

Profissional da Área de Testes

Conhecimentos, habilidades e atitudes requeridas para a área de Testes de Software



Daniel Scaldaferrri Lages

dilages@gmail.com

Possui MBA em Gerência de Projetos pela Fundação Getúlio Vargas, é pós-graduado em Gerência de T.I. pela Universidade FUMEC e Bacharel em Ciência da Computação pela UFMG. Atualmente é Coordenador da equipe de Quality Assurance da CPM Braxis, filial BH. Sua experiência profissional inclui o cargo de Analista de Testes no Synergia (núcleo de engenharia de software do Departamento de Ciência da Computação da UFMG), Gerente da fábrica de software na Unitech (hoje CPM Braxis) e docência no curso de graduação de Sistemas de Informação na PUC-MG. Possui a certificação ITIL para gerenciamento de serviços de T.I. É certificado em testes de software pela ISTQB e em qualidade de software pela IBM.

Quem participa de listas de discussão e fóruns na Internet sobre Testes de Software (ler **Nota 1**) sabe que é frequente o ingresso de novos integrantes solicitando orientações e conselhos para iniciarem a carreira na área. Normalmente, perguntam o que devem saber e fazer para poderem se especializar em testes e, conseqüentemente, encontrar um emprego com maior facilidade. Este artigo poderá ajudá-los no sentido de orientar sobre quais são as competências desejáveis para um profissional de testes. Dessa forma, o iniciante poderá utilizá-lo como guia para montar um plano de estudos e assim aumentar seu conhecimento. Poderá, também, aperfeiçoar-se nos aspectos comportamentais necessários, preparando-se para gerar bons resultados na sua empresa, ou mesmo para uma entrevista de emprego na área. Da mesma forma, uma empresa poderá se basear nas competências indicadas para realizar um processo de seleção para uma vaga específica de testes, ou montar uma equipe de testadores sob medida.

De que se trata o artigo?

Este artigo apresenta um conjunto de conhecimentos, habilidades e atitudes que o profissional de testes (ou o que quer iniciar sua carreira na área) deve possuir para que seja capaz de destacar-se na realização do seu trabalho. É apresentada também a estrutura de equipe mais comum encontrada na área de testes.

Para que serve?

O artigo visa orientar profissionais que desejam ingressar na área de Testes de Software, assim como orientar os profissionais que já atuam a complementarem suas competências. Serve também para empresas que desejam montar uma equipe de testes na medida certa para atender suas necessidades.

Em que situação o tema é útil?

Empresas e profissionais que possuem interesse em conhecer quais são as competências que um profissional e/ou equipe de testes de software devem possuir para obterem sucesso na função.

As competências necessárias para o desempenho das atividades da área de Testes de Software são formadas por características **comportamentais** e **técnicas**. A área de testes exige um equilíbrio maior entre essas duas características,

diferentemente das outras áreas da Engenharia de Software, como por exemplo, a área de Requisitos. Nesta área, o comportamento tende a ter maior importância frente ao conhecimento técnico, pois o profissional passará mais tempo em contato com o cliente, realizando o levantamento dos requisitos. Nos testes, apenas conhecimentos técnicos não são suficientes, pois a área envolve muitos aspectos de psicologia, como serão apresentados adiante. Da mesma forma, apenas um comportamento adequado não credencia uma pessoa a ser um bom profissional de testes. Este deve possuir grandes conhecimentos técnicos para atuar com precisão.

O termo **competência** no contexto desse artigo seguirá o conceito de [GRAMIGNA], que é definido como o “repertório de comportamentos e capacitações que algumas pessoas ou organizações dominam melhor que outras, fazendo-as eficazes em uma determinada situação”. No próximo item, o repertório para os profissionais de testes será apresentado.

O que saber e como agir?

Como dito na introdução, para que um profissional de testes possa começar a atuar, deverá possuir algumas competências técnicas e comportamentais básicas. Naturalmente, ao longo do tempo e com boa orientação, o profissional irá desenvolver essas competências. Não é aconselhável ter muita pressa e sair estudando tudo que aparece pela frente. A prática é fundamental. A experiência virá com o tempo.

Mas para que possa se destacar é preciso ter conhecimentos técnicos amplos, e alguns deles mais profundos. Assim como possuir atitudes exemplares. Segundo [LOVELAND], o profissional de testes é uma “raça especial” por uma série de

razões. Alguns traços de personalidade são necessários para que se tenha sucesso, felicidade e longa carreira na função. O “Qualquer um pode testar” é um mito. O bom testador deve atuar com precisão e segurança, falhando o mínimo possível. Deve preocupar-se ao máximo com sua **credibilidade**, que é um fator determinante para o seu sucesso. Um testador que falha sucessivamente perde a confiança dos colegas de trabalho, que passam a desconfiar da sua competência, por exemplo, duvidando das falhas registradas ou ignorando sugestões, alertas e questionamentos.

