

UNIVERSIDADE DO VALE DO RIO DOS SINOS — UNISINOS
UNIDADE ACADÊMICA DE GRADUAÇÃO
GRADUAÇÃO DE ANÁLISE E DESENVOLVIMENTO DE SISTEMA
NÍVEL GRADUAÇÃO

JOÃO CARLOS BRASILEIRO STEFENON DE ALMEIDA

ANÁLISE DO IMPACTO SOBRE O PROCESSO DE TESTE COM *CLOUD*
COMPUTING

SÃO LEOPOLDO
2011

João Carlos Brasileiro Stefenon de Almeida

ANÁLISE DO IMPACTO SOBRE O PROCESSO DE TESTE COM *CLOUD*
COMPUTING

Dissertação apresentada como requisito
parcial para a obtenção do título de
graduado pelo graduação em Análise e
desenvolvimento de sistemas da
Universidade do Vale do Rio dos Sinos —
UNISINOS

Orientador:
Prof. Magrid Krug

São Leopoldo
2011

Aos nossos pais.

*If I have seen farther than others,
it is because I stood on the shoulders of giants.*
— SIR ISAAC NEWTON

AGRADECIMENTOS

Obrigado!

“Ningu m abre um livro sem que aprenda alguma coisa”.
(An nimo)

RESUMO

Um dos principais objetivos do processo de desenvolvimento de software sempre foi a otimização na utilização dos recursos disponíveis, procurando sempre aumentar a eficiência e diminuir os custos envolvidos. Com as atuais tecnologias este objetivo vem sendo alcançado, e com isso, facilitando o processo de gerência, e possibilitando implementar uma infraestrutura capaz de compartilhar recursos através da internet. Criando assim novos cenários, e novos modelos para gerenciar a tecnologia da informação (TI), e com isso, novas características, limitações e principalmente novos desafios a serem enfrentados. Este artigo tem o objetivo de identificar qualidades, desafios e apresentar um ponto de vista do processo de testes com essas novas tecnologias, apresentando também possíveis tipos de técnicas e estratégia que possivelmente podem ser utilizadas para conduzir o processo de teste e cobrir essas novas características.

Palavras-chave: UNISINOS. ABNT. \LaTeX . Formatação de documentos.

ABSTRACT

Since a long time ago, even in the begin of the software development process, the need of an effective sharing of resource and components always was a main goal. It has been the focus of an optimized process, minimizing the cost effort and facilitating the activities to the management. Now with ours actual technologies, it can be possible and easily achieve this goal, implementing an infrastructure that can generate new sceneries in the Information Technology, and bring new big challenges and changes in the development process. This article has the main focus on identifying the qualities, challenges and show a point of view from test Process with these new technologies, and how possibly we can handle from now on, and how we possibly can conduct the Test process to cover these new features that comes with it.

Keywords: UNISINOS. ABNT. L^AT_EX. Document typesetting.

SUMÁRIO

1	INTRODUÇÃO	15
2	OBJETIVOS DO TRABALHO	17
2.1	Metodologia do Trabalho	17
2.2	Cronograma	17
3	O QUE É COMPUTAÇÃO EM NUVENS (<i>CLOUD COMPUTING</i>)	19
3.1	Conceito de <i>Cloud Computing</i>	19
3.2	Modelos de Serviços	19
3.2.1	Software como serviço (SaaS)	20
3.2.2	Plataforma como Serviço (PaaS)	20
3.2.3	Infraestrutura como Serviço (IaaS)	21
3.2.4	X como Serviço (XaaS)	21
3.3	Modelos de Implementação	23
3.3.1	<i>Public Cloud</i> - Nuvem Pública	23
3.3.2	<i>Private Cloud</i> - Nuvem Particular	23
3.3.3	<i>Hybrid Cloud</i> - Nuvem Híbrida	24
3.3.4	<i>Community Cloud</i> - Nuvem Comunitária	24
3.4	Gestão da Tecnologia com Computação em Nuvens	24
3.5	Os Novos Riscos Envolvidos	26
4	TESTE ESTRUTURADO	29
4.1	Característica do Teste Estruturado	29
4.2	Princípios e Principais conceitos em teste de Software	29
4.3	Modelos de processo de teste	31
4.3.1	MPT.BR - Melhoria do Processo de Teste	32
4.3.2	TMMI - <i>Test Maturity Model integration</i>	35
4.3.3	TPI NEXT - <i>Test Process Improvement</i>	37
5	ESTRATÉGIA DE TESTE	41
	REFERÊNCIAS	45
	APÊNDICE A INFORMAÇÕES COMPLEMENTARES	49
	ANEXO A ARTIGOS PUBLICADOS	51

1 INTRODUÇÃO

A computação em nuvens ainda é um conceito novo na área de Tecnologia da Informação (TI), REESE (2009) afirma que este conceito traz alguns desafios, em particular ao processo de teste de software. Portanto, não se conhece claramente como esse novo conceito irá afetar os processos da TI e qual o impacto nos padrões e modelos utilizados até hoje. MILLER (2008) menciona que o uso da computação em nuvens traz uma série de vantagens, facilidades no acesso, e principalmente o compartilhamento de recursos, que vem de encontro direto as atuais necessidades crescentes do mercado, mas também traz algumas outras características importantes como a velocidade de adaptação, uma maior agilidade de entrega e mesmo no processo de descontinuação dos sistemas e sua infraestrutura.

O gerente de Novas Tecnologias Aplicadas da IBM Brasil, TAURION (2011) afirma que *Cloud Computing*, ou computação em nuvem, é uma verdadeira transformação na maneira de se gerenciar e entregar a Tecnologia da Informação. Esse ambiente vai transformar a área de TI em serviço, não só para quem consome, mas como também para quem fornece. Tudo será negociado em forma de serviço. Baseado nessas variáveis, este novo modo de gerenciar a Tecnologia da informação, embora esteja em um estágio inicial de maturidade, certamente terá uma clara evolução e os chamados “*Cloud Model*” (Modelos em Nuvem) ganharão uma grande parte das instalações no decorrer do tempo, e estará presente em muitas estratégias de negócio das empresas. Conforme é possível visualizar com a seguinte pesquisa realizada:

“ Atualmente, 18% das médias e grandes empresas brasileiras usam alguma aplicação de Cloud computing, sendo que a grande maioria delas utiliza e-mail e ferramenta de publicidade. Essa parcela deve saltar para entre 30% e 35% em 2013, de acordo com projeção da IDC. Segundo a consultoria, o aumento da demanda resultará em um crescimento anual de 60% do mercado de computação em nuvem no Brasil nos próximos anos. “

(FIGUEIREDO, 2011)

Com a utilização e implementação deste novo tipo de estrutura, aonde tornará possível se conectar facilmente com demais tipos de arquiteturas, sistemas ou outros ambientes, tanto as tradicionais quanto as estruturas antigas nas quais são mantidos sistemas legados, normalmente os mais críticos e complexos. Essas facilidades, acabam impactando a forma aonde o processo de teste atua. () afirma que a utilização dos modelos nas nuvens nada do que anteriormente era realizado terá mudança, o que era testado continuará a ser testado, porém, incluirá uma nova perspectiva, novas características para o processo de teste de software. O teste incorporado em “*Cloud Models*”, possui muitos riscos envolvidos na

integração, na utilização em paralelos dos recursos, a exigência de muitos padrões de serviço, ou acordos de Níveis de Serviço - ANS ¹ para este tipo de infraestrutura, novos quantificadores para avaliar atributos, e o aumento do grau de importância dos requisitos não funcionais. São estes novos atributos torna o Teste em Nuvens diferente do processo de teste tradicional. Uma das principais consequências é a característica que esse conjunto de serviços apresenta: recursos virtuais utilizáveis e acessíveis, a capacidade e a dinâmica de como podem ser reconfigurados para se ajustarem de cenário a cenário, ou para uma carga variável de requisições, e a otimização e o compartilhamento do uso dos recursos. Na qual desenvolve a oportunidade de explorar os recursos através dos modelos *pay-per-use*, e apresentam as garantias oferecidas através de acordos de nível de serviço.

¹Tradução do inglês *Service Level Agreement* - *SLA*

2 OBJETIVOS DO TRABALHO

2.1 Metodologia do Trabalho

O presente trabalho utiliza o método de Pesquisa-ação, segundo THIOLENT (1986) é uma pesquisa social com base empírica que é concebida e realizada em estreita associação com uma ação ou com a resolução de um problema coletivo, sendo uma forma de experimentação.

