**Antonio L. Lopes Uliana 62834**

**Daniel L. M. de Moura 743525**

**João Vitor A. Marciano 743554**

**Juliana K. de Sousa 594997**

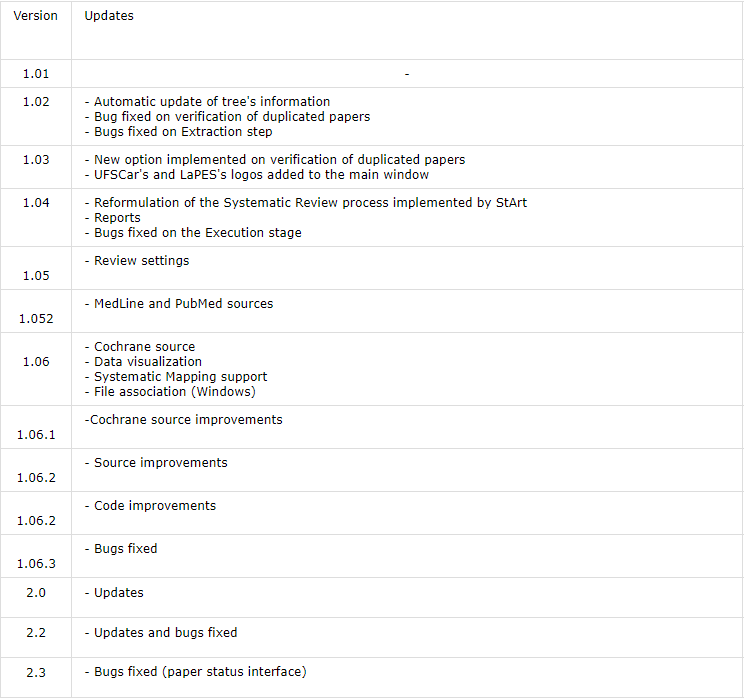
**Matheus Kirstus 628310**

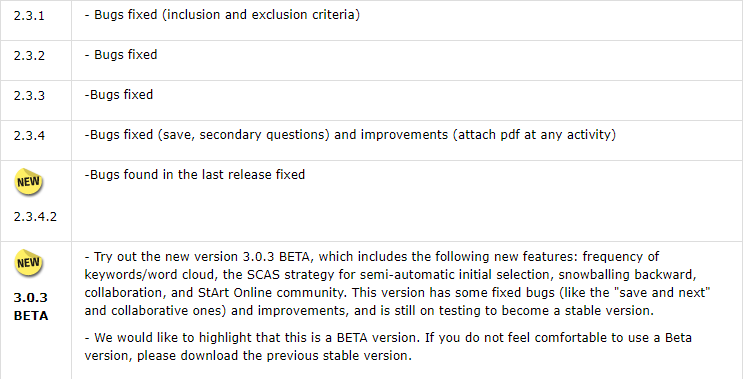
**Plano de Gestão de Configuração da Ferramenta StArt**

1. **Aprovações/Participantes**

Membros do Lapes - Laboratório de Pesquisa em Engenharia de Software.

1. **Histórico de Revisões**

****

****

1. **Introdu ção**

Este Plano de Gestão de Configurações de Software descreve como os artefatos da Ferramenta StArt são gerenciadas.

* 1. **Definições**

A Revisão Sistemática (RS) é uma técnica utilizada para buscar evidências na literatura científica que seja conduzida de maneira formal, aplicando-se etapas bem definidas, conforme protocolo previamente elaborado. Como a RS tem muitas etapas e atividades, sua execução é trabalhosa e repetitiva. Portanto, o suporte de uma ferramenta computacional é essencial para melhorar a qualidade de sua aplicação. Para tanto, foi desenvolvida uma ferramenta denominada StArt (State of the Art, através de Revisão Sistemática), que visa auxiliar o pesquisador, dando suporte à aplicação desta técnica. A ferramenta StArt está sendo usada por estudantes de pós-graduação que declararam seu apoio positivo e suas vantagens em relação a outras ferramentas.

* 1. **Acrônimos**

Lapes - Laboratório de Pesquisa em Engenharia de Software.

IC - Item de Configuração

1. **Gerenciamento**
   1. **Organização**
   2. **Responsabilidades**
      1. **Líder de Configuração**
      2. **Líder de Projeto**
      3. **Engenheiros**
   3. **Políticas Aplicáveis, Diretivas e Procedimentos**
2. O Gerenciamento de Configuração para este projeto devem estar de acordo com as normas estabelecidas pelas: IEEE-828-2012-2005, IEEE-828-2012, IEEE-1028-2008 e IEEE-12207-2017.
3. De acordo com as políticas de modularização de desenvolvimento do Software, ele possui, atualmente, três macro divisões de desenvolvimento, além de updates de correções de bugs.
4. Uma versão BETA está disponível para download, com os seguintes novos recursos: frequência de palavras-chave ou nuvem de palavras; estratégia SCAS para seleção inicial semiautomática; backball; colaboração e a comunidade StArt Online
5. **Atividades**
   1. **Identificação de Configuração**
      1. **Identificando os itens de Configuração**

O líder do projeto é responsável por identificar todos os itens de configuração. Os desenvolvedores que desejarem propor algum item deverão sugerir ao líder.