Para alcançar essa credibilidade, o testador deverá ser persistente na busca pelas competências necessárias (mas lembrando, sem correria). Para desmembrar as competências, um conceito bastante conhecido entre os profissionais da área de recursos humanos é bem útil, o CHA. Este é o acrônimo das palavras **conhecimentos, habilidades e atitudes**.

Conforme [GRAMIGNA], as competências do ser humano podem ser comparadas a uma árvore:

- A raiz corresponde às *atitudes*: A raiz corresponde ao conjunto de valores, crenças e princípios, formados ao longo da vida, e que determinam nossas atitudes. Ela é o início de tudo e o principal componente da competência. Está relacionada com o *querer ser* e o *querer agir*;
- O tronco corresponde ao *conhecimento*: É o segundo componente de uma competência. Trata-se do conjunto de informações que a pessoa armazena e lança mão quando precisa. Quanto maior este conhecimento, mais a competência se fortalece. E permite que o profissional enfrente com flexibilidade e sabedoria os diversos desafios de seu dia-a-dia;



Nota do DevMan 1

Teste de Software

Teste de software é o processo de execução de um produto para determinar se ele atingiu suas especificações e funcionou corretamente no ambiente para o qual foi projetado. O seu objetivo é revelar falhas em um produto, para que as causas dessas falhas sejam identificadas e possam ser corrigidas pela equipe de desenvolvimento antes da entrega final. Por conta dessa característica das atividades de teste, dizemos que sua natureza é “destrutiva”, e não “construtiva”, pois visa ao aumento da confiança de um produto através da exposição de seus problemas, porém antes de sua entrega ao usuário final.

O conceito de teste de software pode ser compreendido através de uma visão intuitiva ou mesmo de uma maneira formal. Existem atualmente várias definições para esse conceito. De uma forma simples, testar um software significa verificar através de uma execução controlada se o seu comportamento corre de acordo com o especificado. O objetivo principal desta tarefa é revelar o número máximo de falhas dispondo do mínimo de esforço, ou seja, mostrar aos que desenvolvem se os resultados estão ou não de acordo com os padrões estabelecidos.

Já o planejamento dos testes deve ocorrer em diferentes níveis e em paralelo ao desenvolvimento do software. Os principais níveis de teste de software são:

- Teste de Unidade: também conhecido como testes unitários. Tem por objetivo explorar a menor unidade do projeto, procurando provocar falhas ocasionadas por defeitos de lógica e de implementação em cada módulo, separadamente. O universo alvo desse tipo de teste são os métodos dos objetos ou mesmo pequenos trechos de código.
- Teste de Integração: visa provocar falhas associadas às interfaces entre os módulos quando esses são integrados para construir a estrutura do software que foi estabelecida na fase de projeto.
- Teste de Sistema: avalia o software em busca de falhas por meio da utilização do mesmo, como se fosse um usuário final. Dessa maneira, os testes são executados nos mesmos ambientes, com as mesmas condições e com os mesmos dados de entrada que um usuário utilizaria no seu dia-a-dia de manipulação do software. Verifica se o produto satisfaz seus requisitos.
- Teste de Aceitação: são realizados geralmente por um restrito grupo de usuários finais do sistema. Esses simulam operações de rotina do sistema de modo a verificar se seu comportamento está de acordo com o solicitado.
- Teste de Regressão: Teste de regressão não corresponde a um nível de teste, mas é uma estratégia importante para redução de “efeitos colaterais”. Consiste em se aplicar, a cada nova versão do software ou a cada ciclo, todos os testes que já foram aplicados nas versões ou ciclos de teste anteriores do sistema. Pode ser aplicado em qualquer nível de teste.

- A copa (com frutos, flores e folhas) corresponde às *habilidades*: Agir com talento, capacidade técnica, obtendo resultados positivos é o que chamamos de habilidade. Uma competência só é reconhecida quando disponibilizada. Não adianta alguém saber que sabe. Para obter o reconhecimento, precisa demonstrar.