Este trabalho tem como objetivo principal entender os impactos do *Cloud Computing*, identificar as principais características e os riscos envolvidos, e então analisar a situação atual da maturidade do processo de teste, com base nestes conhecimentos, projetar uma nova estratégia para cobertura de teste, as possíveis ações a serem tomadas, técnicas a serem utilizadas, e o planejamento para que seja possível cobrir os novos riscos.

Serão abordados e aprofundados os conceitos de *Cloud Computing* em sua parte inicial, entendendo um pouco melhor sua estrutura, principais conceitos envolvidos, identificar alguns benefícios e principalmente os riscos que esta tecnologia poderá, e irá trazer consigo.

Em sua segunda parte, será entender e apresentar conceitos sobre a maturidade do processo de teste, mostrando alguns dos modelos de processo de teste, e então, identificar suas características principais, benefícios da utilização de um modelo de maturidade.

Será abordada, em sua terceira parte, uma nova estratégia para o processo de teste, a fim de cobrir os riscos identificados anteriormente, mostrando as principais necessidades, técnicas, projetar possíveis dados e artefatos necessários para quantificar a qualidade do software/serviços.

Em sua penúltima etapa, será realizado uma tentativa de aplicação prática desta nova estratégia, coletando dados, métricas e valores quantificáveis para monitoramento, identificar a viabilidade, e então apresentar os resultados finais.

Na ultima etapa será realizado a conclusão do trabalho, apresentando todas as considerações, dificuldades, as etapas necessárias para conclusão e por fim o resultado da proposta.

2.2 Cronograma

O trabalho terá como base as datas conforme mostra a FIGURA 1.

	2011/2							2012/1						
	10/08/2011	25/08/2011	26/08/2011	25/09/2011	10/10/2011	25/10/2011	24/11/2011	09/12/2011	15/12/2011	14/01/2012	28/02/2012	29/03/2012	05/2012	06/2012
Reunião Inicial	x													
Elaboração da proposta														
Entrega da Proposta														
Revisão Bibliográfica														
Estudos sobre Teste em Cloud Computing														
Avaliação de modelos de processos														
Definição da Estratégia e Técnicas de Teste														
Desenvolvimento do RA														
Entrega do Relatório de Andamento (RA)								x						
Revisão Bibliográfica														
Analisar mudanças com modelos nas Nuvens														
Definição contrução do Plano de Teste														
Definir, criar e levantar Ambiente de testes														
Avaliar os novos desafios, riscos da estratégia														
Execução da estratégia de testes														
Avaliação dos resultados														
Elaboração das conclusões														
Entrega da monografia													x	
Defesa do trabalho														x

Figura 1: Cronograma do trabalho de conclusão

3 O QUE É COMPUTAÇÃO EM NUVENS (*CLOUD COMPUTING*)

3.1 Conceito de *Cloud Computing*

O NIST, **National Institute of Standards and Technology**, apresenta o seguinte conceito:

“ Computação em nuvem é um modelo que possibilita acesso, de modo conveniente e sob demanda, a um conjunto de recursos computacionais configuráveis (por exemplo, redes, servidores, armazenamento, aplicações e serviços) que podem ser rapidamente adquiridos e liberados com mínimo esforço gerencial ou interação com o provedor de serviços“. (MELL. P, 2009)

Segundo KNORR e GRUMAN (2011) este novo tipo de estrutura vem com o foco de viabilizar um dos grandes objetivos da TI, apresentando um novo modo de aumentar a capacidade, ou mesmo criar novas capacidades temporariamente a partir de uma necessidade pontual sem precisar estar investindo em uma nova infra-estrutura, treinamentos ou licenças para aquisição de software. *Cloud* tem uma referência a computação baseada na internet, utilizando compartilhamento de recursos e software como um serviço por demanda, o que significa uma grande redução de esforço e custo no gerenciamento, e envolve o que é chamado de serviços *On-Demand* (OD), para o português, serviços por demanda, e qualquer serviço por contrato, ou *”Pay-Per-use”*.

3.2 Modelos de Serviços

Quando falamos de *Cloud Computing* devemos ter em mente que podemos escolher entres três desses modelos, e que cada um representa uma estratégia distinta.

- Software as a Service (SaaS) - Software como serviço ,
- Platform as a Service (PaaS) - Plataforma como Serviço ,
- Infrastructure as a Service (IaaS) - Infraestrutura como Serviço.

É possível visualizar na TABELA 1 em negrito, basicamente o que é possível gerenciar nesses diferentes tipos de modelos.

Tabela 1: Tabela de Gerenciamento dos modelos de *Cloud Computing*

IaaS	PaaS	SaaS
Aplicações	Aplicações	Aplicações
Intermediárias	Intermediárias	Intermediárias
Framework de aplicação	Framework de aplicação	Framework de aplicação
Sistema Operacional	Sistema Operacional	Sistema Operacional
Virtualização	Virtualização	Virtualização
Hardware	Hardware	Hardware
Conectividade	Conectividade	Conectividade
Data Center	Data Center	Data Center

3.2.1 Software como serviço (SaaS)

Software As A Service (SaaS), é uma solução de software oferecida como um serviço, que é acessado pelos usuários através da Internet, sem a necessidade de implantar e manter uma premissa em infra-estrutura de TI. (THINKSTRATEGIES, 2005).

Este modelo de serviço é definido como a entrega baseada numa aplicações de software completa, que são executados em um infra-estrutura gerenciada pelo fornecedor, sendo acessadas através da Internet, TAXONOMY (2010). Um exemplo simples, uma empresa que contrata um aplicativo de e-mail para os seus funcionários. Este aplicativo não pode ser modificado, e seus usuários não podem ser alterados, com exceção de algumas possíveis configurações locais para cada usuário ou personalizações que serão permitidas pelo provedor. A aplicação será hospedada no provedor de infraestrutura em nuvem e o usuário não terá nenhum controle sobre esta estrutura.

3.2.2 Plataforma como Serviço (PaaS)

Segundo Zoho (2010) Plataforma como Serviço (PaaS) é a entrega de uma plataforma através da web, na qual, permite criar aplicações e sistemas de forma rápida, sem o custo e a complexidade do processo de aquisição, sendo ao comprar, construir e/ou gerenciar os softwares e hardwares relacionados, o que oferece todas as facilidades necessárias para manter um ciclo desenvolvimento completo, e a entrega de aplicações realizada inteiramente através da internet. Neste cenário, o contratante estará adquirindo um serviço que permite hospedar e desenvolver seus próprios aplicativos em uma plataforma que fornece ferramentas e um ambiente de desenvolvimento, assim sendo possível desenvolver uma

solução. Neste modelo, o fornecedor oferece o uso de sua plataforma, que por sua vez, está alojado em sua infra-estrutura, aonde não será possível para o usuário ter controle sobre a plataforma ou infra-estrutura, mas somente terá sobre suas aplicações.

3.2.3 Infraestrutura como Serviço (IaaS)

MELL. P (2009) define para IaaS como a capacidade de fornecer ao consumidor a prestação de processamento, armazenamento, redes e outros recursos fundamentais onde o consumidor deve ser capaz de implementar e executar software arbitrário, que pode incluir sistemas operacionais e aplicações. O consumidor não administra ou controla o serviço de infra-estrutura, mas tem controle sobre os sistemas operacionais, armazenamento, aplicativos implementados, e possivelmente, controles limitados sobre os componentes definidos da rede (por exemplo, firewalls do host). Neste modelo, o usuário estará contratando apenas a infra-estrutura tecnológica (poder de processamento, armazenamento e/ou comunicações).

A FIGURA 2 de CZERNICKI (2011) mostra como entender de uma melhor maneira cada serviço em uma palavra.

