Procedimentos para a distribuição de produtos da DIMNT

v1.0.1

Garantia de Qualidade para o usuário final e manifestos de adequação à operacionalização

Objetivo

Definir como garantir a qualidade dos produtos entregues da Divisão de Modelagem Numérica do Sistema Terrestre (DIMNT) e como devem ser disponibilizados, de forma que estes estejam organizados para uso pela própria DIMNT e sejam acessíveis por usuários finais e pela Divisão de Previsão de Tempo e Clima (DIPTC), bem como a documentação que descreve as funcionalidades dos seus modelos e produtos.

1. Introdução

Um produto (ex. Modelo Numérico) a ser entregue pela DIMNT deve ter sua qualidade garantida através da gestão do ciclo de vida do produto, isto é, o controle total das modificações ocorridas ao longo da sua existência.

Os produtos da DIMNT são entregues a usuários finais e também à DIPTC, para que sejam implantados e operacionalizados. Esse documento visa definir os requisitos mínimos para a distribuição do produto para estes.

A operacionalização do produto entregue à DIPTC requer necessidades específicas, que são tratadas em um documento à parte, denominado **Procedimentos para operacionalização de processos**. Logo, além dos procedimentos aqui definidos, os produtos entregues à DIPTC também deverão seguir as regras definidas por aquela divisão.

Um sistema de **Controle de Versões** (ou versionamento) tem a finalidade de gerenciar diferentes versões no desenvolvimento de um documento qualquer. No caso da DIMNT, entende-se que todos os modelos/softwares desenvolvidos devem seguir um protocolo de versionamento a fim de permitir que o histórico dos desenvolvimentos, bem como a sua documentação, sejam acessíveis e disponíveis a todos os interessados.

Para uma melhor compreensão deste documento, é necessário entender que o modelo/software não é representado apenas por um programa, mas também por dados, documentação e arquivos de configuração associados. Todos estes elementos são necessários para que o modelo opere de forma completa e correta. Um sistema de software consiste, geralmente, de um conjunto de programas separados; arquivos de configuração, que são utilizados para configurar esses programas; documentação do sistema, que descreve a sua estrutura; a documentação do usuário, que explica como o sistema deve ser utilizado; e outras fontes de informação, como sites web por meio dos quais os usuários podem obter as informações atualizadas.

Diferentes institutos de pesquisas públicos e privados, fazem uso de diferentes ferramentas que os auxiliam no processo de controle de versões, dos quais, entre os softwares livres, pode-se citar o CVS (Concurrent Version System), Mercurial, Git e SVN (Subversion). Na linha dos softwares comerciais, pode-se citar o SourceSafe, TFS, PVCS (Serena) e o ClearCase. Embora tais softwares sejam úteis e necessários para o bom controle de versões, este documento versa sobre o protocolo que deverá ser adotado no âmbito da DIMNT para a padronização do desenvolvimento. Em um primeiro momento, não serão levadas em consideração as vantagens e/ou desvantagens de cada um destes softwares.

A confecção deste documento irá basear-se nas necessidades e características específicas dos diferentes modelos em uso e em desenvolvimento pelo INPE no âmbito da DIMNT, no que diz respeito à transição para diferentes ambientes, configuração, documentação e distribuição, considerando seus usuários internos e externos.

2. Conjunto básico de informações para entrega de produtos da DIMNT

Abaixo são listados os itens básicos requeridos para a entrega de produtos desenvolvidos e mantidos pela DIMNT.

