

GERENCIAMENTO DE VAGAS

PLANO DE TESTE

Versão 2.0

12/07/2016

HISTÓRICO DE VERSÕES

Versão #	Autor	Data da Revisão	Aprovado Por	Data de Aprovação	Revisão
1.0	Lucas Vinicios	07/01/2016	Pedro e Lucas	02/07/2016	Test Plan Template draft
1.1	Joao Henrique	07/02/2016	Pedro e Lucas	02/07/2016	Test Plan Template draft
1.2	Pedro Victor	02/07/2016	Pedro e Lucas	02/07/2016	Introdução finalizada e tópico 2 iniciado
1.3	Lucas	15/07/2016	Pedro e Lucas	15/07/2016	Teste funcional descrito
1.4	Pedro	16/07/2016	Pedro e Lucas	16/07/2016	Teste de unidade descrito
1.5	João	18/07/2016	Pedro e Lucas	18/07/2016	Teste de integração descrito
1.6	Lucas, João e Gabriel	18/07/2016	Pedro e Lucas	18/07/2016	Preenchimento de demais testes
1.7	Pedro	19/07/2016	Pedro e Lucas	19/07/2016	Preenchimento de demais testes e o plano de aprovação
1.8	Pedro, Lucas, Gabriel e João	19/07/2016	Pedro	19/07/2016	Adequação de Casos de Teste do Git no Plano
1.9	João, Lucas, Pedro e Gabriel	19/07/2016	Lucas	19/07/2016	Adequação de Casos de Teste nos Três tipos
2.0	Pedro (Gerente de Projeto)	20/07/2016	Equipe	20/07/2016	Revisão

UP Template Version: 12/31/07

Observação: Todos os Casos de Teste estão disponibilizados no Repositório no Github na pasta “Casos de Teste”.

➤ ***Caminho:*** [vvs-grupo2/2. Casos de Teste/](#)

TABLE OF CONTENTS

INTRODUÇÃO

FINALIDADE DO DOCUMENTO DE PLANO DE TESTE

FUNCTIONAL TESTING

TEST RISKS / ISSUES

ITEMS TO BE TESTED / NOT TESTED

TEST APPROACH(S)

TEST REGULATORY / MANDATE CRITERIA

TEST PASS / FAIL CRITERIA

TEST ENTRY / EXIT CRITERIA

TEST DELIVERABLES

TEST SUSPENSION / RESUMPTION CRITERIA

TEST ENVIRONMENTAL / STAFFING / TRAINING NEEDS

TESTE DE UNIDADE

TEST RISKS / ISSUES

ITEMS TO BE TESTED / NOT TESTED

ABORDAGENS DE TESTE

TESTES DE REGULAMENTAÇÃO/ CRITERIOS DE MANDATO

TESTES DE PASSAGEM/ CRITERIOS DE FALHA

TESTES DE ENTRADA/ CRITERIOS DE SAIDA

TEST DELIVERABLES

SUSPENSÃO DE TESTES

AMBIENTES DE TESTE/EQUIPE DE TESTE

TESTE DE INTEGRAÇÃO

RISCOS ASSOCIADOS AO TESTE

ITENS PARA SER TESTADOS/ NÃO TESTADOS

ABORDAGENS DE TESTE

TESTES DE REGULAMENTAÇÃO/ CRITERIOS DE MANDATO

TESTES DE PASSAGEM/ CRITERIOS DE FALHA

TESTES DE ENTRADA/ CRITERIOS DE SAIDA

TEST DELIVERABLES

SUSPENSÃO DE TESTES

AMBIENTES DE TESTE/ EQUIPE DE TESTES

TEST PLAN APPROVAL

1 INTRODUÇÃO

1.1 FINALIDADE DO DOCUMENTO DE PLANO DE TESTE

O plano de teste, documenta os documentos e rastreia as informações necessárias para definir eficazmente a abordagem a ser utilizada no teste de produto do projeto.

O documento Plano de Teste é criado durante a fase de planejamento do projeto. Seu público-alvo é o gerente do projeto, a equipe de projeto e equipe de testes. Algumas partes deste documento pode ocasionalmente ser compartilhadas com o cliente/usuário e outras partes interessadas cuja entrada/aprovação no processo de teste é necessário.