As competências de um bom profissional devem ser formadas por uma combinação de conhecimentos, habilidades e atitudes. Apenas o conhecimento, sem habilidades e atitudes, nos passa a sensação de um profissional por demais teórico, sem capacidade de externalizar e colocar em prática todo o seu conhecimento. Só habilidades, sem conhecimentos e boas atitudes nos traz a imagem de um bom operador, com conhecimentos restritos à sua função, com pouca capacidade de desenvolvimento e de tomada de atitudes. Este tipo de profissional faz bem o que o mandam. Pode ser facilmente substituído, mas em muitos casos, são extremamente úteis. Da mesma maneira, as atitudes sozinhas, não são suficientes para qualificar um grande profissional, pois este poderá apresentar boa índole, boa postura profissional, mas pouco conseguirá produzir, por falta de conhecimento.

Conhecimentos

Conhecimentos em Engenharia de Software são essenciais para um testador. [DIMAGGIO] divide esse conhecimento em três áreas: *processo de desenvolvimento de software, programação de computadores e sistemas operacionais (SO)*.

O testador deve conhecer o **processo de desenvolvimento** no qual está inserido, pois suas atividades fazem parte desse processo. Para desempenhar bem a função, deve dominar o processo. Ele deve conhecer quais são os artefatos de entrada e saída de cada fase do processo, principalmente das que envolvem atividades de testes. Dessa forma, poderá antever problemas e sugerir soluções antecipadamente, além de planejar as atividades com maior segurança, uma vez que terá uma visão ampla de como funcionam as interfaces das fases do processo. Portanto, deve procurar entender todo o processo de desenvolvimento da sua empresa. Caso não esteja trabalhando, pode estudar processos modelos, como por exemplo, RUP, PRAXIS, XP, SCRUM, V-MODEL, TMAP, pois com certeza, se os processos das empresas não são fiéis a algum desses, são bem próximos. Conhecer a fundo o processo da sua empresa faz grande diferença entre os profissionais da organização.

O bom testador também se destacará caso saiba **programação**. Isso não quer dizer que tenha que dominar linguagens, arquiteturas e programar algoritmos elaborados. Mas deve conhecer lógica de programação, características de linguagens e *frameworks*. Esse conhecimento fará com que ele saiba o ponto fraco dessas tecnologias, uma vez que já as utilizou e sabe onde errou mais. Um testador que já programou em C sabe que muitos erros são cometidos na manipulação de memória, por exemplo, causados pelo esquecimento em desalocá-las, o que poderá gerar um estouro de memória. Já em Java, essa manipulação é automática. Em contrapartida,

falhas de acesso concorrente (acessos a objetos que já foram eliminados) em Java são facilmente reproduzidas, caso essa situação não seja tratada.

Entender dos **sistemas operacionais** é outro aspecto importantíssimo. Conhecer as políticas de permissões de acessos, configurações de variáveis de ambientes é importante para saber diferenciar o que são falhas na aplicação ou apenas configurações desajustadas. Esse conhecimento traz produtividade para os testes, evitando o registro de falhas e investigações sem necessidade. Comandos básicos e intermediários, como por exemplo, os comandos *diff, wc e tail -f* do UNIX são ferramentas “poderosas” que facilitam os testes dos testadores. Saber UNIX (e seus “familiares”), com seus processos e *daemons*, fará com que o testador seja mais completo, se diferenciando daquele que testa apenas sistemas com interfaces amigáveis do *Windows*. Esse conhecimento em SO deve abranger a criação de scripts, sejam eles *shellscripts*, ou os usuais *.bat*. Isto torna o testador apto a automatizar tarefas simples, que geram produtividade caso sejam automatizadas. Por exemplo, um *script* que obtém o código-fonte de uma aplicação em um repositório, compila esse código, realiza o *deploy*, chama a ferramenta de execução de testes automáticos, e depois envia o resultado por e-mail, será bastante útil para executar os testes de regressão executados no período da noite.

Além dos conhecimentos destacados por [DIMAGGIO], saber realizar consultas em **banco de dados** é essencial. Muitas vezes não é possível, através da interface de usuário da aplicação, verificar todos os resultados esperados. Neste caso, realizar consultas SQL é a única saída. Não é requerido que o profissional saiba consultas complexas, mas logicamente, quanto mais souber, mais irá se destacar.