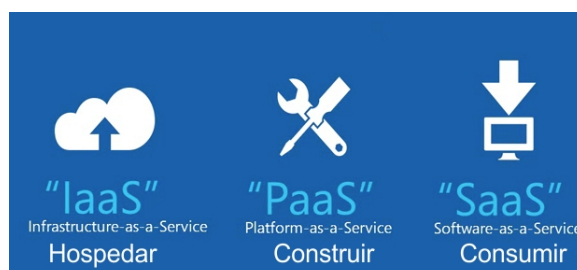


Figura 2: Descrição para cada tipo de serviço

3.2.4 X como Serviço (XaaS)

Segundo NISBET (2011) XaaS vem da tipologia, da essência do significado de *Cloud Computing*, diz ser um termo dito para os inúmeros tipos de serviços, *Anything as a Service*, seriam a solução criada a partir da flexibilidade dos serviços pela internet, isso inclui uma completo serviço, e não apenas um produto, e com base na melhoria continua, e também no total aproveitamento. O XaaS (anything as a service: SaaS,IaaS,PaaS) poderá reduzir a gigante infraestrutura (mão de obra, máquinas, salas, equipamentos...).

O XaaS disponibiliza o software e a infraestrutura na nuvem, assim quanto mais a empresa precisa mais são disponibilizados recursos, em contra partida mais será cobrado, e quanto menos precisa, menos será cobrado, assim como nas contas de luz e água.

Também sendo uma referência aos crescente números de serviços que são entregues através da internet, ao invés de providenciados localmente. Há uma lista de diferentes modelos de Serviço de computação em nuvem oferecidos no mercado da TI, BLOOR (2008) exemplifica alguns serviços:

- Communications as a Service (CaaS) - Serviço de Comunicação
- Network as a Service (NaaS) - Serviço de Rede
- Monitoring as a Service (MaaS) - Serviço de Monitoramento
- Storage as a Service (SaaS) - Serviço de armazenamento
- Backup as a Service (BaaS) - Serviço de cópias de segurança
- Database as a Service (DaaS) - Serviço de banco de dados
- Information as a Service (IaaS) - Serviço de informação
- Process as a Service (PRaaS) - Serviço de processamento
- Integration as a Service (InaaS) - Serviço de integração
- BPM as a Service - Serviço de gerenciamento de processos de negócio
- Security as a Service (SaaS) - Serviço de segurança
- Management/governance as a Service - Serviço de gerenciamento e governança
- Testing as a Service (TaaS) - Serviço de testes

Um usuário pode adotar um ou mais destas estratégias de acordo com suas necessidades. A decisão poderá ser baseado por onde concentrar seus maiores esforços: nas aplicações, plataformas e / ou infra-estrutura tecnológica, e quais os itens que irá agregar maior valor ao seu negócio.

3.3 Modelos de Implementação

Segundo MELL. P (2009), o modelo de *Cloud Computing* possui quatro modelos básicos de implementação:

- *Public Cloud* - Nuvem Pública
- *Private Cloud* - Nuvem Particular
- *Hybrid Cloud* - Nuvem Híbrida
- *Community Cloud* - Nuvem Comunitária

3.3.1 *Public Cloud* - Nuvem Pública

A infraestrutura compartilhada por várias empresas/clientes, na qual é disponibilizada ao público em geral, característica que podemos ver na FIGURA 3, e sendo acessível a qualquer usuário que tenha a localização.

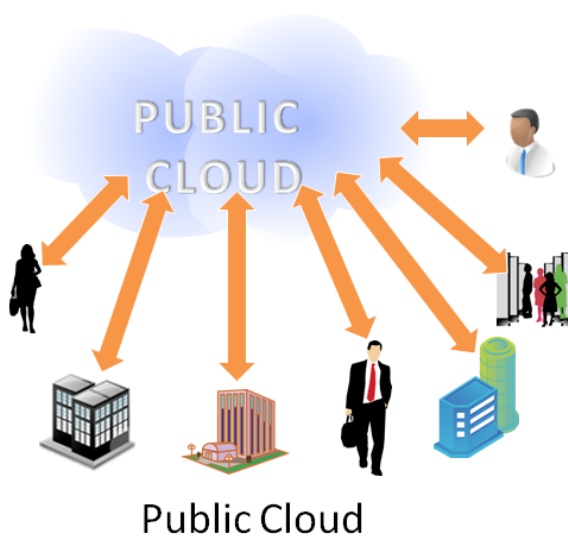


Figura 3: Exemplo de uma Nuvem Pública

3.3.2 *Private Cloud* - Nuvem Particular

A imagem da FIGURA 4 mostra como a infraestrutura é utilizada exclusivamente para uma empresa/cliente, aonde o mesmo administra os recursos disponíveis, localmente ou remotamente.



Figura 4: Exemplo de uma Nuvem Particular

3.3.3 *Hybrid Cloud* - Nuvem Híbrida

A infraestrutura é composta por mais de uma Nuvem (Particular, Comunitária, ou Pública).

3.3.4 *Community Cloud* - Nuvem Comunitária

A Infraestrutura é compartilhada por diversas organizações, ilustrada na FIGURA 5, formando assim um grupo composto para formar uma comunidade com características em comuns, ou mesmo compartilhando interesses, como por exemplo: a missão da organização, um requisitos de segurança, políticas organizacional, entre outros.

3.4 **Gestão da Tecnologia com Computação em Nuvens**

Sem dúvidas a Computação nas nuvens realmente aparece fortemente para revolucionar a infraestrutura da TI, com características que estimulam cada vez mais o número de usuários que estão migrando seus hábitos tecnológicos, estruturas e dados de desktops para as nuvens, contam então, com aplicações web para trabalhar, armazenar e acessar informações, documentos, fotos e e-mail. Com essas novas fronteiras, há novos riscos, reconhecendo ainda que a falta de maturidade no uso dessa nova tecnologia gera uma grande incerteza entre os benefícios e riscos, sendo assim, responsável pela lentidão do mercado em absorver esta nova estratégia.



Figura 5: Exemplo de uma Nuvem Comunitária

De acordo com levantamento da STINCHCOMBE (2010), medidor dos Riscos X Benefícios na Tecnologia da informação, EMEA¹, em relação ao futuro da computação em nuvem:

Constatou ainda que quase dois terços (63% das organizações), afirmaram que estão dispostas a correr riscos relacionados aos negócios de TI na antecipação do retorno de investimento² para o negócio (64,3% Reino Unido), e 12,1% correriam grandes riscos para maximizar o retorno de investimento. Quando perguntado sobre como integrar o gerenciamento de risco com a estratégia global da organização para gestão de risco:

* 4,8% admitiram fazer sem uma abordagem formal para gestão de negócios de risco (3,2% do Reino Unido).

* 22,2% disseram não integrar efetivamente o gerenciamento de risco com a sua abordagem global à gestão de riscos (22% do Reino Unido).

* 24% disseram que são muito eficazes na gestão de risco (20% do Reino Unido).

* 48,7% relataram ter sido pouco eficaz (54% do Reino Unido).

STINCHCOMBE (2010)

¹Europe, the Middle East and Africa

²Tradução de ROI - Return on investment

3.5 Os Novos Riscos Envolvidos

Com o aumento de usuários que estão migrando para esta nova estrutura e armazenado na web seus dados, hospedando processo de desenvolvimento de sistemas, serviços de e-mail, entre outros, acaba criando uma melhor maneira de acessar todas as informações, porém há alguns grandes e importantes riscos que acompanham esta nova tecnologia. RITTINGHOUSE e RANSOME (2010) afirmam que juntamente com suas novas características e benefícios, é necessário avaliar e determinar uma avaliação dos riscos com relação aos principais pontos como a integridades dos dados, confiabilidade, privacidade, processos legais, processo de recuperação pós desastres e auditorias. Um dos grandes problemas encontrados residem no próprio objetivo dessa estrutura, o acesso dos seus dados através da internet, ou de uma rede remota, ou principalmente, servidores que são acessados via internet, nestes cenários, informações confidenciais pertinentes ao negócio ou dados privados estão à mercê de crackers, ou mesmo interrupções na rede e demais ameaças de segurança que possa existir.

ENISA (2010) apresenta uma lista dos riscos em potenciais:

Perda de Governança :

No uso de *Cloud* como infra-estrutura, o cliente, necessariamente, cede controle para o provedor do serviço em inúmeras questões que possam afetar a segurança, e ao mesmo tempo, as SLA's podem não oferecer um compromisso, ou alcançar níveis desejado na prestação de tais serviços por parte do provedor, deixando uma lacuna na segurança.

***Lock-In* :**

Ainda há pouco para oferecer em termos de ferramentas, procedimentos ou formatos de dados padrão ou serviços de interfaces que poderiam garantir a portabilidade da aplicação, dados ou serviço. E isso pode dificultar a transição do cliente ao migrar de um fornecedor para outro, ou migrar dados e serviços de volta, localmente, em um ambiente de TI. Possuindo uma dependência de um provedor em particular para prestação de serviços, especialmente para a portabilidade de dados.