1. **Código Fonte**: Deve ser disponibilizado o código fonte do modelo/software desenvolvido e mantido pela DIMNT com todas as alterações realizadas desde a última versão, incluindo os seguintes itens:
    a. ***Change log*:** Este é um documento texto, entregue junto com o produto (modelo/software) e contém todas as alterações realizadas desde a última versão disponibilizada;
    b. ***README***: Formato: *Markdown*. Este é um documento texto, entregue junto com o produto e contém informações básicas relacionadas à compilação e execução do modelo/software;
2. **Documentação:** Detalhamento no Capítulo 4. Documentação.
3. ***Test case***: Este é um conjunto básico de dados e a configuração padrão do modelo/software a ser utilizada pelo usuário para reproduzir um ciclo de simulação, no caso dos modelos, para um período de integração de 6h a 72h. Neste conjunto deve-se disponibilizar também os resultados que deverão ser encontrados pelo usuário ao final da simulação. Detalhamento no item 5.1 Test cases.
    NOTA: No caso de produto que não seja modelo numérico, deve-se disponibilizar o conjunto mínimo de informações para que o usuário seja capaz de executar o software de forma funcional e que o resultado seja compreensível.

Nas seções seguintes serão apresentados os roteiros básicos para que cada item seja cumprido. O código entregue pelo grupo de desenvolvedores da DIMNT deverá seguir um sistema de versionamento semântico como descrito brevemente na seção seguinte.

3. Versionamento de Software

Os softwares/modelos desenvolvidos pela DIMNT devem seguir o sistema de versionamento semântico em que um número de versão normal deve ter o formato de X.Y.Z, onde X, Y, e Z são inteiros não negativos, e não devem conter zeros à esquerda. Neste esquema X é a versão Maior, Y é a versão Menor, e Z é a versão de correção/ajuste fino. Cada elemento deve ser incrementado numericamente. Por exemplo: 1.9.0 < 1.10.0 < 1.11.0 < 2.0.1 e assim por diante. É importante salientar que, uma vez que um pacote versionado foi lançado (entregue), o conteúdo desta versão não deve ser modificado. Qualquer modificação feita deve ser lançada como uma nova versão.
Desta forma um software/produto/modelo desenvolvido e mantido pela DIMNT deverá ser entregue da seguinte forma[1]:

- **BAM_v1.4.0**
- **BESM_v4.2.0**

[1]As versões dos softwares/produtos/modelos apresentadas são fictícias e não correspondem necessariamente à versão desenvolvida.

- **BRAMS_v5.11.1**
- **Eta_v9.0.1**
- **GEF_v3.9.5**
- **SCANTEC_v2.0.3**
- **SMG_v2.1.10**
- **WRF_v1.0.0**

A determinação dos números que formarão uma versão (vX.Y.Z), dependerá do que está sendo desenvolvido. Se o modelo/produto possuir um planejamento, as características e funções planejadas, podem ser convenientemente distribuídas entre diferentes versões a serem lançadas conforme um calendário pré-estabelecido. De forma geral, a determinação destes números deve atender aos critérios descritos a seguir.

Versão Maior (X)

Representa uma grande modificação/inovação

Considerando o modelo/produto fictício de versão v1.2.3, o primeiro número, neste caso o número 1, representa a primeira versão desenvolvida e consolidada. Esta primeira versão, geralmente, representa um milestone no desenvolvimento, um marco de progresso. Por exemplo, a primeira versão pública do modelo/produto, marcando uma inovação ou uma grande contribuição para o seu estabelecimento.

Versão Menor (Y)

Representa um aperfeiçoamento ou a inclusão de funcionalidades

Na versão v1.2.3 do modelo/produto fictício, mostra que 2 representa uma segunda geração/iteração de desenvolvimentos dentro da primeira versão. Ela pode representar, por exemplo, a inclusão de uma nova parametrização física ou um novo formato de saída dos arquivos pós-processados. Além disso, a versão menor pode representar uma evolução dentro das características planejadas da primeira grande versão, sem que necessariamente, tenha-se alterado as características básicas e principais do modelo/produto.

Versão de Correção (Z)

Representam alterações de baixo impacto

O último número da versão v1.2.3 do modelo/produto fictício, representa uma revisão das características e funções já estabelecidas e marcadas pelos números que representam as versões maior (X) e menor (Y). Por exemplo, uma correção de bug, ajustes em scripts etc. As revisões determinadas pelos números que representam as versões de correção (Z) não são determinadas por grandes alterações de estrutura e funcionamento do modelo/produto.

