TMMi no mundo Agile

Versão 1.3

Produzido pelo *TMMi Foundation*

Editor: Erik van Veenendaal

Copyright Notice
Unlimited distribution subject to Copyright
Copyright © TMMi Foundation, Irlanda.

Este material do *TMMi Foundation* é fornecido "como está".

O *TMMi Foundation* não oferece garantias de qualquer tipo, expressas ou implícitas, em relação a qualquer assunto, incluindo, entre outros, garantia de adequação a um propósito ou comercialização, exclusividade ou resultados obtidos com o uso do material. O *TMMi Foundation* não oferece qualquer tipo de garantia com relação à liberdade de violação de patente, marca comercial ou direitos autorais.

O uso de qualquer marca registrada neste documento não se destina a infringir os direitos do titular da marca.

A permissão para reproduzir este documento e preparar trabalhos derivados deste documento para uso interno é concedida, desde que os direitos autorais e as declarações "Sem garantia" estejam incluídos em todas as reproduções e trabalhos derivados.

Os pedidos de permissão para reproduzir este documento ou preparar trabalhos derivados deste documento para uso externo e comercial devem ser endereçados ao *TMMi Foundation.*

TMMi® é uma marca registrada do *TMMi Foundation*.

Contribuidores

Asim Ali	(Emirados Árabes Unidos)
Katalin Balla	(Hungria)
Clive Bates	(Reino Unido)
Jan Jaap Cannegieter	(Holanda)
Vahid Garousi	(Holanda)
Alon Linetzki	(Israel)
Fran O'Hara	(Irlanda)
Jurian van de Laar	(Holanda)
Leanne Howard	(Austrália)
Poonam Jain	(Índia)
Tim Moore	(Reino Unido)
Alfonsina Morgavi	(Argentina)
Meile Posthuma	(Holanda)
Matthias Rasking	(Alemanha)
Chaobo Shang	(China)
Erik van Veenendaal	(Bonaire - Países Baixos Caribenhos)
Blaine Webb	(Reino Unidos)
Karolina Zmitrowicz	(Polônia)

Histórico

Esta seção é fornecida apenas para fins informativos.

Versão	Data	Comentário
1.0	30/06/2017	Capítulo introdutório e todas as áreas de processo do TMMi Nível 2 abordadas
1.1	16/05/2018	Adicionada a interpretação em um contexto Agile de todas as áreas de processo do TMMi nível 3.
1.2	10/12/2018	Adicionada a interpretação em um contexto Agile de todas as áreas de processo do TMMi nível 4.
1.3	03/07/2019	Adicionada a interpretação em um contexto Agile de todas as áreas de processo do TMMi nível 5.

Conteúdo

1 Introdução 6
1.1 Objetivo 6
1.2 TMMi e Agile 6
1.3 Integração do Modelo de Maturidade do Teste (TMMi) 7
1.4 Agile 8
1.5 Melhoria do processo de teste em um contexto Agile 8
2 TMMi Nível 2 Gerenciado 10
2.1 Área de Processo 2.1 Política e Estratégia de Teste 10
2.2 Área de Processo 2.2 Planejamento de Teste 12
2.3 Área de Processo 2.3 Monitoramento e Controle de Teste 15
2.4 Área de Processo 2.4 Projeto e Execução de Teste 18
2.5 Área de Processo 2.5 Ambiente de Teste 22
3 TMMi Nível 3 Definido 24
3.1 Área de Processo 3.1 Organização de Teste 24
3.2 Área de Processo 3.2 Programa de Treinamento em Teste 26
3.3 Área de Processo 3.3 Ciclo de Vida e Integração de testes 27
3.4 Área de Processo 3.4 Testes Não Funcionais 30
3.5 Área de Processo 3.2 Revisões por Pares 32
4 TMMi Nível 4 Medido 34
4.1 Área de Processo 4.1 Medição de Teste 35
4.2 Área de Processo 4.2 Avaliação da Qualidade do Produto 36
4.3 Área de Processo 4.3 Revisões Avançadas 37
5 TMMi Nível 5 Otimização 37
5.1 Área de Processo 5.1 Prevenção de Defeitos 38
5.2 Área de Processo 5.2 Controle de Qualidade 39
5.3 Área de Processo 5.3 Otimização do Processo de Teste 40
6 Visão Geral, Aplicabilidade, Objetivos e Práticas Específicas do TMMi 41
6.1 Avaliações TMMi 41
6.2 TMMi Nível 2 Gerenciado 41
6.3 TMMi Nível 3 Definido 41
Referências 42

1 Introdução

1.1 Objetivo

Este documento explica como a combinação de uma abordagem Agile para desenvolvimento de software com o modelo de aprimoramento de processo de teste TMMi é uma rota possível para atingir seus objetivos de negócios. Explica como o TMMi pode ser usado e aplicado de forma benéfica em um contexto Agile. Alternativas comprovadas são fornecidas às abordagens de teste tradicionais que implementam práticas de TMMi, mantendo e, talvez até aumentando a agilidade. Este documento aborda como o TMMi pode ajudar as organizações Agile, independentemente da maturidade da organização em sua jornada para níveis crescentes de agilidade, fornecendo lembretes de práticas críticas de teste que frequentemente perdem visibilidade à medida que as organizações crescem e as pressões do projeto aumentam. Cada área de processo do TMMi e seus objetivos específicos são discutidos um por um. É declarado como eles se relacionam com o Agile e como serão as práticas. Se não se espera que uma prática seja executada em um contexto Agile, uma vez que não possui valor agregado, isso também é explicitamente declarado.

Muitas organizações (pequenas) ágeis são bem-sucedidas e crescentes, mas eles podem ter poucos processos documentados e pouco programa formal de treinamento para as pessoas envolvidas. Para manter o sucesso à medida que a organização cresce, será necessário um pouco mais de disciplina no processo. As organizações geralmente têm medo de perder a cultura Agile que levou ao sucesso atual. Esse é um dos desafios à frente ao iniciar uma iniciativa de melhoria de processo de teste TMMi em um ambiente Agile.

Este documento tem muitos públicos-alvo, sendo os dois principais grupos:

- Organizações tradicionais maduras que desejam mudar para o Agile, mantendo a maturidade do processo.
- Organizações Agile que são bem-sucedidas e estão crescendo. Como resultado, eles precisam de algum tipo de maturidade do processo e, ao mesmo tempo, desejam manter os benefícios de serem Agile.

Este documento não se destina a ser usado independentemente do modelo TMMi original. O objetivo é fornecer orientação àqueles que executam a melhoria do processo de teste em um ambiente Agile sobre a interpretação e o significado dos vários objetivos e práticas do TMMi. Este documento complementa o modelo TMMi original e deve ser usado como documento complementar.

1.2 TMMi e Agile

A crença equivocada é que as abordagens do TMMi e do Agile estão em desacordo. As abordagens ágeis e o TMMi podem não apenas coexistir, mas, quando integradas com sucesso, trarão benefícios substanciais. Também existe o desafio de analisar os testes de maneira diferente, sendo totalmente integrado ao desenvolvimento Agile e o que isso significa no contexto de um programa de aprimoramento de "testes". Observe que o “i” do TMMi se refere ao fato de que o teste deve ser uma parte integrada do desenvolvimento de software e não deve ser tratado como algo totalmente separado. A literatura e as apresentações sobre testes em projetos Agile tendem a se concentrar em testes de unidade, automação de testes e testes exploratórios, mas há mais! O uso do modelo TMMi em um contexto Agile fornece lembretes de práticas críticas de teste que geralmente são "esquecidas". Este documento mostrará com exemplos que os métodos TMMi e Agile podem trabalhar efetivamente juntos. O desafio é aplicar princípios enxutos para capacitar práticas ágeis e facilitar as práticas de TMMi.

Ao implementar o TMMi, é preciso levar em consideração que a intenção do modelo do TMMi não é “impor” um conjunto de práticas a uma organização, nem deve ser aplicada como um padrão ao qual se deve “provar a conformidade”. Quando usado adequadamente, o TMMi pode ajudá-lo a localizar as áreas de teste específicas nas quais as mudanças podem fornecer valor, considerando os objetivos de negócios. Isso ocorre independentemente do modelo de ciclo de vida que está sendo aplicado. É importante lembrar sempre que as práticas do TMMi são um componente esperado, mas também podem ser alcançadas pelo que é chamado de prática “alternativa” em relação a uma prática definida do TMMi. Sempre pense, qual é a intenção da prática, qual é a lógica e como ela agrega valor ao negócio? Muitas vezes, em uma cultura ágil, a intenção já é alcançada, mas através de uma prática alternativa. Normalmente, qualquer solução é compatível desde que seja orientada pelas necessidades da empresa! Quando o uso do TMMi não é muito prescritivo, não foi assim que o TMMi se destina. Sempre interprete os objetivos e práticas da TMMi no contexto da sua situação. Em geral, primeiro estabelecendo as necessidades do processo em seu contexto comercial específico, é possível tomar decisões sobre como focar e orientar as prioridades de melhoria do processo.