Widjaya (2011); Hinkle (2010) consideram três tipos de problemas com *LOCK-IN*:

1. Plataforma

Os provedores podem levantar os serviços em certas plataformas de virtualização (Ex: VMware e Xen.). Assim permitindo utilizar recursos a partir de uma administração centralizada, sem a necessidade de administrar qualquer software/hardware. A migração de um provedor que utiliza, por exemplo, Xen para outro provedor que utiliza VMware é muito problemático, devido às plataformas de virtualização serem diferentes.

2. Dados

Usando uma solução em nuvem, os dados estão completamente nas nuvens, porém ao decidir migrar para um outro provedor deverá analisar e projetar, se poderá recuperar todos os dados, manipulando e transferindo completamente os dados de volta? E neste caso verificando, se possui garantia que poderá de fato fazer-lo? As leis aonde estes dados estão contidos lhe permite tal ação? É necessário lembrar que os dados estarão no banco de dados do fornecedor e você deve estar completamente seguro que você continua dono destes dados.

3. Ferramentas

Provedores oferecem tipos diferentes de ferramentas. É necessário evitar este tipo de *lock-in*, aonde preciso ter certeza que haja um monitoramento de ferramentas que são compatíveis com os dois tipos diferentes de infra-estrutura; física e virtual.

Isolamento de falhas :

Recursos compartilhados são características que definem a computação em nuvem. Esta categoria de risco abrange as falhas dos mecanismos de memória, de armazenamento, roteamento e demais componentes.

Risco de Não-Conformidade :

Investimento em obter a certificação (ISO, padrões da indústria) pode ser posta em risco pela migração para a nuvem:

- Se o provedor não pode fornecer provas da sua própria conformidade com os requisitos relevantes
- Se o provedor não permite auditoria por parte do cliente em nuvem .

Em certos casos, isso também significa que o uso de uma infra-estrutura de nuvem pública implica que certos tipos de SLA's não pode ser alcançado.

Segurança :

É possível encontrar várias vulnerabilidades:

-
-

4 TESTE ESTRUTURADO

4.1 Característica do Teste Estruturado

Os benefícios e as preocupações de uma solução de *Cloud Computing*, torna-se muito importante durante o processo de teste, saber definir quais os testes mais apropriados, e como é possível maximizar o potencial desse investimento, é crucial para garantir a qualidade do serviço. BLACK (2009) afirma que a importância do Processo de teste de software e os impactos referentes à qualidade de Software não podem ser subestimados ou esquecidos, e assim deixando-o como um processo a parte do projeto, isso porque há inúmeros aspectos que o teste apresenta e identifica para poder cobrir riscos como: as variações de hardware ou plataforma que podem ser envolvidos, ou os números de possibilidades de cenários a serem testados tendendo a uma quantidade muito alta, e principalmente ainda não é possível projetar ou identificar as ações ao utilizar o software ou as possíveis inconsistências que poderão acontecer. Há muitas possibilidades e diversas estratégias de testes possíveis, (KANER, BACH e PEETTICHORD, 2011) afirma que a estratégia de teste é um conjunto de decisões que podem ser escolhidas, e que cada procedimento leva a um alternativa diferente, resumindo, uma boa estratégia de teste possui uma história convincente que explica, e ao mesmo tempo, justifica claramente como deverá ser executado os testes. A evolução do Teste criou uma variedade de novos conceitos, nomenclaturas, entre outras características que são aceitáveis para todos, POL, TEUNISSEN e VEENENDAAL (2011), pois hoje em uma empresa de *Software*, podemos ver em etapas aonde são discutidos requisitos ou arquitetura, há um grande entendimento sobre teste unitário, testes de aceitação, teste de sistema ou teste de desempenho, e todos conseguem entender tais termos. Com os padrões e modelos de Maturidade para o Desenvolvimento de Software, algumas das vantagens para o processo de teste é a possibilidade de compartilhamento da capacidade, o entendimento desta área para as demais, viabilizando uma melhor e mais clara comunicação.

4.2 Princípios e Principais conceitos em teste de Software

Segundo ISTQB (2007) um processo de teste essencialmente é construindo com base nas seguintes atividades:

- Planejando e controle
- Análise e Design
- Implementação e execução
- Avaliação de critérios de saída e relatórios

- Atividades de fechamento de teste

Estas atividades podem ser implementadas consecutivamente ou alguns podem ser executados em paralelo, por exemplo, análise e design poderiam ser implementados em paralelo com Implementação e execução.

Segundo ISTQB (2007), existem muitos modos diferentes e métodos disponíveis para melhorar software e sistemas, estes métodos tem como objetivo melhorar o processo (e consequentemente os artefatos entregáveis) através de algumas diretrizes e áreas de melhoria. O Processo de Teste freqüentemente responde por uma parte principal dos custos total do projeto, porém, a atenção e importância são bastante limitadas em relação ao processo de desenvolvimento. Gerenciar um projeto de teste, estimativas de esforço, alocação correta do time e tarefas de teste contínuas, com uma equipe time de teste é algo desafiador, mas há várias fontes de inspiração, idéias, e bastante informações para apoio para tal trabalho. Vários padrões podem ser aplicados para que o teste se torne algo efetivo, BLACK (2009) afirma que padrões podem variar de acordo com o segmento do mercado, importância e o risco da segurança ou mesmo políticas externas. É possível, muitas vezes, identificar padrões como um fardo burocrático, porém com eles é possível identificar e eliminar muitas das pressuposições das tarefas. Também é necessário lembrar, KANER; BACH; PEETTICHORD (2011), nenhum dos modelos pode capturar toda a complexidade do processo de software, restrições e premissas têm de ser feitas para o processo, portanto, não existe um modelo único que possa ser utilizado em todas as situações. Logo, nenhum modelo é completo, um modelo pode funcionar bem para um conjunto de características de uma forma melhor, mas pode ser completamente fora do contexto para outros tipos de problemas. É importante selecionar e adaptar as normas e padrões de forma adequada, certificando-se de entender o contexto e a origem dos padrões que você está considerando, bem como a correta aplicação da norma.

Importantes organizações internacionais de normalização como ISO e IEEE divulgam normas de interesse para processo de teste. Podemos listar as normas relacionadas à ISO - *International Organization for Standardization*:

- ISO 9126 - Cobre aspectos de qualidade de produto da engenharia de software. Consiste em quatro normas e relatórios técnicos. ISO - International Organization For Standardization (2011a) aborda um modelo de qualidade, incluindo definição de características e subcaracterísticas que podem ser aplicadas a qualquer produto de software. ISO - International Organization For Standardization (2011b) abrange métricas de qualidade externa, como métricas realizadas durante o teste dinâmico(*Dynamic Testing*¹).

ISO - International Organization For Standardization (2011c) abrange métricas de qualidade internas, como métricas realizadas durante o teste estático(*Static Tes-*

¹*Dynamic Testing* - Teste que envolve a execução do software, de um componente ou de um sistema

*ting*²). ISO - International Organization For Standardization (2011d) define métricas para a avaliação das características de qualidade em uso.

- ISO - International Organization For Standardization (2008) - Cobre o processo de ciclo de vida do *software*.
- (ISO - International Organization For Standardization, 2004) - Cobre a avaliação do processo e também é conhecida como *SPACE*. É similar ao CMM (*Capability Maturity Model*) e ao CMMI (*Capability Maturity Model Integration*).

e podemos identificar e listar as normas apresentadas pelo IEEE - *Institute of Electrical and Electronics Engineers*:

- IEEE Computer Society (2001a), é o glossário do *IEEE standard computer dictionary*, contendo todo o conjunto de definições e conceitos das normas.
- IEEE Computer Society (2001b) é a norma para documentação de teste de software.
- IEEE Computer Society (2001c) é a norma para revisão de software.
- IEEE Computer Society (2001d) é o guia para classificação de inconformidades de software.