4. Documentação

Para que um projeto de software seja iniciado, é essencial que a sua documentação seja criada também. A fim de que o sistema seja desenvolvido com um menor prazo, custo operacional e maior confiabilidade, deve-se elaborar uma documentação mínima, que servirá como direção para o desenvolvimento, homologação e implantação do sistema. Três tipos de documentações deverão ser enviadas:

1. Documentação do Usuário (manual de uso);
2. Documentação do Desenvolvedor (documentação do código escrita no corpo do código);
3. Documentação Científica (referente aos processos físicos representados).

A Documentação do usuários (manual de uso) deve contemplar como realizar:

- a compilação;
- a instalação;
- a execução do(s) *test case*(s) e verificação de sucesso ou falha.

A Documentação do desenvolvedor deve descrever os componentes do sistema, as funcionalidades das rotinas implementadas, bem como os parâmetros de entrada e saída, e o histórico de alterações das rotinas. No documento **Modelo de Desenvolvimento de Software** foi definido um padrão para a confecção desta documentação. Neste momento ainda não é exigida que a documentação siga esse padrão, mas é desejável que os próximos desenvolvimentos a sigam.

A Documentação Científica pode ser escrita separadamente no formato desejado, com o propósito de publicação em revistas ou jornais científicos.

5. Validação

Para que um software possa ser operacionalizado, deve haver um conjunto de dados disponíveis para verificar se o que foi compilado e instalado está correto. Este conjunto de dados, denominado test case, deve permitir também que o usuário possa reproduzir os resultados e o comportamento esperados do modelo/produto.
Novas versões do modelo/produto devem ser apresentadas com melhorias em relação à versão anterior, sejam relacionadas com o desempenho do software ou na qualidade das previsões numéricas dos modelos operacionais. Da mesma forma, o conjunto de dados fornecido, deve sempre produzir os resultados esperados, evidenciando as melhorias e características da nova versão do modelo/produto.

5.1 *Test cases*

Deverá ser disponibilizado um test case mínimo, para validar a funcionalidade básica do modelo ou produto, de forma que o usuário seja capaz de compilar, executar e validar os resultados do modelo/produto de forma independente. O pacote de arquivos do test case deve conter, essencialmente, o *name list* (configurações do tempo de execução do modelo/produto), os arquivos de entrada e os arquivos de saída, para que a validação da instalação e das características principais do modelo/produto possam ser realizadas.

O tamanho total do pacote de arquivos deve ser mínimo, de forma que este possa ser disponibilizado em páginas web ou ftp. No caso de modelos, o *test case* deve contemplar um período curto de uma simulação (eg., 6 a 24 horas) e uma configuração que não demande recursos computacionais que excedam a capacidade de processamento da máquina de um usuário comum. Por exemplo, um *name list* que configure uma grade de baixa resolução, ou apenas um ciclo de assimilação de dados, ou mesmo as primeiras horas de previsão a partir dos membros de um conjunto de condições iniciais.

Test cases “mais completos” também podem ser disponibilizados. Entenda-se “mais completos” por testes que utilizem configurações e dados que reproduzam testes mais robustos e próximos das configurações operacionais. Por exemplo, a utilização de um maior período de previsões e maior resolução, no caso de modelos numéricos.

6. Distribuição de pacotes para usuários finais e à DIPTC

Os produtos enviados à DIPTC deverão seguir uma série de Requisitos e Procedimentos para a entrada de um novo produto da DIMNT em operação, tais como as características do novo produto e a comprovada necessidade de tal processo como operacional, demonstrando a utilidade para o INPE, sociedade ou parceiros além da concordância com o plano diretor do INPE. Também são exigidas documentações básicas, controle de versão (já tratado aqui), padrões de scripts, diretórios configuráveis, etc. Os procedimentos estão detalhados no documento **Procedimentos para operacionalização de processos**.

6.1 Pacote final

Os pacotes deverão ser distribuídos na área de Transição de Modelos, a ser definida. Esta área deverá ser configurada pela equipe de gerenciamento dos discos. O tamanho da área será estimado com base nas informações dos pacotes de cada modelo/produto. Este pacote deve conter os seguintes itens, que devem ser disponibilizados como arquivos comprimidos.