Este documento apresenta o planejamento das atividades de testes do sistema Exemplo o qual será utilizado como base para as atividades de acompanhamento, revisão, verificação e validação do projeto, desde seu início até sua conclusão, a fim de garantir a análise comparativa do resultado realversus planejado. Desta forma, ações corretivas e preventivas poderão ser tomadas, sempre que resultados reais desviarem significativamente do planejado.

2 FUNCTIONAL TESTING

2.1 TEST RISKS / ISSUES

O produto deve contemplar com sucesso os seguintes riscos para ser funcional de acordo com o desejado:

- Login e logout no sistema
- Publicação de vagas
- Realização de inscrição
- Agendamento de entrevista
- Avaliação de candidatos
- Reportação de resultados
- Manter usuários
- Envio de email

2.2 ITEMS TO BE TESTED / NOT TESTED

- ❖ O desenvolvedor (Gabriel) não se enquadra nos testes abaixo, deixando os demais membros simularem as funcionalidades, a fim de que os testes sejam totalmente realizados e validados ou não.

Item to Test	Test Description	Test Date	Responsibility
Login no sistema - Caso de teste	Realizar Login no sistema utilizando nome e senha.	17/07/2016	João
Logout no sistema	Realizar Logout no sistema.	17/07/2016	João

Publicação de vagas - Caso de teste	Publicar uma vaga no sistema	17/07/2016	João
Realização de inscrição - Caso de teste	Realizar uma inscrição no sistema	17/07/2016	Pedro
Agendamento de entrevista - Caso de teste	Agendar uma entrevista	17/07/2016	Pedro
Entrevista de candidatos - Caso de teste	Realizar o agendamento da entrevista dos candidatos	17/07/2016	Pedro
Avaliação de candidatos - Caso de teste	Avaliar os candidatos	17/07/2016	Pedro
Reportação de resultados - Caso de teste	Reportar os resultados	17/07/2016	Lucas
Manter usuários - Caso de teste	Manter os usuários no sistema	17/07/2016	Lucas
Envio de email - Caso de teste	Enviar email para os usuários	17/07/2016	Lucas

2.3 TEST APPROACH(S)

Os testes desta seção serão realizados em um ambiente de homologação do sistema e/ou durante a execução do desenvolvimento do sistema e a respectiva funcionalidade a ser testada. Os dados de clientes e vagas e os mesmos ligados a estes serão armazenados em um banco de dados de homologação de modo a verificar o funcionamento. Clientes, empresas e dados ligados a estes utilizados nos teste são reais para a verificação da conformidade das funcionalidades. A seguir é identificado um esboço do teste recomendado para cada aplicativo:

Objetivo do Teste: Assegurar a navegação correta do sistema, além da entrada/saída, processamento de funcionalidades e recuperação de dados.

Técnica:

- Executar cada caso de uso, fluxo de caso de uso ou função, utilizando dados válidos e inválidos, para verificar o seguinte:
- Os resultados esperados ocorrerão quando forem usados dados válidos.
- As mensagens de erro / aviso apropriadas sejam exibidas quando dados inválidos forem utilizados.
- Cada regra de negócio será adequadamente aplicada.

CrITÉRIOS de Conclusão:

- Todos os testes planejados foram executados.
- Todos os defeitos identificados foram tratados.

2.4 TEST REGULATORY / MANDATE CRITERIA

Não existem regulamentos conhecidos ou mandatos que o sistema deve ser testado contra.

2.5 TEST PASS / FAIL CRITERIA

O produto deve atender a todos os requisitos funcionais especificados. Se não conseguir executar em qualquer um dos ensaios acima referidos, o sistema tem de gerar um erro que irá notificar o usuário.

2.6 TEST ENTRY / EXIT CRITERIA

Checklist critérios de entrada	
#	Itens
1	Tag do código-fonte
2	Lista de defeitos corrigidos
3	Lista de recursos / requisitos implementados ou alterados
4	Lista de problemas conhecidos
5	Caso de teste e relatório de execução para DEV nível (inclui data de modificação para casos de teste, data de execução e resultados) --Unit teste
6	Unidade de Relatório de Teste - # passa / falha, % cobertura
7	Camada está pronta para uso (provisionado, corresponde pilha de tecnologia)
8	documentação de implantação
9	Guia de instalação (instalar a partir do zero)
10	Casos de Teste para esta tag foram revistos pelos requisitos analista ou equivalente
11	Plano de Teste revisado e assinado.