Baseado nas idéias em comum de KANER; BACH; PEETTICHORD (2011); GODBOLE (2004); BLACK (2007) uma premissa importante para melhoria de processo é a entender que a qualidade de um sistema/software é altamente influenciada pela qualidade do processo de desenvolvimento. A Melhoria da qualidade no mercado, reduz a necessidade por recursos para mantê-lo o software e também disponibiliza mais tempo por criar mais soluções, e com melhor qualidade FELIPE DIAS (2008). Melhorias de processo são relacionadas ao processo de desenvolvimento de software mas também para processo de Teste, e aprendendo com próprios erros, torna-se possível melhorar os processos das quais a organizações utiliza para desenvolver e testar o *software*.

4.3 Modelos de processo de teste

A indústria de TI começou a avaliar o uso dos modelos de melhoria de teste visando alcançar um nível maior de maturidade e profissionalismo, (SOFTEX, 2010). Modelos e padrões da indústria estão cada vez mais ajudando o desenvolvimento dos resultados cruzados de métricas e medições das organizações, para que seja possível realizar uma comparação. Dessa necessidade de melhoria de processos na indústria de testes, vários padrões e modelos surgiram, exemplos incluem: STEP, TMMi, TPI e CTP. Modelos em

²*Static Testing* - Teste de um componente ou sistema em um nível de especificação ou implementação sem a execução do *software*, Ex: Revisão, Análise Estática.

estágio como CMMI e TMMI apresenta padrões para apresentar métricas afim de realizar para comparações de resultados entre diferentes organizações e empresas. Os modelos contínuos, permitem a organização endereçar seus problemas de maior prioridade com maior flexibilidade e liberdade na implementação.

Modelos de processo lhe permite ter um ponto de partida para as melhorias, medindo os processos da organização com o modelo. O modelo também pode ser visto como uma framework para melhoria dos processos baseando-se no resultado de uma avaliação. A avaliação de processos leva a uma determinação capacidade do Processo, o que motiva a melhoria, e também é possível utilizar posteriormente uma nova avaliação de processos para medir o efeito da melhoria. Há vários modelos de maturidade Teste e de Melhoria do Processo de Teste reconhecidos no mercado - (ISTQB, 2007; BLACK, 2009):

- Systematic Test and Evaluation Process (STEP)
- Test Maturity Model (TMM),
- Test Process Improvement (TPI)
- Practical Software Testing
- Critical Testing Processes
- Test Maturity Model (TMM)
- Test Maturity Model integration (TMMi)
- Melhoria de Processo de Teste - MPT.BR
- Testability Support Model (TSM)
- Test Organization Maturity (TOMtm)
- Testing Assessment Program (TAP)
- Testing Improvement Model (TIM)

Os modelos apresentados nesta seção serão utilizados para mostrar e proporcionar uma visão representativa do modo como funcionam os modelos e o que está incluído dentro deles.

4.3.1 MPT.BR - Melhoria do Processo de Teste

O MPT é modelo brasileiro voltado para a melhoria das áreas de teste de software de empresas de qualquer porte. Sendo um modelo para Melhoria do Processo de Teste criado para apoiar as organizações de software através dos elementos essenciais para o

desenvolvimento do processo de teste inserida no processo de desenvolvimento de software, (MPT.BR, 2010). Há duas entidades trabalhando em conjunto no desenvolvimento deste modelo:

- ALATS - Associação Latino Americana de Teste de Software
- RIOSOFT - Sociedade Núcleo de Apoio à Produção e a Exportação de Software

Segundo (ALATS, 2002; RIOSOFT, 1990) este modelo é baseado nas melhores práticas do teste de software e promove a integração das atividades de engenharia software, otimizando os processos de teste para alcançar os resultados desejados através da entrega de produtos com qualidade. Sendo passível de ser adotado pela áreas de teste de software para identificar e melhorar seu nível de maturidade, sem afetar seus processos anteriormente implementados. O objetivo principal será garantir que áreas de teste de software de tamanho reduzido possam ser avaliadas e estimuladas a alcançarem níveis maiores de maturidade.

Podemos citar os principais objetivos deste modelo:

- Aumentar a qualidade dos produtos de software através da otimização e melhoria contínua dos processos de teste.
- Fornecer visibilidade relativa a maturidade do processo de teste de uma organização para o mercado de software.
- Fomentar a melhoria contínua dos processos de teste no âmbito do desenvolvimento de software.

Os critérios usados pelas entidades na estruturação do modelo foram:

- Definir critérios que garantam a qualidade do processo de teste de software
- Não criar nada novo, mas usar como base um modelo de melhoria do processo de desenvolvimento de software já existente no mercado no caso o MPS.BR, ou CMMI, FIGURA 6.
- Poder ser usado por áreas de teste de software de empresas pequenas, médias e grandes
- Permitir a evolução das áreas de testes atingindo níveis mais altos de maturidade e, conseqüentemente, níveis mais altos de qualidade

4.3.1.1 Estrutura do modelo MPT

Embora exista uma alta coesão entre o CMMI e o MPS.BR os seus níveis não necessariamente se correspondem, como mostra a TABELA 2:

Podemos ver na TABELA ?? todos níveis de maturidade definidos no modelo:

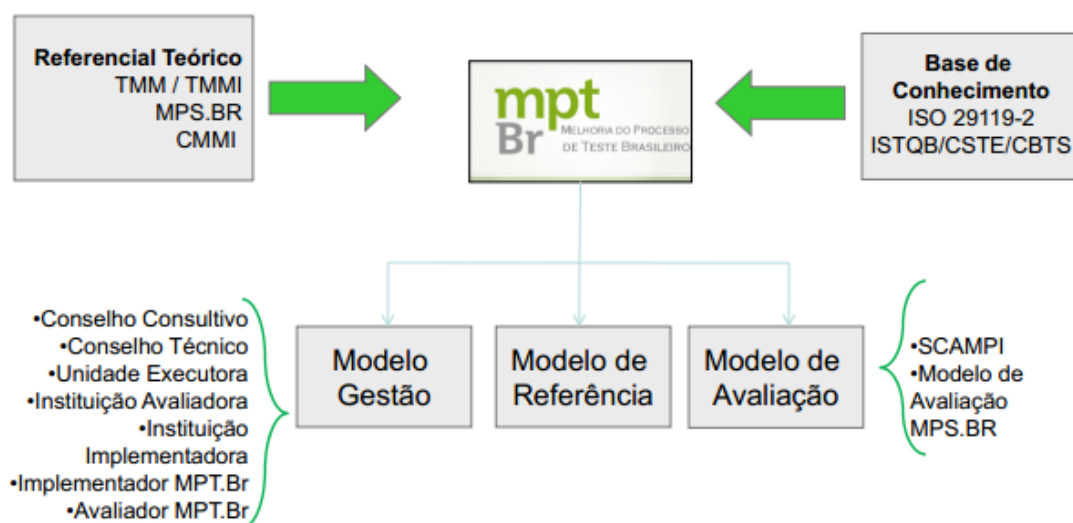


Figura 6: Base de conhecimento para a criação do Modelo

Tabela 2: MPT x MPS x CMMI

<i>MPT</i>	<i>MPS</i>	<i>CMMI</i>
Nível 1	Sem correspondência	Sem correspondência
Nível 2	Nível G	Level 2
Nível 3	Nível F	Sem correspondência
Nível 4	Nível E	Sem correspondência
Nível 5	Nível D	Sem correspondência
Nível 6	Nível C	Level 3
Nível 7	Nível A	Level 4,5

4.3.1.2 Níveis de maturidade e as áreas de processo do modelo MPT

Atualmente é apresentado dois níveis de maturidade:

1. **Parcialmente gerenciado** - este nível representa o primeiro patamar de maturidade de uma organização. Ele contém o mínimo que uma organização precisa para demonstrar que a disciplina de teste é aplicada nos projetos, e também que ocorre de forma planejada e controlada.
2. **Gerenciado** - neste segundo nível a aplicação do processo de teste na organização possui maior visibilidade. O escopo do projeto passa a ser controlado pelo processo de gestão de mudanças, padrões são definidos e os processos são monitorados e controlados.

Cada nível de maturidade é composto por um conjunto de áreas de processo, e cada área de processo é um conjunto de práticas relacionadas, e quando implementadas deve satisfazer um determinado objetivo. Cada nível de maturidade também é associado a

um conjunto de práticas genéricas que devem ser implementadas a cada área de processo dentro do nível de maturidade.

Para uma organização atender a um determinado nível de maturidade, ela deve demonstrar através de uma avaliação que o processo de teste aplicado em seus projetos atende a todas as áreas de processo e que todos os níveis anteriores de maturidade também continuam sendo atendidos.