- **Executável(is):** arquivo(s) binário(s) do programa principal e/ou auxiliares;
- **Códigos Fonte:** arquivos com os fontes do programa principal e/ou módulos e bibliotecas auxiliares;
    - O código fonte deverá ser acompanhado do arquivo README, que deve conter a descrição das configurações de ambientes, tais como as versões do compilador e bibliotecas utilizadas (inclusive as carregadas por módulos). Deve-se especificar qual o *Makefile* utilizado e o *checksum* md5 do pacote com os códigos fonte;
- **Datafix:** conjunto de arquivos de entrada e arquivos fixos (sem data, ie., que não são atualizados na execução do modelo/produto) válidos para a versão entregue;
- ***Test case*:** conjunto de arquivos de entrada e saída, para validação de uma rodada operacional, bem como os *name lists* e outros arquivos específicos de configuração do modelo/produto.

Todos os itens acima são desejáveis para usuários finais, incluindo o código. A DIPCT não exige a entrega de código, uma vez que este não deverá ser manipulado por aquela divisão.

Estrutura de diretórios na área comum de transição:

```
raiz
  L modelname
    L exec
      L [modelname]_exec_[domain]_[resolution]_v1.2.3.xz
      L modelname_exec_domain_resolution_v1.2.4.xz
    L src
      L [modelname]_[domain]_[resolution]_v1.2.3.xz
      L [modelname]_v1.2.4_domain_resolution_.xz
    L datafix
      L [modelname]_datafix_[domain]_[resolution]_v1.2.3.xz
      L [modelname]_datafix_[domain]_[resolution]_v1.2.4.xz
    L testcase
      L [modelname]_testcase_[domain]_[resolution]_v1.2.3.xz
      L [modelname]_testcase_[domain]_[resolution]_v1.2.4.xz
```

Exemplo:

```
raiz
  L eta
    L exec
      L eta_exec_AS_5km_v1.2.3.xz
      L eta_exec_AS_5km_v1.2.4.xz
      L eta_exec_Angra_1km_v1.2.3.xz
      L eta_exec_Angra_1km_v1.2.4.xz
    L src
      L eta_src_AS_5km_v1.2.3.xz
      L eta_src_AS_5km_v1.2.4.xz
      L eta_src_Angra_1km_v1.2.3.xz
      L eta_src_Angra_1km_v1.2.4.xz
    L datafix
      L eta_datafix_AS_5km_v1.2.3.xz
      L eta_datafix_AS_5km_v1.2.4.xz
      L eta_datafix_Angra_1km_v1.2.3.xz
      L eta_datafix_Angra_1km_v1.2.4.xz
    L testcase
      L eta_testcase_AS_5km_v1.2.3.xz
      L eta_testcase_AS_5km_v1.2.4.xz
```

```
L eta_testcase_Angra_1km_v1.2.3.xz
L eta_testcase_Angra_1km_v1.2.4.xz
```

Nesta estrutura de diretórios, será priorizado o nome dos modelos (diretório 'modelname'), dentro dos quais, serão armazenados e organizados os subdiretórios 'src', 'exec', 'datafix' e 'testcase'. Dentro destes subdiretórios, serão armazenados os elementos integrantes dos pacotes, organizados conforme o domínio, a resolução e a versão, comprimidos no formato '.xz'.

Alguns modelos têm *test cases* ou até mesmo códigos específicos (e consequentemente executáveis) para atender diferentes domínios (*domain*) e resoluções (*resolution*), mas a maioria dos modelos não devem ter códigos diferentes para uma mesma versão. Alguns modelos também podem ter mais de um *test case* com diferentes domínios e resoluções, mas ter apenas um código por versão.

O formato de compressão '.xz' é sugerido por ser mais eficiente na compressão de arquivos binários. No tupã não se tem essa opção.