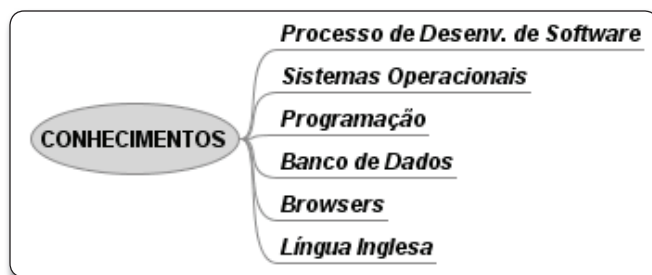


Figura 1. Mapa Mental dos Conhecimentos

Além dos conhecimentos principais ilustrados na Figura 1, é interessante que o testador conheça as características dos *bro-wsers* mais utilizados no mercado e suas configurações, como por exemplo, configurações de *cache*, de segurança e privacidade. Muitas vezes os problemas estão nessas configurações, e não na aplicação *Web*. Principalmente para testes de performance (*stress*, volume, carga etc.), é importante o conhecimento em protocolos de rede, como o *http*. Para esse tipo de testes também é importante que o profissional tenha noção do funcionamento dos itens que comumente são monitorados, como é o caso da memória e CPU. Mas o conhecimento não precisa ser profundo, uma vez que esse tipo de teste envolve (ou deveria envolver) todas as áreas de apoio possíveis, como suporte de rede, SO, DBA, arquitetos, e não apenas testadores.

Ferramentas de testes são fundamentais, mas o profissional de testes não precisa dominá-las para iniciar sua carreira. Com os conhecimentos apresentados aqui, torna-se fácil o aprendizado. Por exemplo, quem sabe programar não terá dificuldades em aprender a automatizar testes. Quem conhece o funcionamento básico da Internet (*browser*, requisições *http*), não terá dificuldades em aprender a utilizar uma ferramenta para testes de *stress*. Mas, logicamente, o profissional que já conhece essas ferramentas poderá ser mais bem avaliado em uma entrevista de emprego.

Interessante citar que conhecer a **língua inglesa** também é uma vantagem, pois os melhores livros e grandes artigos estão escritos nessa língua, além das melhores ferramentas. Dessa forma, o profissional de testes terá maior facilidade para evoluir no que tange aos seus conhecimentos.

Habilidades

Em seu artigo, [DIMAGGIO] enfatiza duas características que serão aqui categorizadas como habilidades: *comunicação* e *organização*. Além dessas, a *capacidade analítica* e a *facilidade de aprendizado* são outras habilidades importantes.

A **comunicação** pode ser dividida em comunicação escrita e comunicação oral. Ambas devem ser uma comunicação clara e objetiva. O analista de testes deve escrever bem, pois uma de suas atividades principais é elaborar os casos de testes. Ele deverá escrever, no mínimo, a descrição do caso de testes, o passo a passo para execução, os pré-requisitos e os resultados esperados. Durante a execução dos testes, a todo o momento deverá registrar as falhas, transcrevendo para a ferramenta as ações para que a falha seja reproduzida. Caso não estejam claros, tanto a execução dos testes, quanto a reprodução das falhas identificadas, poderá ser comprometida, uma vez que na maioria das vezes, outras pessoas que irão consumir os textos escritos. Além desses casos citados, a comunicação por e-mail também é muito utilizada. O analista de testes que não sabe escrever, não passa credibilidade. Portanto, não é um bom testador. Planos de testes, relatórios de execuções são outros exemplos de documentos que também são gerados pela equipe de testes, e que devem ser bem escritos.

Quanto à comunicação oral, ela também deve ser clara. Muitas situações só são resolvidas através de diálogos, e muitas dúvidas através de perguntas bem elaboradas. Saber o que perguntar transmite credibilidade. A interação entre os desenvolvedores e os testadores deve ser constante para evitar mal-entendidos. Apesar de um e-mail ser bem escrito, ainda sim é uma comunicação impessoal. A comunicação com o cliente em uma homologação segue as mesmas idéias. Além dessas situações básicas, é normal a necessidade de treinamento por parte da equipe de testes, seja de ferramentas ou de processos. O profissional que sabe falar em público e transmitir seu conhecimento de forma clara possui um diferencial.

Possuir **organização** é fundamental para executar atividades de testes. [DIMAGGIO] cita um cenário comum: é quase sempre impossível executar os testes planejados em sistema onde o testador não seja obrigado a interrompê-los devido à

identificação de um *bug* que bloqueie o restante da execução e o obrigue a continuar os testes em outra parte do sistema até o *bug* seja removido. O testador deverá ser organizado o suficiente para interromper um teste em uma parte do sistema, começar os testes em outra parte, e depois voltar para a parte anterior de onde ele parou, sem prejuízos para a qualidade do sistema. A organização é uma habilidade essencial para todas as atividades.