A maioria dos conflitos que surgem entre o TMMi e o Agile é baseada em uma visão histórica do TMMi de como deve ser uma “boa prática” quando implementada ou em um mal-entendido da lógica das práticas ágeis com base em como elas devem suportar os valores ágeis. Os especialistas da TMMi, incluindo os avaliadores (líderes), precisarão repensar e potencialmente repensar as mensagens que possam ser compartilhadas inadvertidamente relacionadas a como deve ser a prática "boa prática compatível com TMMi" quando implementada. Quando as abordagens ágeis são implementadas adequadamente, juntamente com os processos do TMMi, isso resulta na implementação efetiva das práticas de teste, não na exclusão delas. Observe que, além das práticas específicas (de teste) no TMMi, também existem práticas genéricas. O objetivo das práticas genéricas é apoiar a institucionalização de uma área de processo, o que significa efetivamente garantir que a organização tenha uma infraestrutura instalada para apoiar a área de processo quando novas pessoas entrarem ou outras mudanças acontecerem dentro da organização.

A mudança do desenvolvimento de software tradicional para o Agile também pode trazer a iniciativa de remover e enxugar os processos conforme são definidos hoje. Dessa maneira, as organizações baseadas no TMMi se beneficiarão do modo de pensar Agile. Tem havido uma tendência para as pessoas lerem coisas no modelo TMMi que não estão lá e, portanto, criam processos e produtos de trabalho desnecessários sem valor agregado. Voltando às raízes e aos objetivos de melhoria e usando o modelo TMMi como pretendido, apoiaremos o alinhamento de processos reais de valor agregado com as necessidades e objetivos reais do processo. Processos de remoção e inclinação com uma mentalidade ágil resultarão em processos que refletem o que as pessoas realmente fazem e garantirão que apenas os dados utilizados sejam coletados. A mentalidade ágil também focará em manter as coisas o mais simples possível, o que normalmente não é fácil, mas trará um benefício para quem pratica uma implementação do TMMi. As melhorias no Agile geralmente ocorrem por meio de pequenas equipes capacitadas que podem agir rapidamente, que é outra maneira pela qual o TMMi pode se beneficiar por ser Agile. Lembre-se também de que existe uma ruptura natural entre o TMMi nível 3 e o TMMi níveis 4 e 5. Recomenda-se que organizações especialmente ágeis escolham e executem criticamente as práticas nos níveis 4 e 5 do TMMi que importam e agregam valor. Embora o TMMi seja abrangente, para ter sucesso, as organizações devem identificar as principais práticas e melhorias de teste que exigem foco.

1.3 Integração do Modelo de Maturidade do Teste (TMMi)

A estrutura do TMMi foi desenvolvida pela Fundação TMMi como uma diretriz e estrutura de referência para a melhoria do processo de teste, abordando questões importantes para gerentes de teste, engenheiros de teste, desenvolvedores e profissionais de qualidade de software. Teste conforme definido no TMMi em seu sentido mais amplo para abranger todas as atividades relacionadas à qualidade do produto de software.

O TMMi usa o conceito de níveis de maturidade para avaliação e melhoria de processos. Além disso, áreas, objetivos e práticas do processo são identificados. A aplicação dos critérios de maturidade do TMMi melhorará o processo de teste e demonstrou ter um impacto positivo na qualidade do produto, na produtividade da engenharia de teste e no esforço de tempo de ciclo. O TMMi foi desenvolvido para apoiar as organizações na avaliação e melhoria de seus processos de teste.

O TMMi possui uma arquitetura faseada para melhoria de processos. Ele contém estágios ou níveis pelos quais uma organização passa à medida que seu processo de teste evolui de um que é ad hoc e não gerenciado para outro que é gerenciado, definido, medido e otimizado. A realização de cada estágio garante que todos os objetivos desse estágio sejam alcançados e as melhorias formam a base para o próximo estágio.

A estrutura interna do TMMi é rica em práticas de teste que podem ser aprendidas e aplicadas de maneira sistemática para suportar um processo de teste de qualidade que melhora em etapas incrementais. Existem cinco níveis no TMMi que prescrevem a hierarquia de maturidade e o caminho evolutivo para testar a melhoria do processo. Cada nível possui um conjunto de áreas de processos que uma organização deve implementar para atingir a maturidade nesse nível. As áreas de processo para cada nível de maturidade do TMMi são mostradas na Figura 1.

Um princípio subjacente principal do TMMi é que ele é um modelo genérico aplicável a vários modelos e ambientes de ciclo de vida. A maioria das metas e práticas definidas pelo TMMi demonstraram ser aplicáveis aos modelos de ciclo de vida sequenciais e iterativos, incluindo o Agile. No entanto, no nível mais baixo do modelo, muitas das sub-práticas e exemplos fornecidos são (muito) diferentes, dependendo do modelo de ciclo de vida que está sendo aplicado. Observe que, no TMMi, apenas os objetivos são obrigatórios, as práticas não são.

O TMMi está disponível gratuitamente no site da *TMMi Foundation*. O modelo também foi traduzido em espanhol, francês e chinês. O TMMi também está disponível em formato de livro publicado.

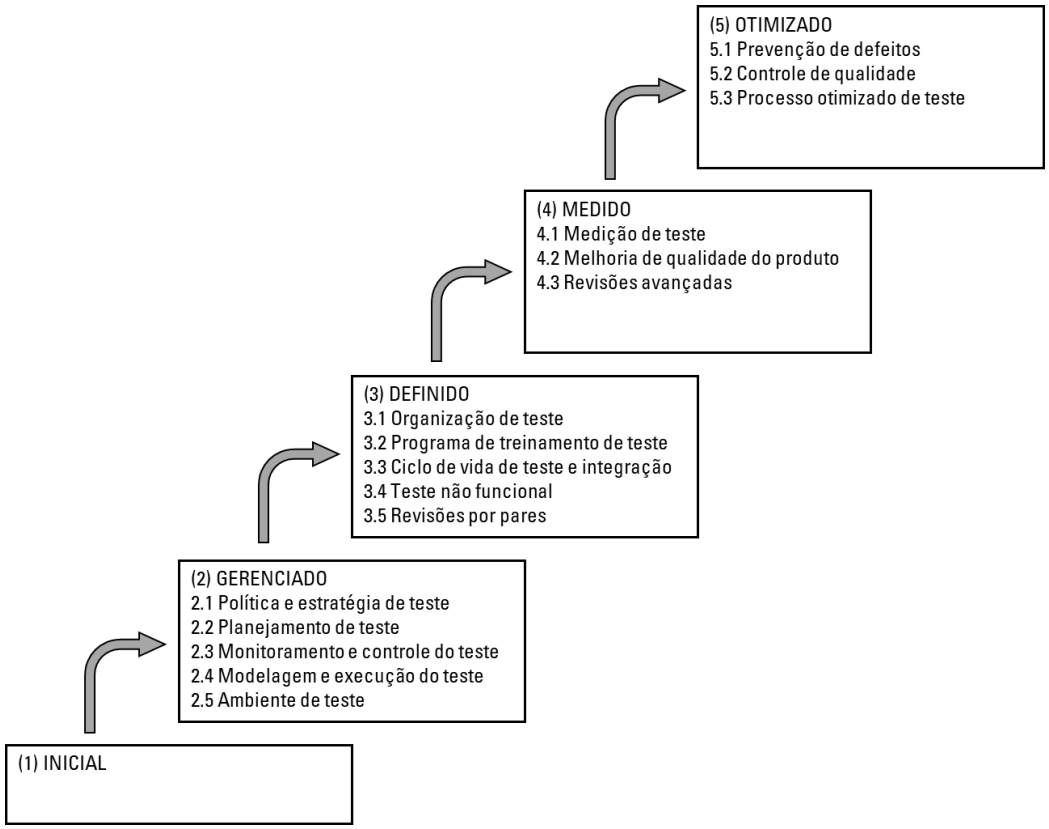


Figura 1: Níveis de maturidade do TMMi e áreas de processo

1.4 Agile

Em 2001, um grupo de indivíduos, representando as metodologias de desenvolvimento de software leve mais amplamente usadas, concordou com um conjunto comum de valores e princípios que ficou conhecido como o Manifesto para o Desenvolvimento Agile de Software ou o Manifesto Agile. O Manifesto Agile contém quatro declarações de valores:

- Indivíduos e interações sobre processos e ferramentas
- Software de trabalho sobre documentação abrangente
- Colaboração do cliente sobre negociação de contrato
- Responder às mudanças ao longo de um plano

O Manifesto Agile argumenta que, embora os conceitos à direita tenham valor, os da esquerda têm maior valor. O Agile em si não é uma metodologia, mas foram desenvolvidas metodologias diferentes que fornecem práticas para colocar esses valores em ação. Três representantes importantes de metodologias que suportam o Agile são *Extreme Programming* (XP), *Scrum* e *Kanban*.