4.3.1.3 Áreas de processo do modelo MPT.Br

A organização das áreas de processo do MPT.Br é exibida na TABELA 3.

Tabela 3: Organização das áreas de processo do MPT.Br

Nível de Maturidade	Áreas de Processo	Práticas Genéricas
Nível 1	GPT - Gerência de Projetos de Teste (Práticas Específicas GPT1 a GPT20) PET - Projeto e Execução de Teste (Práticas Específicas PET1 a PET4)	PG1 a PG6
Nível 2	GRT - Gerência de Requisitos de Teste (Práticas Específicas GRT1 a GRT5) GPT - Gerência de Projetos de Teste (Práticas Específicas GPT21 a GPT25) PET - Projeto e Execução de Teste (Prática Específica PET5)	PG7 a PG9

4.3.2 TMMI - *Test Maturity Model integration*

O TMMI, *Test Maturity Model integration*, em sua estrutura apresentada um *framework*, que consiste em diversos elementos: níveis de maturidade, áreas chaves do processo, objetivos comuns, práticas chaves, sub-práticas, exemplos e literatura recomendada - Foundation (2009).

Podemos ver na FIGURA 7 os níveis e suas práticas chaves relacionadas.

4.3.2.1 Nível 1 - *Initial*(Inicial)

TMMi reconhece neste nível o teste como caótico com processo de teste não definido, sendo considerado uma parte da atividade de *Debug* da aplicação na organização. O Sucesso do projeto, frequentemente é alcançada através feitos heróicos, ou mesmo, pelo cliente aceitar riscos da aplicação. A Organização tem uma forte dependência do alto comprometimento da equipe, não execução do processo em momentos críticos, e a incapacidade

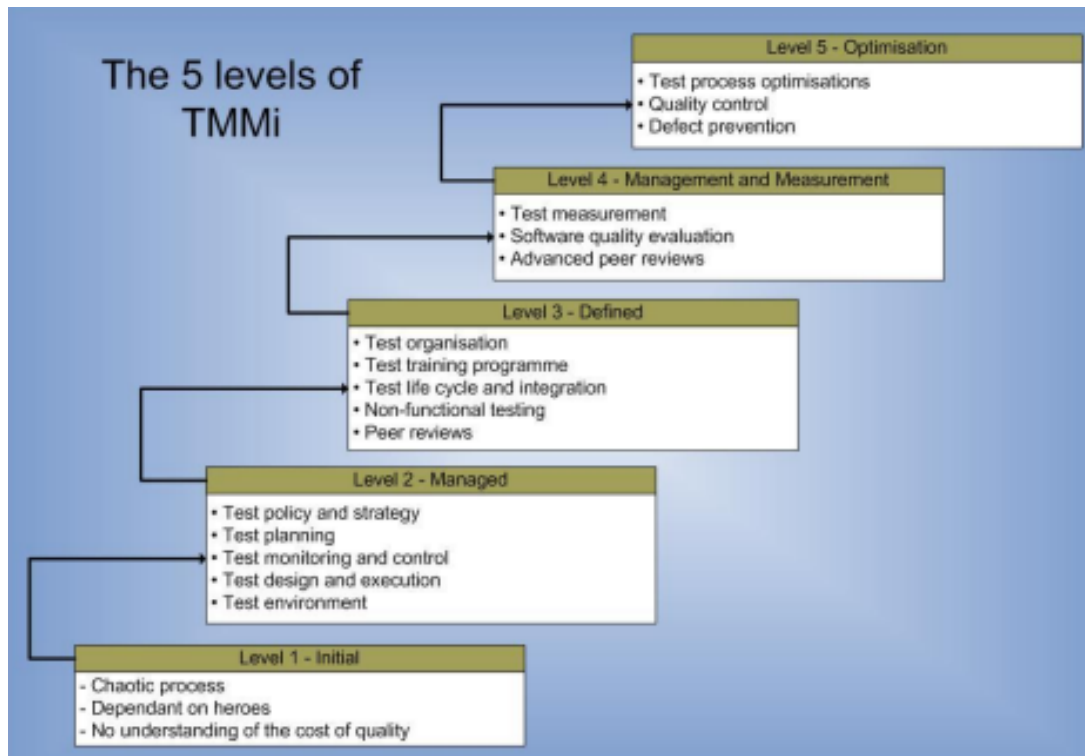


Figura 7: Níveis de maturidade de seus processos chaves

de repetir as práticas para alcançar o sucesso. Não há processos chaves envolvidos neste nível.

4.3.2.2 Nível 2 - *Managed* (Gerenciado)

O teste é um processo gerenciado, e neste nível é possível garantir que as atuais práticas existentes são mantidas em momentos crítico e de alto stress. O principal objetivo do teste é a verificação, assim satisfazendo os requisitos. o processo de teste ainda é visto na organização pelos *stakeholders* como uma fase do projeto que segue logo após a codificação.

Neste nível reconhecido vários níveis de aplicação do teste, na qual variam de teste unitário ao teste de aceitação. Para cada nível de teste identificados existem objetivos específicos definidos na estratégia de teste de toda a organização, ou o na estratégia do programa.

4.3.2.3 Nível 3 - *Defined* (Definido)

Neste nível a organização compreende a importância das revisões no controle da qualidade, e consegue implementar um programa de revisão formal referente ao processo de teste dinâmico. O teste é totalmente integrado no ciclo de desenvolvimento e as há também metas associadas. Neste nível a melhoria de processo de teste é totalmente institucionalizada como uma parte das práticas da organização, sendo identificado como um

papel com responsabilidades distintas.

4.3.2.4 Nível 4 - *Measured* (Medido)

Neste nível Os testes são completamente definidos, bem fundamentados e medidos. Revisões e inspeções são incorporadas ao ciclo de vida do desenvolvimento e considerados parte dos testes. Os produtos de software são avaliados a partir de critérios de qualidade e suas características (reusabilidade, usabilidade e manutenibilidade). Casos de testes são armazenados e gerenciados em uma base de dados central para reuso e testes de regressão.

Com relação à qualidade do produto, um programa de medição permite que uma organização possa implementar um processo de avaliação da qualidade do produto através da definição de necessidades de qualidade, atributos de qualidade e métricas. Os produtos relacionados com o trabalho são avaliados por critérios quantitativos para atributos de qualidade como usabilidade, confiabilidade e facilidade de manutenção.

Nível 4 abrange também estabelecer uma abordagem de teste coordenada entre revisões por pares de testes estáticos e dinâmicos de testes eo uso dos resultados de avaliações pelos pares e dados para otimizar a abordagem de teste com ambos destinados a tornar os testes mais eficazes e mais eficientes. Revisões por pares são diretamente integrados com o processo de ensaio dinâmico e é uma parte da estratégia de teste, plano de teste e abordagem de teste.

4.3.2.5 Nível 5 - *Optimization* (Otimizado)

Ao alcançar o nível 5, a organização é capaz de implantar uma melhoria continua, aperfeiçoando seus processos com base em entendimento quantitativo dos dados estatísticos dos processos. Melhorar o desempenho do processo de teste é realizado através de um processo incremental e melhorias tecnológicas. Os métodos de teste e técnicas são otimizadas e existe uma preocupação contínua com o ajuste refinado e melhoria de processos. A área de processo de prevenção de defeito é estabelecida para identificar e analisar as causas comuns dos defeitos em todo o desenvolvimento do ciclo de vida e definir ações para preveni que eventos semelhantes ocorram no futuro.

4.3.3 TPI NEXT - *Test Process Improvement*

O modelo TPI (*Test Process Improvement*) foi baseado nas melhores praticas para o desenvolvimento do processo de teste, oferecendo uma ponto de vista da maturidade do processo de teste de dentro da organização. Este modelo ajuda a definir gradualmente e de modo controlado a melhoraria dos processos de de teste baseado em alguns aspectos.

Esses aspectos são chamados de áreas chaves, este modelo descreve 16 áreas chaves³:

1. Gestão de *Stakeholder*
2. Grau de envolvimento
3. Estratégia de teste
4. Organização do teste
5. Comunicação
6. Relatórios
7. Gerenciamento de processo de teste
8. Estimativa e planejamento
9. Métricas
10. Gerenciamento de defeitos
11. Gerenciamento de *Testware*
12. Metodologia prática
13. Profissionalismo do Tester
14. Design de caso de teste
15. Ferramentas de teste
16. Ambiente de teste

4.3.3.1 Níveis de Maturidade de teste

O modelo TPI, usa a Matrix de Maturidade para gerar um *overview* das áreas chaves de teste e suas respectivas maturidades. A matriz de maturidade lista todas as 16 áreas de processo de teste, e então mostra o nível global da maturidade baseado em níveis distintos:

- Inicial
- Controlado
- Eficiente

³Traduzidas do inglês.