A **capacidade analítica** é uma habilidade que se refere à capacidade de analisar dados, muitas vezes não relacionados entre si, e após essa análise, gerar informações de valor. O testador deverá analisar o sistema que está sendo testado entendendo não apenas suas partes isoladamente, mas também como se relacionam. Não deve apenas seguir um roteiro de testes, mas sim entender o sistema e seu contexto. É no momento da elaboração dos casos de testes que essa habilidade é mais requerida. O testador deve ser capaz de imaginar testes válidos e inválidos que teste todo o sistema.

Existem diversas ferramentas, gratuitas e pagas, para apoiar e/ou viabilizar os testes de software. Um bom testador deve ter **facilidade de aprendizado** para poder avaliá-las, utilizá-las e disseminá-las dentro da sua empresa. Portanto, deve ser capaz de realizar comparações entre ferramentas que possuem a mesma finalidade de forma a encontrar a melhor delas. Essa facilidade de aprender e trabalhar com ferramentas é muito bem-vinda. A **Figura 2** apresenta o mapa mental para as habilidades.

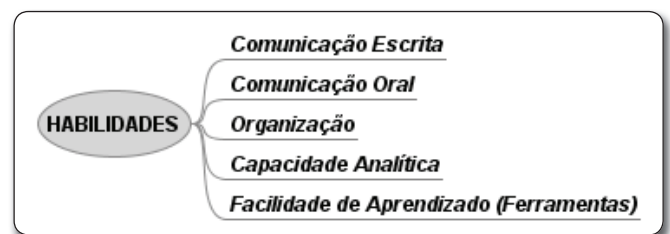


Figura 2. Mapa Mental das Habilidades

Atitudes

No seu tradicional livro, *The Art of Software Testing*, [MYERS] dedica grande parte de um capítulo para discutir questões da psicologia que envolve as atividades de testes. Ele diz que o teste de software é uma tarefa técnica, mas que envolve grandes questões da psicologia humana. Portanto, algumas atitudes devem ser direcionadas a essas questões para que o relacionamento entre desenvolvedores, gerentes e testadores seja saudável. Essas atitudes serão úteis, por exemplo, caso uma notícia não tão boa tenha que ser comunicada pela equipe de testes ao gerente/desenvolvedor. Dependendo da forma como essa informação é passada poderá ou não “sobrar” para o testador: “Se não gosta da mensagem, bate no mensageiro”.

O relacionamento entre o desenvolvedor e o testador é um barril de pólvora prestes a explodir caso os desenvolvedores não entendam o papel dos testadores, e caso os testadores não entendam a posição de um desenvolvedor. Essa última justificada por [SPILLNER], que diz que todos cometem erros,

mas ninguém gosta de admiti-los. Quando um produto gerado pelo desenvolvedor é questionado por outros, ou seja, colocado à prova, a tendência é que ele fique na “defensiva”, cheio de justificativas, pois não quer que suas falhas venham à tona. A natureza humana é assim.

Para justificar essa “guerra fria” basta pensar: o “sucesso” de um é o “fracasso” do outro. Ou seja, quanto mais falhas forem identificadas, melhor será visto o trabalho do testador, mas não o do desenvolvedor. O contrário também é verdadeiro. Quanto menos falhas identificadas, melhor será visto o trabalho do desenvolvedor, mas não o do testador, principalmente se essas falhas forem percebidas pelo cliente em homologação ou em produção. Essa competição é saudável, mas de maneira nenhuma poderá ser levada para o lado pessoal. Ambos devem lembrar que possuem o mesmo objetivo: a qualidade do produto. Quando essa maturidade é atingida, o projeto e a empresa ganham.

Uma atitude que vale para todos em todas as profissões, e dessa forma para os testadores, é a **pró-atividade**. Esta tem um peso muito grande para os testadores, pois o teste costuma ser uma das últimas, senão a última atividade a ser realizada antes do produto entrar em produção. Além disso, geralmente, já começa com atraso e prazo reduzido devido ao acúmulo de atrasos das atividades anteriores. Por conta desses fatores, o profissional de testes não pode ficar parado quando encontra obstáculos, pois não existe tempo hábil para recuperação.

Voltando à questão da credibilidade, uma atitude do testador que proporciona o aumento da confiança do desenvolvedor nos testes de software é a **autoconfiança**. Se questionado, o testador deverá mostrar segurança em suas ações e decisões. Mas a autoconfiança não pode ser confundida com confiança exagerada, pois essa não possui garantia. O testador deve estar balizado para poder apresentar seus questionamentos e justificativas. O “eu acho”, o “não sei direito” devem ser evitados. Normalmente, a falta dessa atitude vem acompanhada do acanhamento, o que pode passar aos desenvolvedores, e principalmente aos clientes, fraquezas e incertezas.

