Control-e-Correção>

Documento de migração: <https://github.com/gabrielaimeeg/DroidMetronome/wiki/TEMPLATE-Documento-de-Migração>

Documento de solicitação de

mudança: <https://github.com/gabrielaimeeg/DroidMetronome/wiki/TEMPLATE-Documento-de-Solicitação-de-Mudança>