Checklist critérios de saída	
#	Itens
1	Tag do código-fonte

2	Lista de defeitos corrigidos - Relatório
3	Lista de recursos / requisitos implementados ou alterados - Relatório
4	Lista de problemas conhecidos - Relatório
5	Casos de teste e relatório de execução para camada (inclui data de modificação para casos de teste, data de execução e resultados)
6	documentação de implantação - Avaliado
7	Guia de instalação (instalar a partir do zero) - Review / Relatório
8	Os casos de teste, e uma lista de casos de teste relacionados ao exercício de funções que poderiam ser afetados por vulnerabilidades de segurança, são dadas a para o sistema.
9	Outras complicações ocorridas(se aplicáveis)

2.7 TEST DELIVERABLES

Desempenho satisfatório nos testes acima devem ser documentadas suficientemente para permitir que futuros integrantes da equipe de discernir claramente como o teste foi realizado, o limites de cada respectivo teste, o desempenho em cada iteração do ensaio, bem como o desempenho nos testes bem sucedidos. Esta documentação deve ser feita disponível para todos os membros da equipe, e pode ser colocado à disposição do cliente em seu pedido. Serão utilizados relatórios e gráficos para tal.

2.8 TEST SUSPENSION / RESUMPTION CRITERIA

Os critérios que podem ser utilizados para suspender a totalidade ou porções de um ensaio funcional seria em grande parte ser relacionado com a instabilidade do ambiente de teste e incluem: falha de hardware ou de software, a presença de um vírus ou malware relacionados identificados de existir em dispositivos utilizados para ensaio tal como servidores e PCs. Os critérios a serem utilizados para retomar o teste seria que qualquer condição que levam a uma falha de hardware, software ou sistemas tem sido atenuado e autoridades competentes determinaram que a condição não está mais presente. Em adição, um subconjunto de ensaios funcionais vai ser executado como um mecanismo para assegurar o ambiente de teste e aplicação em teste são estáveis para retomar o esforço de teste funcional. No caso em que a suspensão viesse a ocorrer e antes de retomar o teste funcional, o sistema fará uso de nossos scripts de testes

automatizados para o navegador para determinar se deve continuar o teste.

2.9 TEST ENVIRONMENTAL / STAFFING / TRAINING NEEDS

Os responsáveis pelo teste devem possuir um conhecimento profundo e completo da operação de ambos os sistemas em teste, bem como o conhecimento das ferramentas de serviço primário e o seu funcionamento adequado. O teste foi realizado por toda a equipe de desenvolvimento. O ambiente para os testes é definido por meio da compatibilidade e deve ser padronizado pela equipe de testes. Os usuários finais devem ser instruídos para o uso das funcionalidades do sistema assim como requer um conhecimento básico de computação destes para o uso deste sistema sendo que para os mesmo serão necessários a realização de treinamentos.

3 TESTE DE UNIDADE

3.1 TEST RISKS / ISSUES

O Teste deve verificar de maneira isolada, simulando dependências que aquela unidade tem. Será analisado as ligações entre as classes e módulos, a fim de saber se isoladamente cada classe mantém o funcionamento em plena execução, mesmo havendo interassões. As classes ou módulos que não funcionarem, após realizado tais testes serão corrigidas.

3.2 ITEMS TO BE TESTED / NOT TESTED

- ❖ Necessidade de automação
- ❖ Clica um botão a qualquer momento para testar tudo!
- ❖ Pelo menos um teste por dia (frequentemente vários testes por hora)
- ❖ Uso de um framework de testes
- ❖ Desenvolvimento e testes em paralelo
- ❖ Absolutamente necessário ter testes de unidade, principalmente se fizer refactoring de código
- ❖ Testar as classes da menos acoplada para a mais acoplada
- ❖ Na realidade, segue a ordem de desenvolvimento
- ❖ Desenvolvimento e testes de unidade feitos em paralelo!
- ❖ Será utilizado o framework JUnit para testar os Casos de Teste criados a partir dos Casos de Uso.
- ❖ Os testes de Unidade serão em maioria realizado por nosso desenvolvedor/testador, uma vez que o mesmo está acostumado com a estrutura.
- ❖ Estes serão revisados pelos demais membros da equipe, a fim de que não haja vícios do desenvolvedor e que a integridade do código seja mantida.

