

Diego S. Cintra, Jainor Souza

# **Engenharia de requisitos do MPS.BR**

Campo Grande, Mato Grosso do Sul  
2015

# Resumo

Foi-nos requisitado a descrição a respeito dos processos de engenharia de requisitos contidos no MPS.BR, tendo como suporte o Guia Geral MPS de Software, que define um modelo de qualidade de processos para médias empresas de software no Brasil. Para cada um desses processos de engenharia associados, exemplos também foram solicitados. Essas requisições visam ampliar nossos conhecimentos ao respeito da inserção da engenharia de requisitos em um modelo de qualidade popularmente conhecido.

**Palavras-chaves:** Engenharia, MPS.BR, requisitos

# Lista de abreviaturas e siglas

UFMS      Universidade Federal de Mato Grosso do Sul

FACOM      Faculdade de Computação

# Sumário

	<b>Sumário . . . . .</b>	<b>3</b>
	<b>Introdução . . . . .</b>	<b>4</b>
<b>1</b>	<b>NÍVEL G - PARCIALMENTE GERENCIADO . . . . .</b>	<b>5</b>
<b>1.1</b>	<b>Gerência de Requisitos - GRE . . . . .</b>	<b>6</b>
1.1.1	GRE 1 . . . . .	7
1.1.2	GRE 2 . . . . .	8
1.1.3	GRE 3 . . . . .	9
1.1.4	GRE 4 . . . . .	9
1.1.5	GRE 5 . . . . .	10
<b>2</b>	<b>NÍVEL D - LARGAMENTE DEFINIDO . . . . .</b>	<b>11</b>
<b>2.1</b>	<b>Desenvolvimento de Requisitos - DRE . . . . .</b>	<b>14</b>
2.1.1	DRE 1 . . . . .	14
2.1.2	DRE 2 . . . . .	15
2.1.3	DRE 3 . . . . .	15
2.1.4	DRE 4 . . . . .	16
2.1.5	DRE 5 . . . . .	17
2.1.6	DRE 6 . . . . .	17
2.1.7	DRE 7 . . . . .	18
	<b>Conclusão . . . . .</b>	<b>19</b>

# Introdução

O MPS.BR (Modelo de Processo de Software) é um modelo de qualidade de processos introduzido no país em volta dos anos de 2003-2006, visto que, em comparação com outros países em desenvolvimento (como a Índia), a maioria das empresas brasileiras não tinha um alto (ou suficiente) nível de maturidade que demonstrasse capacidade para competição global.

Sete níveis de maturidade foram definidos para tal modelo, sendo eles classificados - ou seja, a passagem de um nível para outro demonstra um crescimento e maturidade da empresa em relação ao que ela era antes. Dado isso, temos, em ordem crescente, os seguintes níveis: Parcialmente Gerenciado, Gerenciado, Parcialmente Definido, Largamente Definido, Definido, Gerenciado Quantitativamente e Em Otimização.

Dentre esses níveis, temos processos a serem cumpridos em cada um deles. Do escopo de engenharia de requisitos, existem dois processos, definidos nos níveis G e D, que condizem com essa área. Cada um desses processos também espera por resultados e, ao longo desse documento, iremos explicitar, com o auxílio de exemplos, quais são essas saídas esperadas.

# 1 Nível G - Parcialmente Gerenciado

O Nível G da Melhoria de Processo de *Software* Brasileiro é composto de dois processos:

1. Gerência de Projetos e
2. Gerência de Requisitos.

Estes processos devem atender aos Atributos de Processo(AP) 1.1 e 2.1, cada qual com seus Resultados Esperados de Atributo de Processo(RAP). São eles:

O Atributo de Processo 1.1 apenas visa regulamentar o cumprimento dos propósitos do processo, explicitando quando ele cumpre a que se propõe e quando ele desvia de sua premissa. Seus RAPs são:

Resultados Esperados	Definição
RAP 1	O processo atinge seus resultados definidos

Tabela 1 – RAP do AP 1.1

O Atributo de Processo 2.1 visa regulamentar a gerência do processo, ou seja, explicitar o quanto sua execução é gerenciada. Seus RAPs são(exclusivamente voltados para o Nível G):