1.5 Melhoria do processo de teste em um contexto Agile

O Agile tem um forte foco nas equipes capacitadas que realizam sua própria melhoria contínua localmente também por meio de retrospectivas frequentes. Algumas dessas melhorias podem ser focadas no teste. Elas podem ser direcionadas por um projeto de melhoria baseado no TMMi em um nível superior, mas também podem estar abordando questões de processo da equipe local relacionadas ao teste. É muito importante que a ação no contexto da melhoria do processo de teste não seja interpretada ou usada como afastamento dessa propriedade local das equipes Agile.

Alguns dos principais aspectos a serem considerados, ao analisar a influência do Agile no contexto da melhoria, são:

- Frequência do ciclo de melhoria
- Aspectos organizacionais
- Escopo das melhorias

- Fonte de melhorias
- Nível de documentação (teste)
- Métodos de melhoria.

Nos projetos que usam o Agile, as melhorias geralmente ocorrem em ciclos de feedback frequentes, que permitem que as melhorias do processo de teste sejam consideradas com frequência (p.ex.: no final de um sprint ao usar o Scrum). Como o escopo geralmente é limitado ao ciclo ou iteração anterior, são feitas pequenas mas frequentes melhorias que se concentram principalmente na solução de problemas específicos do projeto. O foco dessas melhorias geralmente não é o aprendizado entre projetos e a institucionalização de melhorias.

Observando como a melhoria do processo de teste é organizada e gerenciada, é provável que haja menos foco em um grupo de processos de teste no nível organizacional e mais ênfase no autogerenciamento das equipes no projeto. Essas equipes geralmente têm o mandato de alterar o processo de teste no projeto usado para atender às suas necessidades, resultando em processos altamente personalizados. No entanto, algumas organizações também usam reuniões semanais de teste para levar as coisas a um nível superior e entre projetos.

Como há um foco mais específico do projeto na melhoria do processo (teste), é provável que seja dada menos ênfase a problemas mais amplos que afetam os testes em toda a organização. Isso pode significar, por exemplo, que problemas fundamentais de teste podem não ser totalmente resolvidos porque estão além desse contexto centralizado no projeto. Um exemplo típico aqui é a abordagem adotada para testar certos atributos de qualidade, como desempenho e confiabilidade. Esses problemas podem ser adiados de iteração para iteração, porque geralmente exigem mais habilidades e recursos do que a equipe do projeto tem disponível. É difícil nessas áreas dar um próximo passo substancial sem grandes investimentos. A solução de problemas apenas no nível do projeto também poderia facilmente levar à sub-otimização e à perda de contato com a imagem maior.

No contexto do Agile, o alcance e o número de idéias de melhoria alternativas a serem consideradas pode ser significativamente maior do que em comparação aos modelos de ciclo de vida não-Agile. Como todos os membros realizam alguns testes no projeto, essas idéias podem vir de qualquer membro da equipe. Isso coloca uma ênfase mais forte na avaliação e priorização de sugestões de melhoria, que podem ser mais um esforço de equipe do que uma tarefa atribuída a um melhorador de processo de teste. Como isso pode exigir o conhecimento específico de teste de um melhorador de processo de teste, eles também podem atuar como consultor da equipe, mediante solicitação.

Em projetos que usam uma metodologia Agile, não espere encontrar o nível de documentação de teste que você esperaria de projetos usando um ciclo de vida sequencial. Pode haver um único “documento de teste” combinado que cubra os elementos essenciais de uma política de teste, estratégia de teste e até mesmo plano de teste de alto nível. Os aperfeiçoadores do processo de teste devem evitar sugestões de "aprimoramento" que exijam documentação de teste mais rigorosa e completa. Um dos princípios principais do Agile é claro que a documentação é criada apenas quando há uma necessidade clara e inequívoca dela. A documentação do teste não será apenas menos detalhada, o mesmo ocorrerá com a documentação do processo. Frequentemente, uma estratégia denominada "formalização da informalidade" é aplicada com sucesso em ambientes ágeis. Se algo estiver funcionando bem, não há necessidade de alterar isso em prol do TMMi. Contudo, documentado, pode ser ensinado e compartilhado com outras pessoas, o que geralmente é benéfico. O que se entende aqui com a estratégia de “formalização da informalidade” é que, se existe um processo que funciona, mas é informal de certas maneiras, pode-se ensinar e documentá-lo exatamente como está sendo executado. Ser leve como um processo significa que nem todos os cenários de uso possíveis são abordados em sua descrição. Portanto, esses processos devem ser apoiados por orientação e assistência no trabalho, especialmente durante o período da implantação inicial. Como resultado, trazendo a maturidade do processo de teste para uma organização Agile, mantendo a cultura Agile, também é necessário mais - e não menos - treinamento. Da mesma forma, durante uma avaliação, o foco para reunir evidências mudará para fazer mais entrevistas em vez de estudar artefatos.

Os métodos usados para propor melhorias no processo de teste ao usar o Agile tenderão a se concentrar em métodos analíticos para avaliar as causas-raiz dos problemas, como diagramas de causa-efeito. Esses são métodos particularmente úteis para a mentalidade de solução de problemas, importante no final de uma iteração.

2 TMMi Nível 2 Gerenciado

2.1 Área de Processo 2.1 Política e Estratégia de Teste

O objetivo da área de processo de Política e Estratégia de Teste é desenvolver e estabelecer uma política de teste e uma estratégia de teste em toda a organização ou em todo o programa em que as atividades de teste, por exemplo, tipos e quadrantes de teste, sejam definidas sem ambiguidade. Para medir o desempenho do teste, o valor das atividades de teste e expor as áreas para melhoria, são introduzidos indicadores de desempenho do teste.

2.1.1 SG1 Estabelecer uma Política de Teste

Qualquer organização que embarque em um projeto de melhoria de teste deve começar definindo uma política de teste. A política de teste define os objetivos, metas e visões estratégicas gerais da organização em relação aos profissionais de teste. É importante que a política de teste esteja alinhada com a política geral de negócios (qualidade) da organização. As melhorias no teste devem ser orientadas por objetivos comerciais claros, que por sua vez devem ser documentados na política de teste (melhoria). É necessária uma política de teste para atingir uma visão comum dos testes e seus objetivos entre todos os envolvidos em uma organização. Essa visão comum é necessária para alinhar as atividades de teste (melhoria de processo) em toda a organização. Observe que os objetivos do teste nunca devem ser objetivos sozinhos, eles derivam do objetivo de nível superior para estabelecer a qualidade do software e do produto em funcionamento.

O exposto acima também é válido em uma organização que pratica o Desenvolvimento Ágil de Software. De fato, em muitas organizações, há muita discussão sobre a mudança do papel dos testes, independência dos testes, automação de testes e testadores profissionais no Desenvolvimento Ágil de Software. Esses itens e outros normalmente são problemas que devem ser abordados em uma discussão com a gerência e os stakeholders e documentados em uma política de teste. Qualquer organização, incluindo aquelas que praticam o Agile, que desejam iniciar um projeto de melhoria de teste precisa identificar e definir os direcionadores de negócios e as necessidades de tal iniciativa. Por que mais iniciar um projeto de melhoria? Ao dedicar tempo para capturar as verdadeiras necessidades de negócios, é possível fornecer um contexto para tomar decisões sobre onde concentrar as prioridades de melhoria (teste), por exemplo, em qual área do processo. Observe que uma política de teste geralmente é um documento enxuto de uma página, página da web ou gráfico de parede no nível organizacional, e não um documento no nível do projeto.

O objetivo específico do TMMi em estabelecer uma política de teste, incluindo suas práticas específicas, é totalmente aplicável a organizações que aplicam o Desenvolvimento Ágil de Software. Obviamente, os elementos de uma política de teste também podem ser incorporados a uma política de desenvolvimento. Não há nenhum requisito específico do TMMi para que seja um documento separado. A política de desenvolvimento em uma organização que aplica o Desenvolvimento Ágil de Software pode, por exemplo, identificar o *Scrum* como sua estrutura de gerenciamento, o XP como método principal sendo usado e a aderência aos valores Agile como princípio principal. Outro princípio importante que poderia ser mencionado é que todos na equipe são responsáveis por tudo, também a qualidade do produto é uma responsabilidade da equipe.