Item para Testar	Data do Teste	Responsabilidade
------------------	---------------	------------------

Casos de Teste 01 (Clique para ir aos CT's).	15/07/2016 à 25/07/2016	João Henrique
Casos de Teste 02 (Clique para ir aos CT's).	25/07/2016 à 05/08/2016	Gabriel Barbosa
Casos de Teste 03 (Clique para ir aos CT's).	05/08/2016 à 15/08/2016	Gabriel Barbosa
Casos de Teste 04 (Clique para ir aos CT's).	15/07/2016 à 25/07/2016	Gabriel Barbosa
Casos de Teste 05 (Clique para ir aos CT's).	25/07/2016 à 05/08/2016	Pedro Victor
Casos de Teste 06 (Clique para ir aos CT's).	05/08/2016 à 15/08/2016	Gabriel Barbosa
Casos de Teste 07 (Clique para ir aos CT's).	15/07/2016 à 25/07/2016	Lucas Vinicios
Casos de Teste 08 (Clique para ir aos CT's).	25/07/2016 à 05/08/2016	Gabriel Barbosa
Casos de Teste 09 (Clique para ir aos CT's).	05/08/2016 à 15/08/2016	Gabriel Barbosa

3.3 ABORDAGENS DE TESTE

Abordagem estrutural (“caixa branca”)

- ❖ Essa técnica trabalha diretamente sobre o código fonte do componente de software para avaliar aspectos tais como: teste de condição, teste de fluxo de dados, teste de ciclos, teste de caminhos lógicos, códigos nunca executados.
- ❖ O Gabriel tem acesso ao código fonte da aplicação e pode construir códigos para efetuar a ligação de bibliotecas e componentes. Este tipo de teste é desenvolvido analisando o código fonte e elaborando casos de teste que cubram todas as possibilidades do componente de software.
- ❖ Iremos usar a ferramenta livre JUnit para desenvolvimento de classes de teste para testar classes ou métodos desenvolvidos em Java .Também se enquadram nessa técnica testes manuais de software.
- ❖ Com este Teste de caixa branca, analisaremos a estrutura interna do Gerenciamento de Cargos e seu comportamento, necessitando que o Gabriel entenda sua implementação, isso será possível, pois ele também é um de nossos desenvolvedores.
- ❖ Foco dos testes caixa branca está em como os resultados dos testes forma

obtidos.

3.4 TESTES DE REGULAMENTAÇÃO/ CRITERIOS DE MANDATO

Não existem regulamentos conhecidos ou mandatos que o sistema deve ser testado contra.

3.5 TESTES DE PASSAGEM/ CRITERIOS DE FALHA

A constatação de que os módulos estão coerentes e seguindo uma linha formada, com simples leitura e se ligações que possam na frente causar impactos em mais de uma classe, caso haja mudanças, é a solução para termos o caso de teste aprovado ou não.

3.6 TESTES DE ENTRADA/ CRITERIOS DE SAIDA

Os testes de entrada serão as dependências entre as classes, a verificação se cada uma, independente, consegue executar a função que fora atribuída.

3.7 TEST DELIVERABLES

Com o desempenho positivo nos testes de unidade realizados, estes devem ser documentadas suficientemente para permitir que futuros integrantes da equipe possam compreender tranquilamente como o teste foi realizado, o limites de cada respectivo teste, o desempenho em cada iteração do ensaio, bem como o desempenho nos testes bem sucedidos. Esta documentação deve ser feita disponível para todos os membros da equipe, e pode ser colocado à disposição do cliente em seu pedido. Serão utilizados relatórios e gráficos para tal. Caso outro membro tenha que realizar o teste, o Gabriel deverá disponibilizar em uma forma de documento ou esboço para toda equipe, todas as ligações entre classes, módulos, suas dependências.