Máquinas com o xz instalado (ex. XC-50)

Para utilizar o formato de compressão '.xz' com fator de compressão máxima, pode-se utilizar a seguinte linha de comando:

```
$ XZ_OPT=-9e tar cJf diretorio.tar.xz diretorio
```

Para descomprimir o arquivo '.tar.xz' criado, pode-se utilizar o seguinte comando:

```
$ xz -d diretorio.tar.xz
```

e em seguida:

```
$ tar -xvf diretorio.tar
```

Máquinas sem o xz instalado (ex. Tupã)

Para utilizar o formato de compressão '.tar.gz' com fator de compressão máxima, pode-se utilizar a seguinte linha de comando:

```
$ GZIP=-9 tar czf diretorio.tar.gz diretorio
```

Para descomprimir o arquivo '.tar.gz' criado, pode-se utilizar o seguinte comando:

```
$ tar xzf diretorio.tar.gz diretorio
```

A partir dos arquivos comprimidos nos formato '.xz', ou ‘tar.gz’ pode-se criar os arquivos '.md5sum'. Para isso, basta utilizar o seguinte comando:

```
$ md5sum diretorio.tar.xz > diretorio.md5sum
```

ou

```
$ md5sum diretorio.tar.gz > diretorio.md5sum
```

A verificação dos arquivos *.md5sum pode ser feita com o comando:

```
$ md5sum -c diretorio.md5sum
```

6.2 Pacotes dos modelos atuais

A tabela abaixo apresenta os tamanhos estimados dos pacotes dos modelos atuais, em MB, enviados até o momento.

Modelo	**Executáveis**	**Fontes**	**Dados Fixos**	**Test Case**	**Total**
BAM	258,00	106,00	3.523,00	24600	28.487,00
BESM	18,00	9,10	12,00	184,00	223,10
BRAMS	8,00	20,00	6.056,00	3.206,00	9.290,00
Ensemble	8,30	0,27	202,00	2.800,00	3.010,57
ETA	?	?	?	?	9.290,00
Pre do BAM	3,30	0,53	208,00	12.000,00	12.211,83
SMG	56,40	6,10	60,82	3.356,00	3.479,32
WRF	61,00	77,00	4.542,00	570,00	5.250,00
WW3	134,00	3,1	6,1	5.516,00	5.659,20
TOTAL	**289,00**	**116,10**	**11.086,92**	**27.632,00**	**76.901,02**

6.3 Área de Transição de modelos

Considerando o tamanho total dos modelos atuais (TOTAL da tabela do item anterior) e que a área a ser criada contemplará 3 versões de cada modelo, teríamos aproximados 77 GB x 3, totalizando 231 GB. Adicionando possíveis alterações e inclusões de dados nos modelos, o tamanho total da área foi definido com 500GB.

A área de transição está localizada no Netapp, onde serão guardados os arquivos disponibilizados para a operação. Poderá ser acessada através de:

1. Dentro da rede do CPTEC: tupa.cptec.inpe.br://share/transicao
2. Fora da rede do CPTEC: ftp.cptec.inpe.br/pesquisa/transicao

7. Considerações finais e próximos passos

O presente trabalho definiu os procedimentos para a entrega de modelos (documentação e arquivos exigidos) da DIMNT para usuários finais e à DIPTC, a fim de padronizar as versões dos modelos, armazenar as últimas três versões dos modelos e

protocolar as entregas, garantindo a qualidade mínima do produto e o gerenciamento dos arquivos entregues. A DIPTC têm as suas próprias necessidades, que devem ser seguidas adicionalmente às definições deste documento, no documento intitulado **Procedimentos para operacionalização de processos**.

À medida que forem realizadas as primeiras entregas, dúvidas e necessidades de alterações no processo surgirão, amadurecendo o processo. Estas alterações deverão passar por uma série de discussões, para só assim serem inseridas em novas versões deste trabalho.

O processo de entrega, isto é, para quem deverão ser entregues os arquivos dos modelos e suas respectivas documentações, será definido e comunicado a todos da DIMNT.