No entanto, há uma consideração importante para o Agile em relação à política de teste e às metas especialmente definidas de teste (melhoria). Embora possa haver metas gerais relacionadas à melhoria do processo de teste na organização, isso precisa ser equilibrado com projetos individuais e as equipes Agile sendo responsáveis por melhorar seu próprio processo. O desafio do aprimoramento do processo Agile é orientar e enquadrar o aprimoramento no nível da organização, sem reduzir o senso de propriedade de seu próprio processo por uma equipe Agile individual.

2.1.2 SG2 Estabelecer uma Estratégia de Teste

A estratégia de teste segue a política de teste e serve como ponto de partida para as atividades de teste nos projetos. Uma estratégia de teste é geralmente definida em toda a organização ou em todo o programa. Uma estratégia de teste típica é baseada em uma avaliação de risco de alto nível do produto e incluirá uma descrição dos tipos de teste, quadrantes de teste e níveis de teste a serem executados, por exemplo: teste de unidade, aceite, regressão e desempenho. Não basta dizer, por exemplo, que dentro de uma unidade de iteração, serão realizados testes de aceite e aceite. Precisamos definir o que se entende por teste de unidade e aceite, como estas são tipicamente realizadas e quais são seus principais objetivos. A experiência mostra que, quando uma estratégia de teste é definida e seguida, é provável que ocorra menos sobreposição entre as várias atividades de teste, levando a um processo de teste mais eficiente. Além disso, como os objetivos e a abordagem dos vários tipos e níveis de teste

estão alinhados, é provável que haja menos furos, o que leva a um processo de teste mais eficaz e, portanto, a um nível mais alto de qualidade do produto. Para estabelecer esse alinhamento, é altamente recomendável cobrir o teste Agile e qualquer teste não Agile que esteja ocorrendo no mesmo projeto sob uma estratégia de teste abrangente. No entanto, se houver projetos Agile e não-Agile separados, normalmente existem duas estratégias de teste.

Uma estratégia de teste é um documento vital dentro de um ambiente ágil. Ele define em alto nível o teste a ser realizado nas equipes Agile (equipes de iteração), que tipos de teste, quadrantes de teste e níveis de teste são executados e, em alto nível, sua abordagem. O documento descreve como os testes são organizados, por exemplo, quais testes ocorrem dentro das equipes do Agile e quais testes são realizados fora. Ele definirá o relacionamento das equipes Agile com os níveis de teste executados fora de seu escopo, por exemplo, teste de integração de hardware ou software, teste de integração de sistema ou teste beta. Uma estratégia de teste garante que todos os envolvidos no teste compreendam o panorama geral do teste.

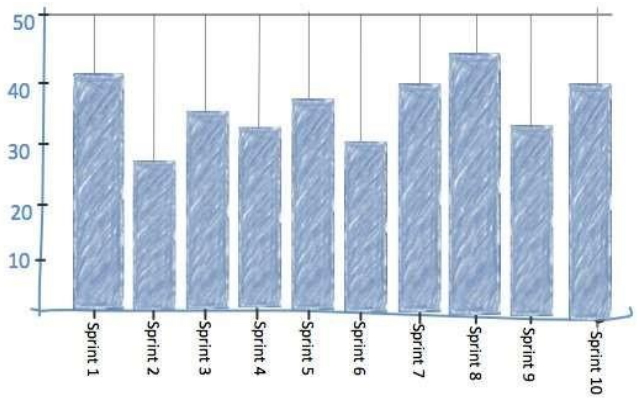
O documento enxuto de estratégia de teste geralmente é uma boa solução para a organização Agile e uma saída dos planos de teste detalhados (do projeto). Durante o planejamento da liberação, a estratégia de teste disponível em toda a organização ou em todo o programa é discutida e confirmada ou uma estratégia de teste derivada é criada especificamente para o projeto. A estratégia de teste confirmada ou criada fornece uma estrutura de teste que abrange todas as iterações.

O objetivo específico do TMMi em estabelecer uma estratégia de teste, incluindo suas práticas específicas, é totalmente aplicável a organizações que aplicam o Desenvolvimento Ágil de Software. Observe que atividades ativas de "distribuição", por exemplo, para o documento de estratégia de teste, podem ser menos relevantes no ágil. Aqueles que são partes interessadas já devem ter se envolvido em discussões anteriores como parte da abordagem de toda a equipe. No entanto, uma estratégia de teste não é de nível de equipe, mas de nível superior. Toda a equipe opera no nível da equipe, para que um documento organizacional ou do programa ainda precise ser distribuído para as equipes e, de fato, para os stakeholders, por exemplo, aqueles que executam atividades de teste fora da equipe, se existir um modelo híbrido.

2.1.3 SG3 Estabelecer indicadores de desempenho de teste

Os objetivos de negócios para aprimoramento de teste, conforme definidos na política de teste, precisam ser traduzidos em um conjunto de indicadores-chave de desempenho de teste. A política de teste e os indicadores de desempenho que acompanham fornecem uma direção clara e um meio de comunicar os níveis esperados e alcançados de desempenho do teste. Os indicadores de desempenho devem indicar o valor do teste e a melhoria do processo de teste para os stakeholders. Como os investimentos em melhoria de processos precisam de suporte de gerenciamento de longo prazo, é crucial medir quantitativamente os benefícios de um programa de melhoria para mantê-los motivados. Lembre-se de que esse objetivo específico do TMMi é definir um número limitado (p.ex.: 2 ou 3) indicadores de desempenho do teste. Não se trata de configurar e implementar um programa de medição completo, mas de definir um conjunto principal de indicadores que informam como o valor dos testes está mudando ao longo do tempo e em diferentes ambientes de entrega.

No Agile, o foco será mais baseado em equipe e pensamento sistêmico. Isso pode resultar em uma ampliação correspondente dos indicadores para a equipe e para o sistema geral, em vez de se limitar apenas às especificidades do próprio teste. Além disso, os indicadores no nível 2 do TMMi estão principalmente relacionados aos resultados finais das iterações. Os exemplos incluem defeitos perdidos, velocidade, índices de satisfação do cliente, esforço ou desperdício, porcentagem de automação de teste etc. realizações de um programa de aprimoramento de teste baseado no TMMi.



O objetivo específico do TMMi em estabelecer indicadores de desempenho de teste, incluindo suas práticas específicas, é totalmente aplicável, embora os indicadores de desempenho selecionados e aplicados possam ter um escopo maior e ser mais amplo do que estar relacionado apenas ao teste. Obviamente, este último tornará a análise e interpretação dos indicadores de desempenho mais desafiadora. De fato, os indicadores de desempenho que estão sendo usados podem não ser chamados de indicadores de desempenho de teste, mas de indicadores de desempenho da equipe ou de desempenho do sistema. Isso ainda está correto no contexto do TMMi, desde que tenha elementos relacionados ao teste e esteja sendo usado para avaliar o progresso que está sendo feito na melhoria do teste.

2.2 Área de Processo 2.2 Planejamento de Teste

O objetivo do Planejamento de Teste é definir uma abordagem de teste com base nos riscos identificados e na estratégia de teste definida, e estabelecer e manter planos bem fundamentados para executar e gerenciar as atividades de teste.

Cuidado para que a chave para o planejamento bem-sucedido do teste esteja no pensamento inicial ("a atividade"), não na definição do plano de teste associado ("o documento").

Para os ciclos de vida Agile, geralmente ocorrem dois tipos de planejamento: liberar o planejamento e planejar a iteração. A área de processo Planejamento de Teste no TMMi nível 2 concentra-se nas atividades relacionadas ao teste no planejamento de liberação e de iteração. O planejamento de liberação antecipa o lançamento de um produto no início de um projeto. O planejamento de liberação requer uma lista de pendências definida do produto e pode envolver o refinamento de histórias de usuários maiores em uma coleção de histórias menores. O planejamento de liberação fornece a base para uma abordagem de teste e um plano de teste que abrange todas as iterações. Os planos de liberação são de alto nível. Após o planejamento da liberação, o planejamento da iteração é iniciado. O planejamento da iteração antecipa o final de uma única iteração e se preocupa com o backlog da iteração.

Observe que a área de processo Planejamento de Teste possui várias práticas específicas. O TMMi não indica quando ou como conduzir essas práticas. Também não é possível planejar de forma incremental. A abordagem tradicional tem sido a de solidificar o máximo de decisões possíveis, para que o custo e o cronograma relacionados também possam ser solidificados. A lógica dessa abordagem foi estimar melhor o trabalho e reduzir o risco de fluência no escopo. As abordagens ágeis geralmente assumem a posição de que agregamos maior valor ao aprimoramento contínuo do plano com base nas informações mais recentes e na colaboração contínua com o cliente.