3.8 SUSPENSÃO DE TESTES

Os critérios que podem ser utilizados para suspender a totalidade ou porções de um ensaio funcional seria em grande parte ser relacionado com a instabilidade do ambiente de teste e incluem: falha de hardware ou de software, a presença de um vírus ou malware relacionados identificados de existir em dispositivos utilizados para ensaio tal como servidores e PCs. Se a ferramenta utilizada para ajudar na inspeção e realização dos testes de unidade, estiver trazendo resultados muito diferente do que esperado. Os critérios a serem utilizados para retomar o teste seria que qualquer condição que levam a uma falha de hardware, software ou sistemas tem sido atenuado e autoridades competentes determinaram que a condição não está mais presente. Em adição, um subconjunto de ensaios funcionais vai ser executado como um mecanismo para assegurar o ambiente de teste e aplicação em teste são estáveis para retomar o esforço de teste funcional. No caso em que a suspensão viesse a ocorrer e antes de retomar o teste

funcional, o sistema fará uso de nossos scripts de testes automatizados para o navegador para determinar se deve continuar o teste.

3.9 AMBIENTES DE TESTE/EQUIPE DE TESTE

Os responsáveis pelo teste devem possuir um conhecimento profundo e completo da operação de ambos os sistemas em teste, bem como o conhecimento das ferramentas de serviço primário e o seu funcionamento adequado. O teste foi realizado por toda a equipe de desenvolvimento. O ambiente para os testes é definido por meio da compatibilidade e deve ser padronizado pela equipe de testes. O membro mais experiente da equipe em relação à teste, Gabriel, foi alocado justamente pois conhece bem o ambiente e sabe mais que os demais membros, como fazer a verificação unitária.

4 TESTE DE INTEGRAÇÃO

4.1 RISCOS ASSOCIADOS AO TESTE

O riscos associados com os testes de produto podem ser encontrados no **tópico 3** dos casos de uso presentes no [repositório](#).

4.2 ITENS PARA SER TESTADOS/ NÃO TESTADOS

Após a implementação dos módulos, estes devem passar por uma etapa de integração, de forma a dar início à formação de um sistema como um todo. Para verificar sua corretude após a integração, são realizados os testes de integração. Vale lembrar que, ainda que os testes de unidade verifiquem que os componentes estão agindo da forma esperada, após suas integrações, a situação pode não ser a mesma, uma vez que haverá a união de diferentes (ou não) tipos de dados.

Item to Test	Test Description	Test Date	Responsibility
Login no sistema - Caso de teste	Realizar Login no sistema utilizando nome e senha.	17/07/2016	João
Logout no sistema	Realizar Logout no sistema.	17/07/2016	João
Publicação de vagas - Caso de teste	Publicar uma vaga no sistema	17/07/2016	João
Realização de inscrição - Caso de teste	Realizar uma inscrição no sistema	17/07/2016	Pedro
Agendamento de entrevista - Caso de teste	Agendar uma entrevista	17/07/2016	Pedro
Entrevista de candidatos - Caso de teste	Realizar o agendamento da entrevista dos candidatos	17/07/2016	Pedro
Avaliação de candidatos - Caso de teste	Avaliar os candidatos	17/07/2016	Pedro
Reportação de resultados - Caso de teste	Reportar os resultados	17/07/2016	Lucas
Manter usuários -	Manter os usuários no sistema	17/07/2016	Lucas

Caso de teste			
Envio de email - Caso de teste	Enviar email para os usuários	17/07/2016	Lucas

4.3 ABORDAGENS DE TESTE

A integração será realizada de forma contínua, ou seja, incremental, onde o sistema será integrado por partes. Dessa forma, conseguimos isolar o erro e evitar a chamado integração “Big Bang”, onde, neste último caso, integrariamos todas as funcionalidades de uma só vez, mas estaríamos abertos à uma grande quantidade de erros, podendo ser encontrados em qualquer parte do sistema. Buscamos evitar ao máximo esta situação.

4.4 TESTES DE REGULAMENTAÇÃO/ CRITERIOS DE MANDATO

Não existem regulamentos conhecidos ou mandatos que o sistema deve ser testado contra.

4.5 TESTES DE PASSAGEM/ CRITERIOS DE FALHA

Para averiguarmos se um item (ou conjunto de itens, neste caso) passou ou falhou em um teste, utilizaremos conceitos muito básicos, como uma entrada de dados, ou seja, valores pré estabelecidos pelo testador que serão introduzidos no local desejado, e também saídas esperadas, valores e respostas que deveriam ser apresentáveis após o sistema ser executado. Caso os valores de saída caracterizem de forma correta os valores de entrada, gerem os resultados esperados, tais casos de teste serão identificados como “passaram no teste”, e caso contrário, “não passaram no teste”.