A **paciência**, acompanhada da **persistência**, deve fazer parte do repertório de atitudes dos testadores. Muitas vezes eles são obrigados a realizarem tarefas repetitivas e “manuais”, e também testar a mesma funcionalidade várias vezes. A paciência também é um componente da “**diplomacia**”. Conforme dito anteriormente, o relacionamento entre desenvolvedor e testador, muitas vezes, é tenso. O bom desenvolvedor possui um ego muito forte. O testador deve ser diplomático, ou seja, utilizar bons modos e delicadeza para conduzir uma questão, pois, querendo ou não, na maioria das vezes, é portador de más notícias. Conforme [LOVELAND], terá que dizer ao desenvolvedor “*The baby is ugly!*”.

[DIMAGGIO] diz que o profissional de testes deve possuir alto grau de **ceticismo**, mas sem hostilidades. Nunca deve acreditar nas coisas na primeira vez que as vê. Deve questionar até que tudo seja provado. “Mostre-me!”. Deve usar a sua teimosia para **investigar** o correto funcionamento do sistema. Essa atitude investigativa também é ressaltada por [DIMAGGIO].

Muitas vezes vários artefatos devem ser consultados para que se chegue a uma conclusão.

A próxima atitude é a **criatividade destrutiva**. O profissional de testes não pode ter medo de “derrubar” o sistema durante os testes. Nos testes de software os limites existem para serem ultrapassados. É normalmente ali onde os programadores cometem mais erros. “O que irá acontecer caso eu faça isso?”. Esse tipo de pergunta deve estar na cabeça do testador. Apesar de existirem casos de testes a serem seguidos, essa atitude destrutiva é um *plus* no momento dos testes, podendo identificar várias falhas que, se corrigidas, podem tornar o sistemas bem mais robusto.

As duas últimas atitudes que [DIMAGGIO] cita em seu artigo, e que são bastante interessantes são: **ânsia por novas tecnologias** e **valorização da perspectiva do usuário**. A primeira é mais global, aplicando não só aos profissionais de testes e se refere à vontade de aprender e trabalhar com as novas tecnologias que surgem constantemente na área. Não se pode parar no tempo. O profissional deve se manter atualizado, estudando novas tecnologias, arquiteturas, padrões de mercado, processos etc. Sempre poderão aplicar algum aprendizado. Já a segunda é mais específica para os testadores. Eles devem se colocar no lugar do usuário quando estão executando os testes, tornando-se um “advogado” do usuário. Nesta perspectiva, e na falta de uma equipe de usabilidade, sugestões de melhorias levantadas pelos testadores podem render novos serviços e um produto de melhor qualidade.

[LOVELAND] ainda lembra da **curiosidade**. Perguntas como “Porque?”, “E se...?” devem ser comuns. Os testadores não devem ser tímidos em perguntar. A **Figura 3** apresenta o mapa mental para as atitudes.

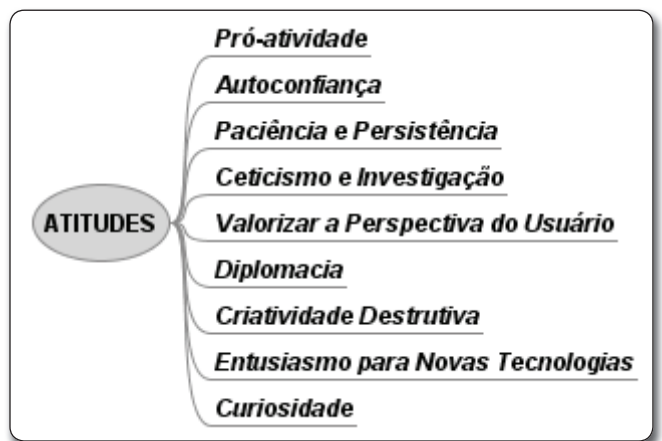


Figura 3. Mapa Mental das Atitudes

O que fazer?

Após a avalanche de conhecimentos, habilidades e atitudes descritas, o profissional deve estar se perguntando como e por onde começar a estudar e colocar em prática isso tudo. Não existe um roteiro para isso. Para atacar os conhecimentos citados aqui, o profissional poderá estudar sozinho, ou mesmo se matricular em cursos específicos. Atualmente existem vários

sobre Testes de Software, que poderão ser úteis para complementar o currículo das faculdades. Já as disciplinas de Banco de Dados, Programação e Sistemas Operacionais, que fazem parte da grade do curso de graduação, essas os profissionais devem levar muito a sério, aproveitando o momento na universidade. Para quem gosta de estudar sozinho, a Internet proporciona vários sites sobre testes com fóruns e listas de discussão onde é possível encontrar materiais bastante úteis. Os profissionais devem estar sempre lendo artigos, reportagens e *cases* de sucessos relacionados à área de Testes de Software.