Resultados Esperados	Definição
RAP 2	Existe uma política organizacional estabelecida e mantida para o processo;
RAP 3	A execução do processo é planejada;
RAP 4	A execução do processo é monitorada e ajustes são realizados;
RAP 5	As informações e os recursos necessários para a execução do processo são identificados e disponibilizados;
RAP 6	As responsabilidades e a autoridade para executar o processo são definidas, atribuídas e comunicadas;
RAP 7	As pessoas que executam o processo são competentes em termos de formação, treinamento e experiência;
RAP 8	A comunicação entre as partes interessadas no processo é planejada e executada de forma a garantir o seu envolvimento;
RAP 9	Os resultados do processo são revistos com a gerência de alto nível para fornecer visibilidade sobre a sua situação na organização;
RAP 10	O processo planejado para o projeto é executado.

Tabela 2 – RAPs do AP 2.1

Como a parte de Gerência de Projetos do Nível G não é o tema abordado nesta matéria, discutiremos apenas sobre o processo de Gerência de Requisitos.

## 1.1 Gerência de Requisitos - GRE

A Gerência dos Requisitos de um projeto tem como objetivo identificar inconsistências entre os requisitos, os planos do projeto e os produtos e artefatos criados para o projeto, bem como manipular mudanças a serem feitas em requisitos de forma íntegra e integrada com o resto do projeto.

São cinco os resultados esperados da Gerência de Requisitos de um projeto:

- GRE 1. O entendimento dos requisitos é obtido junto aos fornecedores de requisitos;
- GRE 2. Os requisitos são avaliados com base em critérios objetivos e um comprometimento da equipe técnica com estes requisitos é obtido;
- GRE 3. A rastreabilidade bidirecional entre os requisitos e os produtos de trabalho é estabelecida e mantida;
- GRE 4. Revisões em planos e produtos de trabalho do projeto são realizadas visando identificar e corrigir inconsistências em relação aos requisitos;
- GRE 5. Mudanças nos requisitos são gerenciadas ao longo do projeto.

### 1.1.1 GRE 1

Uma boa definição das partes interessadas-os *stakeholders*- é imprescindível para que o GRE 1 seja cumprido. Saber quem tem interesses no projeto, quem usufruirá dele e quem fornecerá recursos/ajudará na sua realização são questões que necessitam não possuir sequer um rastro de dúvida ou sombreamento. Todos os *stakeholders* possuem requisitos específicos ou gerais que devem ser devidamente elicitados, analisados, avaliados e gerenciados para que o andamento do projeto cumpra o cronograma estipulado.

Por mais que a equipe de desenvolvimento tenha conhecimento do negócio da organização que a contratou, não há dúvidas que existem quesitos específicos de negócio que a equipe não conhece, como regras de negócio e padrões que mudam de empresa para empresa. Por este motivo, um engajamento de ambas as partes é fundamental para que os requisitos sejam bem entendidos e aplicados ao projeto.

Um exemplo básico são os infames requisitos implícitos. Em entrevistas com *stakeholders*, é comum que, por conhecer seu negócio rotineiramente, eles omitam certas informações que fazem a completa diferença no momento de implementar o requisito elicitado. Por exemplo, numa empresa de locação de veículos, pode ser



que, ao procurar informações sobre a reserva de um veículo, a organização contratada entrevistou o gerente funcional que cuida desta área e, em dado momento da entrevista, a pergunta "Quando um cliente confirma uma reserva, o veículo recebe o *status* de reservado?" é feita, a qual o cliente responde com "Sim". No entanto, na empresa de locação, o prazo máximo de reservas sempre foi de 24h e o cliente não fez esta observação, por ser uma prática adotada há muitos anos.

### 1.1.2 GRE 2

Requisitos não avaliados ou avaliados de forma subjetiva mostram, na maioria das vezes, ser um grande problema para o andamento do projeto. Boas práticas de modelagem, revisão, verificação e validação de requisitos são muito bem-vindas, e isto quer dizer, os critérios utilizados devem ser muito bem definidos para que não restem dúvidas quanto às necessidades do cliente ao apontar um requisito do projeto. Técnicas e métricas de qualidade de projetos também são interessantes de serem aplicadas no GRE 2, pois é parte da garantia da qualidade avaliar, com uma visão externa e imparcial, o produto em desenvolvimento e do controle da qualidade revisar, agora com uma visão centrada na equipe e nas partes interessadas, o que foi produzido.