* + 1. **Nomeando os itens de Configuração**

Por se tratar de um Software de desenvolvimento constante, os itens de configuração a serem desenvolvidos estão relacionados à equipe de desenvolvimento vigente. Essa equipe é formada tanto por alunos, como professores do LAPES.

As versões são armazenadas em um master numa ferramenta de gerenciamento de versão, de modo que os desenvolvedores com acesso ao master podem requisitar branches, fazerem as alterações de código necessárias e depois efetuarem o merge para o repositório, no master. Existem, basicamente, dois níveis de atualização do Software: alteração nos itens de configuração (são mudanças relevantes que alteram a versão atual do software - como versão 1.0, 2.0 e 3.0) e alterações de correção de bugs e melhoria de funcionalidades (mudanças dentro das próprias versão, por exemplo, 1.1, 1.2, 1.3 etc)

* + 1. **Adquirindo itens de Configuração**

O processo de desenvolvimento deste software precisa, necessariamente, entender e consultar quais as bases onde serão consultados os materiais de artigos científicos. Após isso, é necessário entender como acessar a base das ferramentas a serem consultadas, de forma a se construir um programa que seja capaz de identificar a base consultada, e realizar uma busca atendendo aos seus critérios de consulta.

* 1. **Controle de Configuração**

Esta seção explicita o processo pelo qual os itens de configuração são alterados. Esse processo deve ser flexível o suficiente para permitir mudanças rápidas, mas controlado o suficiente para manter as mudanças muito ordenadas, de modo que elas melhorem a aplicação, sem danificá-la.

* + 1. **Requisição de Mudanças**

Conforme especificado no Plano de Gerenciamento de Projetos de Software (item 5.1.1), a equipe designará um líder que é responsável pelo gerenciamento do projeto. Antes de solicitar uma mudança, os desenvolvedores devem obter uma inspeção da mudança proposta de uma equipe de inspeção ou, se isso não for possível, de seu gerente. Para solicitar a incorporação de um item de configuração alterado no master, o desenvolvedor deve realizar uma solicitação de branch.

* + 1. **Avaliação de Mudanças**

A avaliação de mudanças é realizada pelos desenvolvedores responsáveis por partes do software direta ou indiretamente afetadas por ela, em conjunto com o gerente, de modo a analisar e aprovar a mudança, ou reprová-la.

* + 1. **Implementação de Mudanças**

Uma vez que um item de configuração seja aprovado para incorporação no merge, o gerente será responsável por coordenar o teste e a integração do IC alterado. Isso deve ser executado de acordo com a documentação do teste de regressão descrita na seção “Documentação de teste de software”. Em particular, o gerente deve alocar desenvolvedores, responsáveis pelos testes realizados no software, para realizarem o desenvolvimento de uma versão para teste.

* 1. **Contabilização do Status de Configuração**

Toda e qualquer atualização pertinente ao desenvolvimento do software que seja alterado no merge deve ser documentado e registrado pelo gerente do projeto, de modo a se manter um histórico de atualizações realizadas, assim como os motivos das mudanças. Deste modo, a consulta por futuros desenvolvedores se torna mais fácil e coerente. Também deve-se realizar backups de versões anteriores, com o intuito de minimizar possíveis danos causados ao software com alterações de versões.

* 1. **Controle de Interface**

Existem três interfaces relacionadas ao software: duas indiretas (site da página do Lapes onde constam as versões e descrição do software e página da comunidade de interessados na ferramenta) e uma direta (é o próprio software). Todas elas podem ser encontradas disponíveis em: http://lapes.dc.ufscar.br/tools/start\_tool.

1. **Cronograma**

Não é possível se obter acesso ao cronograma exato de desenvolvimento do software, uma vez que ele foi desenvolvidos por diferentes equipes, ao longo dos anos de 2009/2010 até 2013, de onde se tem informação através da página da ferramenta.

1. **Recursos**

O software foi desenvolvido por alunos (de graduação ou pós-graduação) e professores do Lapes.

1. **Planejamento de Manutenção**

A manutenção foi realizada de modo a priorizar correção de bugs e funcionalidades “vitais” do funcionamento do software.