As certificações são muito bem vindas. A preparação para elas envolve muito estudo. Além disso, muitas empresas já valorizam e até exigem que o profissional seja certificado para que se torne apto a preencher uma vaga. Algumas licitações já exigem das empresas participantes que possuam profissionais certificados.

Quanto às habilidades e atitudes, essas não são adquiridas apenas estudando. Devem ser praticadas. Redija textos e peça para algum amigo revisar. Apresente oralmente trabalhos na faculdade, ou na empresa onde trabalha. Seja candidato para avaliar ferramentas de testes. Não evite essas situações, ao contrário, vá de encontro a elas. Pense e aja de acordo com as atitudes citadas e comece a perceber os resultados. Repare nas outras pessoas que não possuem as mesmas atitudes, e perceba as diferenças.

Estrutura de uma Equipe de Testes

Uma equipe onde todos os membros possuem as mesmas características, ou seja, o mesmo conjunto de CHA, não é uma equipe desejável. A não ser, é claro, que todos eles possuam **todos** os conhecimentos, habilidades e atitudes necessárias. Esse cenário seria o ideal, mas na prática, isso é muito difícil de acontecer. Segundo [CHIAVENATO], a variabilidade humana é enorme. As pessoas apresentam profundas diferenças individuais. Cada pessoa tem a sua própria personalidade, a sua história pessoal, seus conhecimentos e habilidades, seus objetivos e motivações, suas limitações pessoais etc. Elas reagem de forma diferente a uma mesma situação. Além desses aspectos, a experiência dos profissionais também é diferente, tornando-os melhores que os outros em uma determinada competência.

Por essa razão, deve-se buscar montar uma equipe de testes equilibrada, formada por membros que se complementam. Devem possuir pessoas experientes (consequentemente, com salários mais altos), e outras nem tanto. Até mesmo para tornar o custo da equipe viável. As competências são identificadas e agrupadas para satisfazerem os **papéis** dentro de uma equipe de testes. Os livros da área costumam propor estruturas de equipes. As mais comuns, conforme citado em [SPILLNER], possuem *testadores*, *analistas de testes*, *arquitetos de testes*, *automatizadores de testes*, *líder e/ou coordenador de testes*.

Os *testadores* são os responsáveis pela execução dos testes. Seguem um roteiro de casos de testes e registram as falhas quando encontradas. São responsáveis também por verificar as correções realizadas pelos desenvolvedores. Podem também

revisar os casos de testes. Normalmente são os profissionais que estão começando na área.

Já os *analistas de testes* são mais experientes que os testadores e já realizaram esta função. São os responsáveis por criar os casos de testes, baseando-se nos requisitos do usuário ou na especificação funcional do sistema. Devem identificar os pontos a serem testados e o resultado esperado; extrair os pré-requisitos e os dados de entrada; elaborar o passo-a-passo, criando os casos de testes para que o testador possa executá-los. Os casos de testes devem ser elaborados de forma que o sistema seja coberto da melhor forma possível, levando em consideração o tempo e os recursos à disposição. O analista de testes também deverá indicar a ordem dos casos a serem executados.

Os *arquitetos de testes* são especialistas nas técnicas de testes. Eles sabem identificar quais são os tipos de testes que devem ser realizados em um determinado projeto, arquitetando todas as questões referentes aos testes, como por exemplo, preparação do ambiente, geração da massa de dados, seleção da melhores ferramentas e a identificação dos casos de testes a serem automatizados.

O *automatizador de testes* deve possuir bons conhecimentos em programação, pois a automação envolve intervenções manuais nos *scripts* gerados pelas ferramentas de automação. Esses profissionais devem, por exemplo, inserir no código pontos de verificações, criar rotinas para leitura de dados de entrada proveniente de arquivos ou banco de dados, e implementar mecanismos de *logs* para facilitar a análise da execução do testes. A criação de bibliotecas com funções relativas às intervenções citadas é uma boa prática, pois gera produtividade no momento da automação de testes, devido ao reuso. Esse profissional também seria o responsável pelos testes de performance, que também são testes automatizados. É importante ressaltar que nesse tipo de teste, seja ele de stress, volume, carga, tempo de resposta etc., é necessário o envolvimento de várias áreas específicas de dentro da organização, como por exemplo, analistas de rede, analistas de banco de dados, desenvolvedores, administradores do sistema operacional. O profissional de testes deve ter todo esse apoio para que o trabalho seja realizado de forma satisfatória.