Uma vez que a avaliação seja realizada, a equipe finalmente se encontra pronta para desenvolver o que lhe foi estipulado, dados os processos que ela adota. Uma boa avaliação define papéis e atividades, que permite que a estrutura do projeto seja bem distribuída e controlada, de acordo com a tecnologia, o conhecimento, e os recursos alocados para cada indivíduo que faz parte da equipe.

Um exemplo pode ser pensar nos recursos de tempo e custo que estão disponíveis para um projeto. Nos critérios de avaliação, se constarem palavras subjetivas como "muito" ou "pouco", pode ser que a equipe tenha surpresas desagradáveis no andar de sua execução. Por exemplo, pode ser que "pouco" dinheiro seja mais do que há disponível para usar!

### 1.1.3 GRE 3

Os requisitos refletem diretamente no comportamento de um projeto e, em seu andar, vários artefatos são produzidos, desde diagramas conceituais a protótipos funcionais, culminando no produto final. Se os requisitos foram bem elicitados e avaliados, é possível mapeá-los para funcionalidades do sistema produzido e vice-versa. Isto se dá pela clareza e pela objetividade dos requisitos, que são características necessárias ao desenvolver um projeto.

Como exemplo, num trabalho de Banco de Dados sobre uma locadora de veículos, era possível analisar o documento de requisitos do sistema, imaginar como seria uma funcionalidade(sem ter o visto ou desenvolvido ainda), e, ao analisar o produto, podíamos facilmente mapear decisões de *design* e funcionalidades para os requisitos do sistema. Uma mão dupla que é muito necessária, principalmente no desenvolvimento de *software*.

### 1.1.4 GRE 4

O quarto resultado esperado da gerência de requisitos usa uma mentalidade autocrítica. Se uma empresa não pretende identificar e corrigir inconsistências em seus planos e produtos com relação aos requisitos, não há a necessidade de revisá-los. Seria só um gasto de recursos desnecessário e é esta a questão em pauta aqui. Além disto, obviamente, errar é uma característica humana e, por isto, revisões são altamente necessárias. Há a necessidade de não ser arrogante e, às vezes, revisar até mesmo processos de negócio envolvidos durante o andamento de um projeto.

Pela facilidade que existem em tomar atalhos na execução de projetos, revisar o caminho usado e o que foi criado a partir desta decisão devem ser feitas o quanto antes, pois o custo de se corrigir possíveis erros em estágios avançados de um projeto aumenta drasticamente. Um exemplo no desenvolvimento de *software* é aproveitar código, uma prática bastante recorrente. Por mais que sistemas sejam muito semelhantes, usar um mesmo motor para os dois pode levar a inconsistências de requisitos(observadas mais facilmente nos requisitos não funcionais).

### 1.1.5 GRE 5

Um projeto tem a característica de estar sempre em mudança. Esta faceta é oriunda exatamente dos requisitos, que sempre estão mudando. Obviamente, a mudança em si é um requisito do projeto, que tem de ser tratado minuciosamente para que a harmonia não seja quebrada. Por isto o controle integrado de mudanças é uma boa prática de gerência de projetos(segundo o PMBOK), e não seria diferente com os requisitos, uma vez que são uma decomposição das necessidades de um projeto.

Se um requisito muda, os processos de analisar, especificar e avaliar devem ser aplicados novamente, para que esta mudança seja bem analisada e para que se tenha total certeza de sua viabilidade. Estas mudanças devem ser geridas devidamente para que os recursos não se acabem e para que as necessidades do *stakeholder* sejam atendidas, lembrando que a mudança em si é uma necessidade de um projeto. Se, durante o desenvolvimento de um sistema para a locação de um veículo, a empresa contratante mudar o processo de negócio que define como o aluguel de um veículo é feito, há uma grande chance que ela peça para que a equipe de desenvolvimento mude como o sistema trata este tipo de transação. Se esta mudança não for gerida, o produto será datado e não atenderá às novas necessidades da empresa, o que acarretará em problemas(grandes!) entre as duas organizações.