- Otimizado

O objetivo principal da matriz é mostrar os pontos fortes e fracos do atual processo de teste e apoiar a elaboração de propostas de melhoria, e facilitar a identificação de prioridade para ações de melhoria. A situação atual do processo de teste pode ser visto claramente através da matriz.

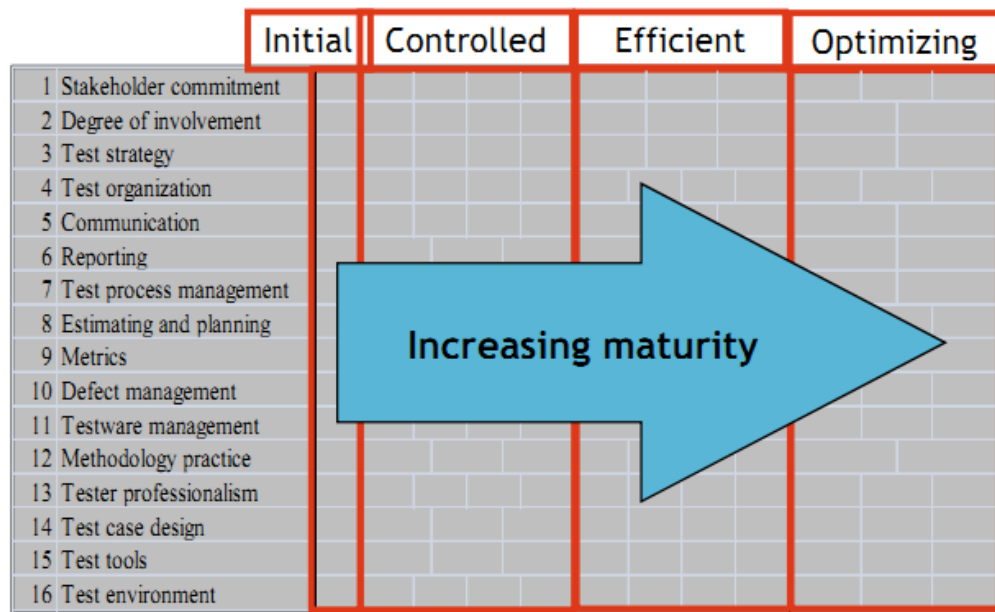


Figura 8: Contrução da Matrix de Maturidade

As aéreas chaves podem ser avaliadas entre conceitos A e D, sendo o A o mais baixo, para que se encaixem em um conceito, é necessário elaborar questionamentos para avaliação, com base em alguns critério e conceitos definidos no modelo.

As combinações das áreas chaves, níveis de maturidade e *checkpoint* são utilizados para identificar os pontos fortes e fracos do atual processo de teste, e então auxiliar na definição plano de ação para melhoria.

Completando a matriz torna-se fácil de identificar e avaliar propostas de melhorias. Os requisitos para cada níveis são definidos no forma de *checkpoint*, que são questões necessariamente precisam ser respondidas, para que seja possível então, qualificar quanto ao nível avaliado. Esses *checkpoint* permitem que seja possível classificar os objetivos por níveis.

Podem ser definidas ações de melhoria focadas apenas em níveis mais altos, *checkpoints* ajudam a determine quais ações são necessárias para alcançar o próximo nível, podemos ver um exemplo de Matriz de Maturidade completa na FIGURA 9.

Um das vantagens, as sugestões de melhoria que este modelo oferece para apoiar melhoria de processo de teste inclui uma lista de dicas e idéias que podem ajudar a alcançá-lo.

	Initial	Controlled				Efficient				Optimizing		
1 Stakeholder commitment		A	B	B	C	F	H	H		K	M	M
2 Degree of involvement		A	B	C	E	H	H	J		L		L
3 Test strategy		A	A	B	E	F	F	H		K		L
4 Test organization		A	D	D	E	I	I	J	J	K	L	L
5 Communication		B	C	C	D	F	F	J		M		M
6 Reporting		A		C	C	F	G	G		K		K
7 Test process management		A	A	B	B	G	H	J		K		M
8 Estimating and planning		B	B	C	C	G	H	I	I	K	L	L
9 Metrics		C		C	D	G	H	H	I	K		K
10 Defect management		A	A	B	D	F	F	H	J	K	L	L
11 Testware management		B	B	D	E	I	I	J		L	L	L
12 Methodology practice		C		D	E	F	H	J	J	M		M
13 Tester professionalism		D	D	E	E	G	G	I	I	K	K	M
14 Test case design		A	A		E	F	I	I	J	K	K	M
15 Test tools		E		E	E	F	G	G	I	L	M	M
16 Test environment		C	D	D	E	G	H	J	J	L	M	M

Figura 9: Matrix de Maturidade Completa

5 ESTRATÉGIA DE TESTE

Com base nos pontos apresentados até o momento, podemos identificar a necessidade de definir as atividades a serem executadas, resultados esperados e demais informações para melhor entendimento dos *stakeholders* e demais envolvidos.

Será criado um documento para detalhar o plano de ação para as atividades de engenharia de teste de software para performance e segurança, fornecendo uma explicação dos objetivos, estratégia e detalhes das tarefas referente ao esforço. O plano de teste irá conter todas as informações necessárias referente a avaliação e execução de testes de desempenho e de segurança obedecendo os seguintes itens:

1. Objetivos
2. Procedimentos
3. Escopo de teste
 - Overview
 - Objetivos
 - Estratégia de Teste
 - Casos de Teste
 - Cenários
 - SLAs
4. Missão e critérios de saída de teste
5. Artefatos de saída
6. Técnicas Utilizadas
7. Riscos
8. Ambiente de Teste

1. Proposito

- Detalhar o plano de ação para todas as atividades do teste de performance e segurança.
- Prover uma explicação das abordagens de teste definidas para o esforço de execução.

2. Procedimentos

- Identificar pontos de garantia para requisitos avaliados.
- Definir a missão da execução dos testes, (Ex: "Identificar o maior número de defeitos", "promover a integração com o sistema legado", "Identificar módulos com maior risco de invasão ou ataque").

3. Escopo de teste

- Definir o que será teste e o que não será testado

3.1 Teste de Segurança

3.2.1 Overview

- Considerar os requisitos de segurança e analisar arquitetura do software, aspectos gerais, riscos para definir a abordagem. casos de teste e técnicas necessárias para cobertura de teste.

3.2.2 Objetivos

- Acompanhar e monitorar a implementação dos controles.
- Identificar métodos de prevenção contra ataques internos e externos da rede.
- Identificar invulnerabilidades.
- Garantir um grau de segurança.

3.2.3 Estratégia de Teste

- Será utilizada técnica conhecida como "*Man-in-the-middle*", simulando uma ataque interno, para identificar as possíveis ações e requisitos necessários para prevenir e/ou interromper tal ataque.
- Será explorado e analisado as vulnerabilidades do estrutura e arquitetura, através de ataques externos.

Casos de Teste

Cenários

- Lista de cenários bases para execução dos testes de segurança :
 - Ataque Interno.
 - Ataque Externo.

3.2 Teste de Performance

3.2.1 Overview

- Considerar os requisitos de performance e analisar arquitetura do software, aspectos gerais, riscos para definir a abordagem, limites, carga de informações suportadas e as técnicas necessárias para cobertura de teste.

3.2.2 Objetivos

- Teste de Performance irá ter como objetivos esclarecer os seguintes pontos:
 - Tempo de resposta para cada transação.
 - Tempo de duração para cada carga.
 - Recursos utilizados (CPU, memória, etc...) em um cenários específico (Normal, Exaustão).
 - Tempo de exaustão para uma dado cenário.
 - Números de usuários concorrentes.

3.2.3 Estratégia de Teste

- Será utilizado ferramentas para simular interações do usuário, *Virtual Users*.
- Informações sobre tempo de respostas serão coletados através de *Logs* das aplicações.