2.2.1 SG1 Executar avaliação de risco

Os testes exaustivos são impossíveis, e sempre é preciso fazer escolhas e definir prioridades. Portanto, este objetivo do TMMi também é aplicável a projetos Agile. Para projetos Agile, uma avaliação de risco de alto nível do produto deve ser realizada com base em um documento de visão do produto ou em um conjunto de histórias de usuário de alto nível no planejamento da liberação. Para cada iteração, uma sessão de risco do produto mais detalhada deve ser realizada com base nas histórias do usuário ou em outros requisitos para essa iteração como parte da sessão de planejamento da iteração. O processo de avaliação de risco do produto em um projeto Agile terá um formato muito mais leve comparado aos aplicados em projetos tradicionais, seguindo um modelo de ciclo de vida sequencial. Um exemplo de uma técnica de risco de produto leve a ser usada é o *risk poker* [Van Veenendaal].

No planejamento da versão, os representantes comerciais que conhecem os recursos da versão fornecem uma visão geral de alto nível da funcionalidade a ser desenvolvida, e toda a equipe, incluindo os testadores, ajudará na identificação e avaliação de riscos.

Durante o planejamento da iteração, a equipe do Agile identifica e analisa os riscos do produto com base nas histórias do usuário a serem implementadas na próxima iteração. De preferência, todos os membros da equipe Agile e possivelmente alguns outros stakeholders participam da sessão de risco do produto. O resultado é uma lista priorizada de itens de risco do produto, identificando as áreas críticas para teste. Isso, por sua vez, ajudará a determinar a quantidade apropriada de esforço de teste a ser alocado, a fim de cobrir cada risco com testes suficientes e a sequência desses testes de maneira a otimizar a eficácia e a eficiência do trabalho de teste a ser realizado. As tarefas estimadas no painel de tarefas podem ser priorizadas em parte com base no nível de risco do produto associado a elas. As tarefas associadas a riscos mais altos devem começar mais cedo e envolver mais esforço de teste. Tarefas associadas a riscos mais baixos devem começar mais tarde e envolver menos esforço de teste.

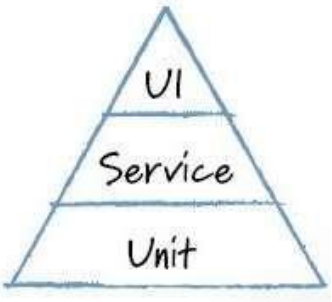
2.2.2 SG2 Estabelecer uma abordagem de teste

Uma abordagem de teste é definida para mitigar os riscos identificados e priorizados do produto. Para uma iteração específica, os itens e recursos a serem testados são identificados durante o planejamento da iteração. Essa atividade também é baseada no resultado da sessão de risco do produto. A lista priorizada de itens a serem testados geralmente se refere às histórias de usuário a serem testadas nesta iteração. Os recursos geralmente se relacionam, entre outros, às várias características de qualidade do software a serem testadas. O risco de novos produtos pode se tornar aparente durante a iteração, exigindo testes adicionais. Questões como riscos de novos produtos que exigem testes adicionais geralmente são discutidas em reuniões levantamento diárias.

A abordagem de teste definida no nível da iteração para mitigar os riscos pode cobrir, por exemplo, revisão adicional de histórias de usuários e critérios de aceite, esforço de teste proporcional ao nível de risco, a seleção de técnicas

de teste apropriadas com base no nível e tipo de risco. A abordagem de teste no nível do release estará em um nível muito mais alto e será baseada na estratégia de teste definida no nível do programa ou da organização. Frequentemente, uma abordagem de teste é realizada ou exibida no wiki da equipe ou projeto.

Um risco importante que é sempre aparente no desenvolvimento iterativo é o risco de regressão. A abordagem de teste precisa definir como o risco de regressão é gerenciado. Normalmente, isso será feito criando um conjunto de testes de regressão específico, que é preferencialmente automatizado. Nesse contexto, a pirâmide de automação de teste [Cohn] é útil. Ele mostra como maximizar o valor da automação de testes de regressão, começando com testes de unidade no nível mais baixo da pirâmide e passando para o teste de nível de serviço. O teste da interface do usuário fica no topo. Os testes de unidade são rápidos e confiáveis. A camada de serviço permite testar a lógica de negócios no nível da API ou de serviço, onde você não está sobrecarregado pela interface do usuário (UI). Quanto mais alto o nível, mais lento e mais quebradiço o teste se torna.



Os critérios de entrada, normalmente parte de uma abordagem de teste definida (prática específica 2.3), provavelmente não são relevantes para o desenvolvimento Agile. No Desenvolvimento Ágil de Software, o teste é parte integrante do processo da equipe e uma atividade quase contínua. Consequentemente, não há necessidade de uma lista de verificação ou gateway específico para determinar se o teste pode ou não ser iniciado. Isso também se aplica a um componente que entra em uma equipe Agile de um estágio de teste (p.ex.: teste de unidade) para outro (p.ex.: teste de aceite).

O teste dos critérios de saída (prática específica 2.4) faz parte da chamada Definição de Concluído (DoD - Definition of Done). É importante que o DoD tenha critérios específicos relacionados ao teste, por exemplo, para cobertura de teste e qualidade do produto (defeitos). A iteração deve resultar na implementação do conjunto acordado de histórias de usuários e atender aos critérios de saída (teste), conforme definido no DoD. Normalmente, as histórias que não cumprem os critérios de saída são colocadas na lista de pendências e podem ser tratadas na próxima iteração. Obviamente, no Agile, não há critérios de saída para um componente que vai de um estágio de teste para outro. Além de existir uma Definição de Concluído no nível da iteração, também existe um DoD no nível do release, abrangendo várias iterações. O DoD no nível do release normalmente terá novamente critérios relacionados à cobertura e à qualidade do produto.

Prática específica 2.5 Definir critérios de suspensão e retomada provavelmente não é relevante para os ciclos de vida do Agile. Como o teste é parte integrante do processo de Desenvolvimento Ágil de Software, é claro que não será tratado como atividade separada e independente de outras atividades de iteração. Quando há problemas de bloqueio que podem ser considerados ameaças potenciais ou reais ao progresso dos testes, eles são discutidos na reunião de levantamento diária. Neste fórum, a equipe decidirá quais ações, se houver, devem ser tomadas para resolver os problemas. Assim, os critérios formais de suspensão e retomada não serão exigidos e definidos, os problemas relacionados a isso serão tratados como parte da rotina normal do Agile. A rotina Agile, portanto, serve como uma prática alternativa para essa prática específica.

2.2.3 SG3 Estabelecer estimativas de teste

Para equipes Agile, estimativas detalhadas para testes serão feitas durante o planejamento da iteração. Estimativas de alto nível (teste) são feitas durante o planejamento da liberação e, possivelmente, também nas sessões de refinamento da lista de pendências. É claro que todas as estimativas são feitas como uma estimativa da equipe, que inclui todo o esforço necessário para entregar cada história. É importante garantir que as atividades de teste sejam realmente levadas em consideração durante as sessões de estimativa. Isso pode ser feito com a identificação de atividades de teste como tarefas separadas e a estimativa subsequente de cada tarefa de teste individual ou a estimativa de cada história de usuário em que o teste a ser realizado é explicitado e, portanto, levado em consideração. Um testador, fazendo parte da equipe Agile, participará das sessões de estimativa. O *Planning Poker* ou *shirt-size* são técnicas típicas de estimativa usadas no Desenvolvimento Ágil de Software.

O trabalho a ser feito para a próxima iteração geralmente é definido pelas histórias do usuário. As histórias de usuários precisam ser pequenas em tamanho para serem estimadas. Estimável é um dos critérios da história do usuário definidos pelo INVEST [Wake] e é aplicável às histórias do usuário que devem fazer parte de uma

iteração. Durante o planejamento da iteração, as equipes Agile geralmente identificam e definem tarefas relacionadas ao teste. As tarefas de teste serão capturadas em um painel de tarefas junto com as outras tarefas de desenvolvimento. Esse conjunto de tarefas é a base para a estimativa da iteração. O conjunto de tarefas definidas para a próxima iteração serve como uma espécie de estrutura de interrupção do trabalho (usada em projetos sequenciais).

No planejamento da versão, as histórias de usuários ou épicas geralmente são definidas com mais alto nível e ainda não são detalhadas em tarefas específicas. Obviamente, isso tornará as estimativas mais difíceis e menos precisas. Conforme declarado, os projetos ágeis não estabelecerão uma estrutura de detalhamento do trabalho como base para estimativas, mas poderão, no nível da liberação, se beneficiar de um diagrama simples que visualiza o produto a ser desenvolvido e, portanto, escopo adequadamente o trabalho.