Gerenciar mudanças é bem complicado, pois o ser humano inerentemente rejeita mudanças, especialmente quando são radicais. Mas, como qualquer outra coisa, elas acontecem e devem ser devidamente tratadas para que os objetivos sejam alcançados com sucesso. Gerir mudanças possui exemplos recorrentes como o dito acima, mas existe um outro por experiência própria: o Núcleo de Tecnologia da Informação(NTI) da Universidade Federal de Mato Grosso do Sul(UFMS) estava desenvolvendo um projeto envolvendo as atividades docentes, e, no meio do andamento, ao aplicar uma revisão, foi notada a inconsistência de um requisito com uma mudança que tinha sido tratada no final da *sprint* anterior(o NTI usa um método ágil baseado em SCRUM), o que levou a uma adaptação do projeto.

Mudanças são tão constantes quanto somos inconstantes.

## 2 Nível D - Largamente Definido

No nível D, temos, assim como supracitado, alguns resultados esperados, sendo eles os atributos de processo (AP) 1.1, 2.1, 2.2, 3.1 e 3.2. Assim como já vimos, cada um possui seus Resultados Esperados de Atributos de Processo (RAP); já foram mencionados os RAPs dos APs 1.1 e 2.1, portanto cabe agora definir o restante deles.

No atributo de processo 2.2, o propósito é provar que a quantidade de produtos de trabalho produzidos pelo processo é gerenciada de maneira correta. Temos nele 4 RAPs:

Resultados Esperados	Definição
RAP 11	Os requisitos dos produtos de trabalho do processo são identificados
RAP 12	Requisitos para documentação e controle dos produtos de trabalho são estabelecidos
RAP 13	Os produtos de trabalho são colocados em níveis apropriados de controle
RAP 14	Os produtos de trabalho são avaliados objetivamente com relação aos padrões, procedimentos e requisitos aplicáveis e são tratadas as não conformidades

Tabela 3 – RAPs do AP 2.2

O próximo atributo de processo esperado aqui define a demonstração de um processo padrão que é mantido para apoiar a implementação de um processo definido. Ele possui também 4 RAPs, definidos a seguir:

Resultados Esperados	Definição
RAP 15	Um processo padrão é descrito, incluindo diretrizes para sua adaptação
RAP 16	A sequência e interação do processo padrão com outros processos são determinadas
RAP 17	Os papéis e competências requeridos para executar o processo são identificados como parte do processo padrão
RAP 18	A infra-estrutura e o ambiente de trabalho requeridos para executar o processo são identificados como parte do processo padrão

Tabela 4 – RAPs do AP 3.1

Por fim, no AP 3.2, define-se o quão bem um processo padrão é implementado como um processo definido para atingir seus resultados. Temos 3 RAPs definidos em 5.

Resultados Esperados	Definição
RAP 19	Um processo definido é implementado baseado nas diretrizes para seleção e/ou adaptação do processo padrão
RAP 20	A infra-estrutura e o ambiente de trabalho requeridos para executar o processo definido são disponibilizados, gerenciados e mantidos
RAP 21	Dados apropriados são coletados e analisados, constituindo uma base para o entendimento do comportamento do processo, para demonstrar a adequação e a eficácia do processo, e avaliar onde pode ser feita a melhoria contínua do processo

Tabela 5 – RAPs do AP 3.2

Portanto, no nível D, temos os seguintes processos:

1. Desenvolvimento de Requisitos - DRE;
2. Integração do Produto - ITP;
3. Projeto e Construção do Produto - PCP;
4. Validação - VAL;
5. Verificação - VER;

Aquele que se relaciona com a engenharia de requisitos é o DRE, e portanto será o abordado nessa seção. Vamos inicialmente identificar os resultados esperados para, depois, explicitá-los com mais detalhe.

## 2.1 Desenvolvimento de Requisitos - DRE

Os principais resultados que se espera nesse processo são os seguintes:

- DRE 1. As necessidades, expectativas e restrições do cliente, tanto do produto quanto de suas interfaces, são identificadas;
- DRE 2. Um conjunto definido de requisitos do cliente é especificado e priorizado a partir das necessidades, expectativas e restrições identificadas;
- DRE 3. Um conjunto de requisitos funcionais e não-funcionais, do produto e dos componentes do produto que descrevem a solução do problema a ser resolvido, é definido e mantido a partir dos requisitos do cliente;
- DRE 4. Os requisitos funcionais e não-funcionais de cada componente do produto são refinados, elaborados e alocados; Interfaces internas e externas do produto e de cada componente do produto são definidas;
- DRE 5. Conceitos operacionais e cenários são desenvolvidos;
- DRE 6. Os requisitos são analisados, usando critérios definidos, para balancear as necessidades dos interessados com as restrições existentes;
- DRE 7. Os requisitos são validados.