O *líder de testes* (ou coordenador ou gerente) é a pessoa responsável pela equipe. Além de conhecimentos específicos da área de testes, deverá possuir liderança e gestão.

Normalmente, uma pessoa experiente assume vários papéis dentro da equipe de testes. Mas a seqüência comum de atuação e aprendizagem seria está: Testador – Analista de Testes e Automatizador de Testes – Arquiteto de Testes – Líder de testes. O acúmulo de experiência proporciona essa “escalada” nas funções. Por exemplo, um bom arquiteto de testes deve possuir boa experiência como testador e analista de testes. Não faz sentido nomeá-lo arquiteto se nunca executou e nem elaborou casos de testes, ou se não possui experiência suficiente nessas atividades.

Na conclusão desse artigo, com base na experiência do autor, são apresentadas tabelas que sugerem o grau de aderência aos conhecimentos, habilidades e atitudes exigidas para cada uma das funções apresentadas.

Conclusão

As Tabelas 1, 2 e 3 não representam as competências acumuladas, mas sim o necessário para a realização de uma determinada função isoladamente. Ou seja, mesmo que um bom arquiteto de testes deva ter sido um bom testador, os quadros sugerem para o arquiteto apenas os CHA necessários para essa função, esquecendo-se do tempo em que ele foi testador. ●

Legenda:

	Baixo
	Médio
	Alto

	Papéis	Testador	Analista de Testes	Automatizador	Arquiteto de Testes	Líder de Testes
Conhecimentos						
Processo de Software						
Sistemas Operacionais						
Programação						
Banco de Dados						
Browser						
Língua Inglesa						

Tabela 1. Componente Conhecimento

	Papéis	Testador	Analista de Testes	Automatizador	Arquiteto de Testes	Líder de Testes
Habilidades						
Comunicação escrita						
Comunicação oral						
Organização						
Capacidade analítica						
Facilidade de aprendizado						

Tabela 2. Componente Habilidade

	Papéis	Testador	Analista de Testes	Automatizador	Arquiteto de Testes	Líder de Testes
Atitudes						
Pró-atividade						
Autoconfiança						
Paciência e Persistência						
Ceticismo e Investigação						
Perspectiva do Usuário						
Diplomacia						
Criatividade Destrutiva						
Entusiasmo com Novas Tecnologias						
Curiosidade						

Tabela 3. Componente Atitude

Referências

- CHIAVENATO, Idalberto. Administração de Recursos Humanos – Fundamentos Básicos. São Paulo: Atlas, 1999.
- DIMAGGIO, Len. A Roadmap for Software Test Engineering. Dr. Dobb's, 2001. <http://www.ddj.com/architect/184414700> - Visitado em 01/08/2009.
- GRAMIGNA, Maria Rita. Modelo de Competências e Gestão dos Talentos. São Paulo: Pearson Education do Brasil, 2002.
- LOVELAND, Scott; MILLER, Geoffrey; PREWITT, Richard; SHANNON, Michael; Software Testing Techniques – Finding the Defects that Matter. Charles River Media, INC. Hingham, Massachusetts, 2005.
- MYERS, Glenford J. The Art of Software Testing. John Wiley & Sons, Inc., New York, NY, 1979.
- SPILLNER, Andreas; LINZ, Tilo; SCHAEFER, Hans; Software Testing Foundations – A Study Guide for the Certified Tester Exam. Rio de Janeiro: Fundação Getúlio Vargas, 2006.
- CRAIG, R.D., JASKIEL, S.P., "Systematic Software Testing", Artech House Publishers, Boston, 2002.
- PRESSMAN, R.S., "Software Engineering: A Practitioner's Approach", McGraw-Hill, 6th ed, Nova York, NY, 2005.
- ROCHA, A.R.C., MALDONADO, J.C., WEBER, K.C. et al., "Qualidade de software – Teoria e prática", Prentice Hall, São Paulo, 2001.

Dê seu feedback sobre esta edição!

A Engenharia de Software Magazine tem que ser feita ao seu gosto. Para isso, precisamos saber o que você, leitor, acha da revista! Dê seu voto sobre este artigo, através do link:

www.devmedia.com.br/esmag/feedback