Embora uma equipe do Agile faça uma estimativa de maneira relativamente informal, a justificativa para as estimativas deve ser clara (ou seja, quais fatores estão sendo considerados). Discussões baseadas na lógica promovem um nível mais alto de precisão das estimativas. Normalmente, em projetos Agile, a estimativa é focada no tamanho (usando pontos da história) ou esforço (usando dias de trabalho ideais como uma unidade de estimativa). Os custos normalmente não são abordados como parte das sessões de estimativa em projetos Agile.

Esse objetivo específico do TMMi e suas práticas específicas são, portanto, totalmente aplicáveis, com exceção da prática específica. 3.2 Definir o ciclo de vida do teste. Um dos princípios básicos do desenvolvimento iterativo e de software Agile é trabalhar em pequenos blocos. Portanto, as tarefas que foram identificadas geralmente são detalhadas o suficiente para servir de base para a estimativa (teste). Portanto, não é necessário no Agile definir também um ciclo de vida das atividades de teste para servir como base adicional para estimativa.

2.2.4 SG4 Desenvolver um plano de teste

Substituindo o comentário de que o planejamento do teste é sobre o pensamento inicial ("a atividade") e não sobre a definição do plano de teste associado ("o documento"). No Agile, a maioria das atividades de planejamento de teste, conforme definido por este objetivo específico do TMMi, será executada durante o planejamento de liberação e iteração. No entanto, o resultado dessas atividades normalmente não será documentado em um plano de teste, especialmente no planejamento da iteração, onde elas podem ser refletidas no quadro de tarefas.

Como o teste é realizado como parte integrante do planejamento de liberação e iteração, os "agendamentos" resultantes também incluirão atividades de teste. Em vez de um cronograma detalhado, como desenvolvido com ciclos de vida sequenciais, o cronograma em um projeto Agile é muito mais como um pedido de histórias de usuário (itens de lista de pendências) e tarefas que refletem as prioridades de liberação e iteração, por exemplo, com base na entrega desejada de valor comercial. Mapas mentais são frequentemente usados aqui como uma técnica de apoio. O quadro de tarefas refletirá as prioridades da iteração. Portanto, nenhum cronograma explícito (teste) é estabelecido, espera-se que prioridades claras de liberação e iteração sejam definidas para as histórias de usuários, respectivamente as tarefas a serem executadas, incluindo as tarefas de teste.

No nível do projeto, a equipe de teste faz parte da construção da equipe, a necessidade de teste ou recursos com várias qualificações é identificada antecipadamente. À medida que os projetos mudam ou crescem, pode-se esquecer facilmente a justificativa para a seleção inicial de um indivíduo para uma determinada equipe ou projeto, que geralmente inclui necessidades ou experiências específicas de habilidades relacionadas especificamente à equipe ou projeto. Isso fornece uma boa justificativa para anotar as necessidades de habilidades das pessoas, fornecendo informações de backup relacionadas ao motivo pelo qual elas foram selecionadas para a equipe / projeto. Uma vez definidas as equipes do Agile, a equipe de teste fica mais ou menos fixa. Durante o planejamento da iteração, a identificação dos recursos e habilidades (adicionais), por exemplo, para testes não funcionais, necessários para executar o teste em uma iteração, se necessário, pode ser discutida para garantir que a equipe tenha recursos, conhecimentos e habilidades de teste suficientes para executar os testes necessários.

O *Scrum master* deve garantir que o proprietário do produto forneça informações para os testes, conforme necessário, por exemplo, respondendo a perguntas e conversando sobre histórias de usuários. O proprietário do produto deve estar disponível o suficiente, o que deve ser novamente organizado antecipadamente.

Uma sessão inicial de risco do projeto deve fazer parte do planejamento de liberação e iteração. A identificação (e gerenciamento) de outros riscos do projeto durante a iteração faz parte das reuniões de levantamento diárias e normalmente são documentadas por meio de um log de impedimentos. É importante que também sejam notados problemas de teste no log de impedimentos. Os impedimentos devem ser discutidos na reunião de levantamento diária, até que sejam resolvidos.

Embora nenhum plano de teste específico e detalhado seja desenvolvido e documentado, a prática específica 4.5 Estabelecer o plano de teste ainda é relevante dentro de um contexto Agile. No entanto, é provável que o resultado

das discussões que ocorrem no contexto do planejamento de testes seja capturado de forma leve, possivelmente um mapa mental.

2.2.5 SG5 Obter compromisso com o plano de teste

No Agile, o processo para desenvolver e estabelecer uma abordagem de teste e o plano de teste é um exercício baseado em equipe, possível liderança por um profissional de teste (sendo um dos membros da equipe). A qualidade do produto é uma responsabilidade da equipe. Assim, desde que a equipe siga o processo correto, o comprometimento com a abordagem e o plano já é um resultado implícito do planejamento de liberação e iteração, pois é um esforço da equipe. É claro que isso é uma grande diferença na maneira de trabalhar em um ambiente tradicional, onde normalmente o testador prepara o plano de teste e, posteriormente, precisa obter um compromisso explícito.

Nos projetos Agile, a equipe (incluindo o proprietário do produto) deve entender e concordar com a lista priorizada de riscos do produto e as ações de mitigação de teste a serem executadas. O entendimento e o comprometimento podem, por exemplo, ser alcançados por meio de uma breve apresentação seguida de uma discussão durante o planejamento de liberação ou iteração, explicando os riscos do produto, a abordagem de teste e sua lógica para a equipe.

Durante a sessão de estimativa (consulte SG 3 Estabelecer estimativas de teste), a carga de trabalho é estimada. Os recursos (teste) para uma equipe são corrigidos pela configuração da equipe Agile. As histórias de usuários estimadas e selecionadas para serem desenvolvidas tomam os recursos disponíveis como ponto de partida (restrição). Portanto, conciliar novamente os níveis de trabalho e de recursos não é uma atividade significativa. A prática específica 3.2 Reconciliar níveis de trabalho e recursos é, portanto, uma prática que normalmente não é relevante em um contexto Agile. As reuniões de levantamento diárias serão usadas para resolver quaisquer problemas de recursos durante a iteração e (re) alocar recursos apropriados imediatamente ou remover um produto final de uma iteração para ser reconciliado no planejamento de iteração ou lançamento futuro.

2.3 Área de Processo 2.3 Monitoramento e Controle de Teste

O objetivo do Monitoramento e Controle de Teste é fornecer um entendimento do progresso do teste e da qualidade do produto, para que ações corretivas apropriadas possam ser tomadas quando o progresso do teste se desvia significativamente do plano e a qualidade do produto se desvia significativamente das expectativas.

Seguindo o manifesto Agile e seus princípios associados, existem algumas coisas que são fundamentalmente diferentes para monitorar e controlar projetos Agile em comparação com projetos tradicionais. Embora o monitoramento e o controle sejam elementos essenciais de um projeto Agile, isso não implica que o objetivo seja seguir um plano rígido, de fato, o oposto é verdadeiro, pois o manifesto e os princípios falam em acolher mudanças. Em um contexto ágil, o Teste de Monitoramento e Controle poderia ser interpretado como fornecendo práticas recomendadas para ajustar continuamente o plano para mantê-lo atualizado, o que as abordagens ágeis recomendam.

De uma perspectiva de monitoramento e controle de teste, isso significa que não somos orientados a planejar, mas revisamos constantemente nosso progresso e resultados dos testes e adaptamos nosso plano e abordagem conforme apropriado - os riscos de novos produtos podem se tornar aparentes. O plano de teste, por mais que seja capturado, é uma entidade viva e precisa ser revisado e atualizado continuamente à medida que novas informações são apoiadas ou são fornecidas informações. Projetos ágeis também precisam manter em mente a 'imagem maior', monitorar e controlar no nível mais alto (release) e no nível da iteração.

É importante observar que, como o teste é um processo totalmente integrado ao processo geral da equipe do Agile, o monitoramento e o controle de teste também são parte integrante dos mecanismos gerais de monitoramento e controle da equipe do Agile. Como resultado, os testadores não se reportam a um gerente de teste como nos projetos tradicionais, mas à equipe.

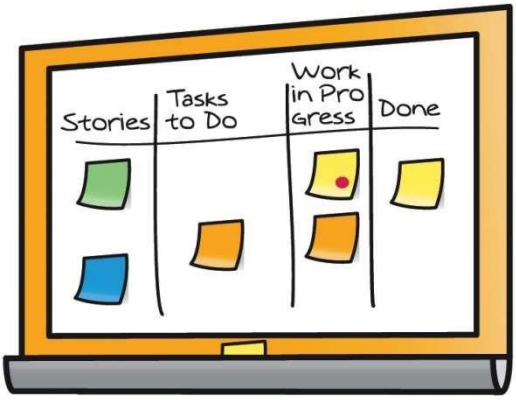
Como a área de processo Planejamento de Teste no TMMi nível 2 se concentra no planejamento de liberação e de iteração, o mesmo se aplica ao monitoramento e controle de teste. Espera-se que o monitoramento e controle do teste sejam realizados no nível de planejamento e liberação.