Agora, o que iremos descrever é como cada uma dessas saídas se relaciona com o que temos nas 5 áreas de Engenharia de requisitos - elicitação, análise, especificação, avaliação e gerência. Iremos também utilizar exemplos ilustrar esses resultados esperados.

### 2.1.1 DRE 1

Aqui, temos que contar com uma boa elicitação de requisitos, para que haja a identificação correta do que o cliente requisita do produto e interfaces. A especificação de requisitos também é essencial para que haja a documentação daquilo que

foi descoberto. É óbvio que, para o alcance desse objetivo, é necessário utilizar de diferentes técnicas de elicitación de requisitos - casos de uso, *brainstorming*, entrevistas e questionários são alguns exemplos que podemos citar.

Um exemplo básico disso que vivenciamos foi em uma entrevista, durante um trabalho de Engenharia de Software, no primeiro período de 2014, aonde pôde-se entender o que o cliente desejava - certos requisitos que nem mesmo o interessado sabia que existia foram identificados, graças a aplicabilidade de tal técnica; algumas respostas acabaram gerando outras dúvidas que, no final, trouxeram à tona funcionalidades ocultas e não previstas.

### 2.1.2 DRE 2

Cabe aqui a área de avaliação de requisitos, responsável por definir quais requisitos são relevantes ou não, bem como a validação dos requisitos - questionando-se se tal requisito está correto ou foi corretamente elicitado. Um exemplo verídico que pode ser citado diz respeito ao mesmo trabalho de Engenharia de Software supracitado, ministrado no primeiro período de 2014 pela professora Débora.

O escopo de nosso "projeto" era a realização do trabalho, que consistia no preenchimento do documento de requisitos, criação de protótipos de interface e análise de pontos por função; ora, para realizar todos esses passos, era necessário primordialmente conduzirmos uma entrevista visando a elicitación dos requisitos. Portanto, aqui, assumindo que a concepção do nosso trabalho consistia em um projeto, tivemos um claro exemplo de priorização de requisitos de acordo com as nossas necessidades, dada que a entrevista deveria ser feita prioritariamente.

### 2.1.3 DRE 3

Como esse resultado envolve manter requisitos, utilizamos a gerência de requisitos para verificar se eles não estão inconsistentes ou incompletos e também para assegurar sua rastreabilidade. Aqui o foco também é na classificação de requisitos funcionais e não-funcionais.



Requisitos funcionais são aqueles que descrevem os serviços que o sistema provê, sendo mais focado no detalhamento de técnicas e métodos a serem utilizados pelo sistema; essa é uma descrição mais específica. Também é possível inferir desse tipo de requisito que, caso um deles deixe de funcionar, o sistema como um todo, pelo menos teoricamente, ainda deveria funcionar, mesmo que incorretamente em alguns aspectos.

Requisitos não-funcionais são aqueles em que uma descrição sobre características e propriedades do sistema são descritas, como tempo de resposta, confiabilidade, usabilidade e portabilidade. É dito que, caso um desses requisitos venha a falhar, o sistema é comprometido como um todo - ou seja, a falha de qualquer um desses requisitos é basicamente inaceitável.

Um exemplo desse resultado esperado se deu durante a elaboração dos *mockups* do trabalho exemplificado: inicialmente descrevemos o sistema com seus requisitos e, após expormos os protótipos aos interessados (como por exemplo o entrevistado), de acordo com suas necessidades, conseguimos definir corretamente requisitos funcionais e não-funcionais.

#### 2.1.4 DRE 4

O processo de refinamento e elaboração de requisitos constitui parte essencial da validação de requisitos. Protótipos são os componentes-chave que definem interfaces externas e internas, e são esperados como resultado aqui. Sua existência permite com que a visualização de tipos e formatos de entrada e saída de dados dos componentes sejam identificados, evitando erros de comunicação entre componentes do produto no momento de sua implementação.

Voltando ao exemplo do trabalho de Engenharia de Software, após a concepção inicial da entrevista, o próximo passo a ser realizado era da concepção dos protótipos de interface, o que caracteriza esse resultado esperado desenvolvendo os requisitos do projeto.