4.6 TESTES DE ENTRADA/ CRITERIOS DE SAIDA

Será utilizado como critério de entrada de que, antes de darmos início aos testes de integração, todos os testes de verificação funcionais e de unidade tenham sido tentados, e que pelo menos 90% deles tenham sido concluídos. Ao passo que temos como critério de saída que todos os testes de integração tenham pelo menos sido tentados, e que pelo menos 90% destes tenham sido concluídos.

4.7 TEST DELIVERABLES

Após obtermos um desempenho satisfatório nos testes de integração que serão realizados, estes devem ser documentadas suficientemente para permitir que futuros integrantes da equipe de discernir claramente como o teste foi realizado, o limites de cada respectivo teste, o desempenho em cada iteração do ensaio, bem como o desempenho nos testes bem sucedidos. Esta documentação deve ser feita disponível para todos os membros da equipe, e pode ser colocado à disposição do cliente em seu pedido. Serão utilizados relatórios e gráficos para

tal.

4.8 SUSPENSÃO DE TESTES

Os critérios que podem ser utilizados para suspender a totalidade ou porções de um ensaio funcional seria em grande parte ser relacionado com a instabilidade do ambiente de teste e incluem: falha de hardware ou de software, a presença de um vírus ou malware relacionados identificados de existir em dispositivos utilizados para ensaio tal como servidores e PCs. Os critérios a serem utilizados para retomar o teste seria que qualquer condição que levam a uma falha de hardware, software ou sistemas tem sido atenuado e autoridades competentes determinaram que a condição não está mais presente. Em adição, um subconjunto de ensaios funcionais vai ser executado como um mecanismo para assegurar o ambiente de teste e aplicação em teste são estáveis para retomar o esforço de teste funcional. No caso em que a suspensão viesse a ocorrer e antes de retomar o teste funcional, o sistema fará uso de nossos scripts de testes automatizados para o navegador para determinar se deve continuar o teste.

4.9 AMBIENTES DE TESTE/ EQUIPE DE TESTES

Os responsáveis pelo teste devem possuir um conhecimento profundo e completo da operação de ambos os sistemas em teste, bem como o conhecimento das ferramentas de serviço primário e o seu funcionamento adequado. O teste foi realizado por toda a equipe de desenvolvimento. O ambiente para os testes é definido por meio da compatibilidade e deve ser padronizado pela equipe de testes.

TEST PLAN APPROVAL

O abaixo assinado reconhece que o membros revisaram o Plano de Teste do projeto *Gerenciamento de Vagas* e concordam com a abordagem que apresenta. Quaisquer alterações a esta Definição de Requisitos serão coordenadas com e aprovado pelo abaixo-assinado ou os seus representantes designados.

Signature:	_____	Date:	_____
Print Name:	Pedro Victor de Almeida Lopes		
Title:	Gerente de Projeto		
Role:	Planejar e acompanhar áreas do Projeto, juntamente com os demais gerentes atribuídos a cada Área.		

Signature:	_____	Date:	_____
Print Name:	Lucas Vinicios Ferreira S.		
Title:	Gerente de Teste		
Role:	Planejar e Acompanhar se o plao de teste está sendo seguido.		

Signature:	_____	Date:	_____
Print Name:	João Henrique C. S. Silva		
Title:	Gerente de Requisitos		
Role:	Rastrear os Requisitos		

Signature:	_____	Date:	_____
Print Name:	Gabriel Barbosa Oliveira		
Title:	Desenvolvedor/Testador		
Role:	_____		

Appendix A: References

[Insert the name, version number, description, and physical location of any documents referenced in this document. Add rows to the table as necessary.]

The following table summarizes the documents referenced in this document.

Document Name and Version	Description	Location
<Document Name and Version Number>	<i>[Provide description of the document]</i>	<URL or Network path where document is located>

Appendix B: Key Terms

[Insert terms and definitions used in this document. Add rows to the table as necessary. Follow the link below to for definitions of project management terms and acronyms used in this and other documents.]

<http://www2.cdc.gov/cdcup/library/other/help.htm>

The following table provides definitions for terms relevant to this document.

Term	Definition
<i>[Insert Term]</i>	<i>[Provide definition of the term used in this document.]</i>
<i>[Insert Term]</i>	<i>[Provide definition of the term used in this document.]</i>
<i>[Insert Term]</i>	<i>[Provide definition of the term used in this document.]</i>