2.3.1 SG1 Monitorar o progresso do teste em relação ao plano

Os testadores nas equipes do Agile utilizam vários métodos para monitorar e registrar o progresso do teste, por exemplo, a progressão das tarefas e histórias de teste no painel de tarefas do Agile e nos gráficos de *burndown*.

Eles podem ser comunicados ao restante da equipe usando mídias como painéis de *wiki* e e-mails em formato de painel, bem como verbalmente durante reuniões de *stand-up*. As equipes podem usar gráficos *burndown* para rastrear o progresso de todo o *release* e dentro de cada iteração. Um gráfico *burndown* normalmente mostra o progresso em relação à velocidade esperada e ao atraso dos recursos a serem implementados (desempenho da equipe). Às vezes, quando os recursos do ambiente de teste são escassos e vitais, por exemplo, para testes não funcionais, gráficos *burndown* específicos são usados para monitorar o uso dos recursos do ambiente de teste em relação aos acordados durante o planejamento da iteração. Outra prática comum é rastrear problemas do ambiente de teste por meio do quadro de tarefas, por exemplo, usando adesivos marcados como 'ambiente de teste bloqueado' nos cartões de histórias ou criando uma coluna separada no quadro onde todas as histórias bloqueadas pelos ambientes aguardam até serem desbloqueadas. É tudo sobre como criar visibilidade do impacto de um ambiente de teste que está bloqueando o progresso.

Para fornecer uma representação visual detalhada e instantânea do status atual de toda a equipe, incluindo o status dos testes, as equipes podem usar os painéis de tarefas Agile. Os cartões de histórias, tarefas de desenvolvimento, tarefas de teste e outras tarefas criadas durante o planejamento da iteração são capturados no painel de tarefas, geralmente usando cartões coordenados por cores para determinar o tipo de tarefa. Durante a iteração, o progresso é gerenciado através do movimento dessas tarefas pelo quadro de tarefas em colunas como executar, trabalhar em andamento e concluído. As equipes do Agile podem usar ferramentas para manter seus cartões de histórias nos painéis de tarefas do Agile, que podem automatizar painéis e visão geral do status. No entanto, algumas equipes não criam tarefas específicas para as atividades individuais, mas podem apenas usar o cartão de história e fazer anotações nos comentários deste cartão para testes, fazendo referência a ferramentas Agile ou *wiki* onde os testes podem ser documentados. As tarefas de teste no painel de tarefas geralmente estão relacionadas aos critérios de aceite definidos para as histórias de usuários. À medida que os *scripts* de automação de teste, testes com *script* manual e testes exploratórios para uma tarefa de teste atingem o status de aprovação, a tarefa passa para a coluna concluída do painel de tarefas.



A definição de concluído (DoD - Definition of Done) serve como critério de saída em relação ao qual o progresso está sendo medido. O DoD também deve estar relacionado às atividades de teste e mostra todos os critérios que precisam ser atendidos antes que o teste de uma história de usuário possa ser chamado de 'Concluído'. Observe que os critérios de teste relacionados às tarefas de teste fazem apenas parte do que a equipe concorda em concluir como a definição de concluído. A definição de critérios concluídos é normalmente aplicada em vários níveis, por exemplo, no nível de iteração e no nível de liberação. Toda a equipe analisa o status do quadro de tarefas regularmente, geralmente durante as reuniões de levantamento diárias, para garantir que as tarefas se movam pelo quadro a uma taxa aceitável. Se alguma tarefa (incluindo tarefas de teste) não estiver se movendo ou estiver se movendo muito lentamente, isso desencadeia uma discussão em equipe, na qual são analisados os problemas que podem estar bloqueando o progresso dessas tarefas.

A reunião de levantamento diária inclui todos os membros da equipe Agile, incluindo testadores. Nesta reunião, eles comunicam seu status atual e progresso real ao restante da equipe. Quaisquer problemas que possam bloquear o progresso do teste são comunicados durante as reuniões diárias, para que toda a equipe esteja ciente dos problemas e possa agir de acordo. Dessa maneira, o gerenciamento de riscos também é integrado nessas reuniões diárias. Qualquer risco do projeto, incluindo os de teste, por exemplo, a falta de disponibilidade nos ambientes de teste, pode ser comunicado e resolvido durante o stand-up diário. (Observe que o monitoramento de riscos do produto faz parte do monitoramento da qualidade do produto e, portanto, discutido a seguir com o objetivo específico SP 2 Monitorar a qualidade do produto em relação ao plano e às expectativas.) técnicas comprovadas que atendem ao objetivo das práticas específicas do TMMi na área de processo de Monitoramento e Controle de Testes.

A revisão do marco no Agile está na conclusão de uma iteração. As realizações dos testes, por exemplo, em relação ao DoD, farão parte da revisão da iteração. As demonstrações são organizadas com os stakeholders para discutir o valor comercial e a qualidade do produto que está sendo entregue.

Os stakeholders são representadas pelo proprietário do produto no planejamento da iteração, nas revisões da iteração (demos) e nas retrospectivas. O proprietário do produto é envolvido por meio de discussões sobre o backlog do produto e fornecerá feedback e informações para a elaboração de histórias de usuários e a modelagem de testes ao longo da iteração. Outras partes interessadas estão envolvidas no final de cada iteração durante a revisão da iteração. Nenhum monitoramento específico do envolvimento das partes interessadas é necessário, pois a representação das partes interessadas por um proprietário do produto é incorporada à maneira Agile de trabalhar. Observe que o envolvimento ativo do proprietário do produto em projetos Agile é um fator crucial de sucesso, mas

às vezes é bastante difícil de realizar na prática. Se este for o caso, isso será relatado e discutido diariamente e gerenciado como um risco do projeto (veja acima) que afetará a eficácia e a eficiência de toda a equipe.

2.3.2 SG2 Monitorar a qualidade do produto em relação ao plano e expectativas

Para o monitoramento da qualidade do produto, em grande parte, os mesmos mecanismos são usados para o monitoramento do progresso (consulte o SG1 acima). No Agile, para monitorar o risco do produto, o foco está na revisão da lista de riscos do produto em reuniões regulares, e não na revisão de qualquer documentação detalhada dos riscos. Os riscos do produto recentemente identificados ou os riscos alterados do produto, por exemplo, como resultado de testes exploratórios, serão discutidos e as ações de teste necessárias serão acordadas. O status dos vários riscos do produto geralmente é mostrado por meio de gráficos no painel.

Uma prática recomendada é que nenhum recurso seja considerado concluído até que tenha sido integrado e testado com sucesso com o sistema. Em alguns casos, as iterações de proteção ou estabilização ocorrem periodicamente para resolver quaisquer defeitos remanescentes e outras formas de dívida técnica. As equipes ágeis usam métricas baseadas em defeitos semelhantes às capturadas nas metodologias de desenvolvimento tradicionais, como taxas de aprovação / reprovação de teste, taxas de descoberta de defeitos, defeitos encontrados e corrigidos, para monitorar e melhorar a qualidade do produto. O número de defeitos encontrados e resolvidos durante uma iteração, bem como o número de defeitos não resolvidos que possivelmente fazem parte do backlog da próxima iteração devem ser monitorados durante as reuniões de levantamento diárias. Para monitorar e melhorar a qualidade geral do produto, muitas equipes Agile também usam pesquisas de satisfação do cliente para receber feedback sobre se o produto atende às expectativas do cliente.

Os critérios de saída de teste, por exemplo, para cobertura de teste e qualidade do produto (defeitos), fazem parte da Definição de Concluído (DoD - Definition of Done). A adesão aos critérios de saída acordados é normalmente monitorada através do painel de tarefas, pelo qual uma história só pode ser indicada como "concluída" se estiver em conformidade com os critérios do DoD.

A reunião de levantamento diária é o mecanismo usado para executar quase continuamente análises de qualidade do produto. A revisão importante da qualidade do produto no Agile está na conclusão de uma iteração. As demonstrações são organizadas com os stakeholders para discutir o valor comercial e a qualidade do produto que está sendo entregue. A qualidade do produto é verificada e validada de acordo com os critérios de qualidade definidos pelo DoD.

Após a área de processo Planejamento de Teste, as práticas específicas no monitoramento dos critérios de entrada, suspensão e retomada provavelmente não serão relevantes. Para obter mais informações, consulte a SG 2 Estabelecer uma abordagem de teste da área de processo Planejamento de teste, em que foi dada uma explicação do motivo pelo qual esses critérios geralmente não são relevantes em um ambiente Agile.