### 2.1.5 DRE 5

Aqui, basicamente o que temos em curso é a utilização da análise de requisitos, realizando uma modelagem desses para conceber os conceitos operacionais e cenários. Cenários tem como principal objetivo mostrar o comportamento entre o usuário e o sistema, ao passo que conceitos operacionais dependem da solução do projeto.

Para ambas as modelagens dessas definições, podemos utilizar da UML para atingir esse fim. No caso dos cenários, diagramas de casos de uso, bem como casos de uso, que descrevem interações entre usuário e sistema, podem ser utilizados para representação. Já no caso de conceitos operacionais, temos diagramas de atividades ou de implantação que auxiliam a compreender melhor o funcionamento entre o produto e o usuário final.

Como exemplos, se retomarmos o projeto de Engenharia de Software que fizemos, foi necessário a elaboração de casos de uso, dessa maneira desenvolvendo os cenários dos requisitos gerados. Para os conceitos operacionais, temos um outro exemplo: durante a matéria de Análise e Projeto de Software Orientado a Objetos, ministrada no segundo período de 2014, foi-nos requisitado um diagrama de atividades para nosso sistema, que era o trabalho da disciplina - portanto, utilizamos essa modelagem para desenvolver os requisitos necessários.

### 2.1.6 DRE 6

Aqui, muito do que se realiza durante a avaliação de requisitos é feita: verifica-se se o que o cliente espera está de acordo com as restrições impostas a esse sistema, através da validação propriamente dita. Também pode-se usar de revisões de requisitos para garantir com que esses sejam necessários, corretos, suficientes e testáveis para os objetivos do produto.

Verificações de erros e inconsistências podem ser realizadas aqui para garantir com que esse resultado seja esperado, bem como remoção de conflitos entre necessidades e restrições de certas expectativas dos *stakeholders*; a utilização dos resultados do DRE 5 para refinamento dos requisitos e a verificação de ambiguidades também são realizadas.

Novamente adotando o trabalho de Engenharia de Software como exemplo, dotados de um roteiro a ser seguido - que no caso era a descrição do trabalho prático - e sendo isso os critérios definidos, as necessidades do entrevistado foram balanceadas com aquilo que podíamos fazer - ou seja, chegamos a gerar as telas como protótipos, mas não implementamos de fato alguma funcionalidade do projeto.

### 2.1.7 DRE 7

Para alcançar esse resultado, o método de validação deve ser definido pela organização de acordo com o projeto em questão. É aqui que ocorre a validação de tudo que foi verificado e avaliado previamente, utilizando-se, junto ao cliente, protótipos, *checklists*, simulações, entre outros.

Como um exemplo, a entrega do trabalho da Engenharia de Software, citado como exemplo durante toda essa seção, serve para definir a validação dos requisitos: Se o que foi entregue está de acordo com aquilo que o cliente (no caso, o professor) esperava, então a validação garantirá a completude dos requisitos; do contrário, a validação e a verificação irão apontar falhas na execução de análise, elaboração e desenvolvimento de cenários dos requisitos desse projeto.

## Conclusão

Nesse documento, abordamos os dois processos do MPS.BR que envolvem a engenharia de requisitos, e seus principais resultados esperados. O modelo de qualidade supracitado vem sendo aceito em médias empresas de software no país ao longo dos anos, e a atenção voltada para a engenharia de requisitos aumentou justamente por causa disso - tanto que o processo de Engenharia de Requisitos é definido no nível G, o primeiro de aceitação do MPS.BR.

Também é importante a elaboração desse documento para entender melhor como se relacionam as saídas esperadas com o que é apresentado na matéria citada, e através de exemplos, é mais fácil compreender os resultados desses processos.

Portanto, esse documento teve como principal finalidade apontar o quão importante a Engenharia de Requisitos é para o desenvolvimento pleno de softwares, visto que ela é definida em processos de níveis do MPS.BR, o modelo de referência que é adotado em nosso país para melhoria de processos de software.

## Referências bibliográficas

Slides disponíveis no EAD na disciplina de Engenharia de Requisitos.

[http://www.softex.br/wp-content/uploads/2013/07/MPS.BR\\_Guia\\_Geral\\_Software\\_20121.pdf](http://www.softex.br/wp-content/uploads/2013/07/MPS.BR_Guia_Geral_Software_20121.pdf)