3.2.4 Casos de Teste

- Serão definidos conforme necessidade.

3.2.5 Cenários

- Lista de cenários bases para execução dos testes de performance :
 - Cenário de Consumo Normal de Recursos.
 - Cenário de Consumo em Exaustão dos Recursos.

4. Artefatos de saída

- Plano de Tete de Performance
- Plano de Teste de Segurança

REFERÊNCIAS

ALATS. **ALATS**. Associação Latino Americana de Teste de Software, www.alats.org.br.

BLACK, R. **Pragmatic Software Testing**: becoming an effective and efficient test professional. 1 edition. ed. [S.l.]: Wiley, 2007. (978-0470127902).

BLACK, R. **Advanced Software Testing - Vol. 2**: guide to the istqb advanced certification as an advanced test manager. 1 edition. ed. [S.l.]: Rocky Nook, 2009. (978-1933952369).

BLOOR, R. **Everything as a Service**: the growth of cloud computing.
<http://www.havemacwillblog.com/2008/06/everything-as-a-service-the-growth-of-cloud->

CZERNICKI, B. **IaaS, PaaS and SaaS Terms Clearly Explained and Defined**.
<http://www.silverlighthack.com/post/2011/02/27/IaaS-PaaS-and-SaaS-Terms-Explained-a->
www.silverlighthack.com.

ENISA. **Cloud Computing - Benefits, risks and recommendations for information security**. <http://www.enisa.europa.eu/>.

FELIPE DIAS, A. **Por que investir em melhoria de processos?**
http://pronus.eng.br/artigos_tutoriais/processo_desenvolvimento/melhoria_processo.p
PRONUS® Engenharia de Software.

FIGUEIREDO, A. **Cloud computing já é usado em 18% das companhias**.
<http://www.ceti.org.br/web/noticias.php?id=126>, Disponível em www.ceti.org.br.

FOUNDATION, T. **Test Maturity Model Integrated (TMMi)**. 2009.

GODBOLE, N. S. **Software Quality Assurance**: principles and practice. [S.l.]: Alpha Science Intl Ltd, 2004. (978-1842651766).

HINKLE, M. **Three Cloud Lock-in Considerations**. 2010.

IEEE610-1990. **610-1990 - IEEE Standard Computer Dictionary**: compilation of iee standard computer glossaries. [S.l.: s.n.], 2001.

IEEE829-2008. **829-2008 - IEEE Standard for Software and System Test Documentation**. [S.l.: s.n.], 2001.

IEE1028-2008. **1028-2008 - IEEE Standard for Software Review and Audits**. [S.l.: s.n.], 2001.

IEE1044-2009. **IEE1044-2009 - IEEE Standard Classification for Software Anomalies**. [S.l.: s.n.], 2001.

ISO/15504. **ISO/IEC TS 15504:2004 information technology – process assessment – part1-9**. 1 (Monolingual). ed. [S.l.: s.n.], 2004.

ISO/IEC9126:1. **ISO/IEC 9126-1:2001 software engineering - product quality - part 1: quality model**. 1 (Monolingual). ed. [S.l.: s.n.], 2011.

ISO/IEC9126:2. **ISO/IEC 9126-2:2003** software engineering - product quality - part 2: external metrics. 1 (Monolingual). ed. [S.l.: s.n.], 2011.

ISO/IEC9126:3. **ISO/IEC 9126-3:2003** software engineering - product quality - part 3: internal metrics. 1 (Monolingual). ed. [S.l.: s.n.], 2011.

ISO/IEC9126:4. **ISO/IEC 9126-4:2004** software engineering - product quality - part 4: quality in use metrics. 1 (Monolingual). ed. [S.l.: s.n.], 2011.

ISO/IEC12207. **ISO/IEC 12207:2008** system and software engineering - software life cycle processes. 2 (Monolingual). ed. [S.l.: s.n.], 2008.

BOARD, I. S. T. Q. **Advanced Level Syllabus**. Version 2007. ed. [S.l.: s.n.], 2007. Advanced Level Working Party: Bernard Homès (chair), Graham Bath, Rex Black, Sigrid Eldh, Jayapradeep Jiothis, Paul Jorgensen, Vipul Kocher, Judy McKay, Klaus Olsen, Randy Rice, Jürgen Richter, Eric Riou Du Cosquer, Mike Smith, Geoff Thompson, Erik Van Veenendaal;

KANER, C.; BACH, J.; PEETTICHORD, B. **Lessons Learned in Software Testing: a context-driven approach**. [S.l.]: John Wiley & Sons, 2011.

KNORR, E.; GRUMAN, G. **What cloud computing really means**. <http://www.infoworld.com/d/cloud-computing/what-cloud-computing-really-means-031>, Available at infoworld.com.

MELL, P, G. T. **Draft NIST Working Definition of Cloud Computing**. <http://csrc.nist.gov/groups/SNS/cloud-computing>, National Institute of Standards and Technology.

MILLER, M. **Cloud Computing: web-based applications that change the way you work and collaborate on-line**. 1 edition. ed. [S.l.]: Que, 2008. (0789738031).

MPT.BR. MPT Melhoria do Processo de Teste. , [S.l.], June 2010. Centro de Excelência em Tecnologia de Software.

NISBET, B. **Monetizing the Cloud: xaas opportunities for service providers**. [S.l.]: IDC Analyze the Future, 2011.

POL, M.; TEUNISSEN, R.; VEENENDAAL, E. V. **Software Testing - A guide to the TMAP Approach**. 1 edition. ed. [S.l.]: Addison-Wesley, 2011. (0201745712).

REESE, G. **Cloud Application Architectures - Building Applications and Infrastructure in the Cloud**. 1 Edition. ed. [S.l.]: O'Reilly Media, INC, 2009. (987-0-596-15636-7).

RIOSOFTE. **RIOSOFTE**. Cia. Brasileira de Software e Serviços Ltda.

RITTINGHOUSE, J.; RANSOME, J. **Cloud Computing: implementation, management, and security**. 1 edition. ed. [S.l.]: CRC Press, 2010. n. 978-4398-0680-7. (0789738031).

SOFTEX. **Softex**. Centro de Excelência em Tecnologia de Software, Softex.br.

STINCHCOMBE, N. **Study finds organisations using cloud think risks outweigh benefits.** The ISACA IT Risk/Reward Barometer survey is based on online polling in March 2010 of 1 529 IT professionals who are ISACA members in Europe, Editorial Eskenzi PR -

<http://www.itweb.co.za/office/isaca/PressRelease.php?StoryID=206464>,
<http://www.isaca.org/>.

TAURION, C. **A “nuvem” que traz uma revolução.** NEI Meetings -
http://www.nei.com.br/artigos/download/julho_2011_tecnologia_informacao.pdf,
 NEI International Industrial Conference & Show.

TAXONOMY, C. **Software as a Service (SaaS).**
<http://cloudtaxonomy.opencrowd.com/taxonomy/software-as-a-service/>,
 OpenCrowd - Cloud Taxonomy.

THINKSTRATEGIES. **The Future of IT in Large Corporations - A Whitepaper on Software as a Service.**
http://www.thinkstrategies.com/images/SaaS/_WP/_Final/_072405.pdf,
 THINKSTRATEGIES.

THIOLLENT, M. **Metodologia da Pesquisa-ação.** 1 Edition. ed. [S.l.]: São Paulo: Cortez: Autores Associados, 1986.

WIDJAYA, I. **Three Types of Cloud Lockin.** 2011.

ZOHO. **Platform as a Service.** <http://www.zoho.com/creator/paas.html>,
 ZOHOCreator.

APÊNDICE A INFORMAÇÕES COMPLEMENTARES

Informações Complementares informam e são O Apêndice A o lugar para incluir textos complementares, que não são essenciais para o entendimento do assunto principal da monografia, mas que podem contribuir com informação relevante (por exemplo, uma prova matemática, uma conceitualização básica, etc.). Ele deve seguir o formato normal do documento.

ANEXO A ARTIGOS PUBLICADOS

Existe diferença entre os Apêndices e os Anexos. Os apêndices trazem informação escrita pelo próprio autor do trabalho, incorporando-se ao formato da monografia como um todo. Já um anexo é um material à parte, definido/publicado por si só, e que o autor julga conveniente ser apresentado juntamente com a monografia. Normalmente também vai apresentar formato próprio, como um artigo publicado, um folder, uma planilha, etc.