2.3.3 SG3 Gerenciar as ações corretivas para encerramento

As equipes ágeis notariam rapidamente problemas como desvios das expectativas em um gráfico burndown ou falta da progressão de tarefas (de teste) e histórias no quadro de tarefas do Agile. Esses e outros problemas, por exemplo, problemas que bloqueiam o progresso dos testes, são comunicados durante as reuniões de *stand-up* diárias (veja acima), para que toda a equipe esteja ciente dos problemas. Isso desencadeia uma discussão em equipe, na qual os problemas são analisados. O resultado pode ser que a linha de base precise de uma atualização (p.ex.: remoção de uma ou mais histórias de usuários da lista de pendências da iteração), a Definição de Concluído talvez seja muito rigorosa ou que mudanças devam ser feitas na maneira de trabalhar. A equipe decidirá sobre as ações corretivas a serem tomadas em conjunto com o proprietário do produto. Nos casos em que o *backlog* da iteração é alterado, o proprietário do produto apresenta um *backlog* atualizado da iteração para a equipe e os itens removidos do *backlog* da iteração são levados para a próxima iteração e discutidos durante o planejamento da iteração.

O gerenciamento de ações corretivas em projetos Agile é principalmente uma responsabilidade da equipe auto-organizada. A equipe pode definir e implementar ações corretivas apropriadas ou escalar qualquer problema como "impedimento" para o *scrum master*. O *scrum master* normalmente tem a responsabilidade de apoiar a equipe no gerenciamento de problemas até o encerramento. Eventos típicos para discutir e gerenciar ações corretivas são levantamentos diários e reuniões retrospectivas. As ações corretivas acordadas podem ser gerenciadas para serem encerradas como "tarefas" ou "itens de lista de pendências" por meio da lista de pendências do produto ou (dentro da iteração) pelo painel de tarefas.

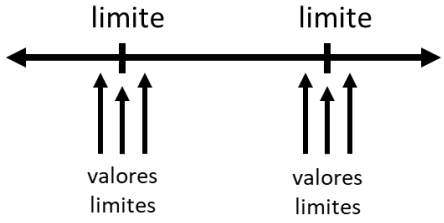
2.4 Área de Processo 2.4 Projeto e Execução de Teste

O objetivo da Modelagem e Execução de Testes é melhorar a capacidade do processo de teste durante a modelagem e a execução do teste, estabelecendo especificações de modelagem, utilizando técnicas de modelagem de teste, executando um processo estruturado de execução de teste e gerenciando incidentes de teste até o encerramento.

Embora o objetivo subjacente ("mitigar os riscos e testar o software") seja o mesmo para um projeto sequencial ágil e tradicional, a abordagem adotada sobre como testar é tipicamente muito diferente. No Agile, a flexibilidade e a capacidade de responder às mudanças são pontos de partida importantes. Além disso, a análise e a modelagem do teste, a implementação e a execução do teste não são fases de teste subsequentes, mas sim executadas em paralelo, sobrepostas e iterativamente. O nível de detalhe da documentação de teste estabelecida é outra diferença importante. Normalmente, as técnicas baseadas em defeitos e com mais experiência são usadas em projetos Agile, embora as técnicas de modelagem de teste baseadas em especificações também possam ser aplicáveis e utilizadas. Com mais ênfase nos testes de unidade e integração, técnicas de caixa branca, como teste de declaração e decisão, também são muito mais populares. Uma grande diferença final é o nível do risco de regressão, que exige mais testes de regressão nos vários níveis de teste. Idealmente, o teste de regressão é altamente automatizado. Existem muitas diferenças, mas, no final, no contexto da área de processa modelagem e execução de testes, é sempre sobre a criação de testes, riscos de produtos de mitigação, execução de testes e localização de defeitos.

2.4.1 SG1 Executar a análise e modelagem de teste usando técnicas de modelagem de teste.

No Agile, a análise, a modelagem e a execução de testes são atividades de suporte mútuo que normalmente são executadas em paralelo durante uma iteração. Em projetos de ciclo de vida sequencial, a análise de teste é realizada pelos testadores revisando a base de teste, por exemplo, os requisitos e avaliando a testabilidade da base de teste, uma vez criada. Nos projetos Agile, os testadores fazem parte de uma equipe que cria e refina, de forma colaborativa, histórias de usuários. Revisões informais frequentes são realizadas enquanto os requisitos estão sendo desenvolvidos, incluindo para cada história de usuário os critérios de aceite. Esses critérios são definidos em colaboração entre representantes de negócios, desenvolvedores e testadores. Normalmente, a perspectiva exclusiva do testador aprimora a história do usuário, identificando detalhes ausentes e tornando-os testáveis. Um testador pode contribuir fazendo perguntas abertas aos representantes de negócios sobre a história do usuário e seus critérios de aceite e propondo maneiras de testar a história do usuário. Portanto, a análise de teste não é uma atividade separada explícita, mas sim uma atividade implícita que os testadores realizam como parte de sua função no desenvolvimento colaborativo de histórias de usuários.



Com base na análise das histórias de usuários, as condições de teste são identificadas. De uma perspectiva de teste, a base de teste é analisada para ver o que poderia ser testado - essas são as condições de teste. As condições de teste (às vezes chamadas de situações de teste) são basicamente uma identificação de "coisas" que precisam ser testadas ou cobertas [Black, Van Veenendaal]. Desde que os critérios de aceite definidos sejam detalhados e claros o suficiente, podem assumir o papel das condições de teste tradicionais. As condições do teste são posteriormente convertidas em testes, necessários para fornecer cobertura dos critérios de aceite definidos e acordados. Muitas vezes, também é benéfico executar a análise de teste em um nível superior, em vez de apenas histórias de usuários. Por exemplo, analisando um recurso ou épico ou uma coleção de histórias para identificar condições de teste mais abstratas do que aquelas no nível da história do usuário e também abrangem várias histórias do usuário. As técnicas de modelagem de teste baseadas em especificação são geralmente úteis para derivar condições de teste de histórias de usuários e critérios de aceite. No entanto, no Agile, na maioria das vezes, essas técnicas de modelagem de teste são usadas mais implicitamente do que explicitamente, pelo que os testadores com base em sua experiência dominam as técnicas e podem usá-las com flexibilidade no contexto. As condições de teste serão documentadas em um formato leve, contrário à maneira mais tradicional de trabalhar, onde estão documentadas como parte de um documento de especificação de projeto de teste. Às vezes, especialmente em projetos onde o teste exploratório é amplamente utilizado, as condições de teste são apenas identificadas por meio de brainstorming e são documentadas como idéias de teste (para fazer parte das cartas de teste). A definição das condições de teste também é a base para estabelecer a rastreabilidade horizontal (ver adiante) e gerenciar a cobertura do conjunto de testes de regressão (automatizado) pelo qual os testes automatizados são desenvolvidos para cobrir condições específicas de teste.

Com o princípio do teste primeiro sendo aplicado com o Agile, os testes que abrangem o conjunto de condições de teste serão identificados (e possivelmente automatizados) antes ou pelo menos paralelamente ao desenvolvimento

do código. Para testes de unidade automatizados, pode ser considerada uma abordagem como Desenvolvimento Orientado a Testes. Para níveis de teste mais altos, o Desenvolvimento Orientado a Comportamentos (BDD - *Behavior-Driven Development*) e o Desenvolvimento Orientado a Testes de Aceite (ATDD - *Acceptance Test-Driven Development*) são abordagens populares de desenvolvimento Agile que também suportam testes. O BDD e o ATTD também estão altamente relacionados à automação de testes.

Para a maioria dos testes manuais, os testes serão identificados ou refinados à medida que a equipe avança na execução do teste. Os testes geralmente não são documentados com tantos detalhes quanto nos projetos tradicionais, mas em um formato de idéias de teste usando testes exploratórios. Para áreas complexas e críticas, uma abordagem mais tradicional para a modelagem de teste e o desenvolvimento de casos de teste usando técnicas formais de modelagem de teste pode ser a melhor maneira de cobrir os riscos. No entanto, também com essa abordagem, a quantidade de documentação será limitada em comparação com os testes em um ambiente tradicional de ciclo de vida sequencial. A priorização de testes segue a priorização da história do usuário que eles estão cobrindo. A priorização das histórias de usuários é baseada no valor comercial, a maior prioridade está relacionada ao maior valor comercial. É importante que os testes sejam estabelecidos para cobrir riscos funcionais e não funcionais, mas especialmente no Agile, também especificamente para cobrir o risco de regressão.

Dados de teste específicos necessários para suportar as condições de teste e a execução de testes são identificados. No entanto, no Agile, em contraste com os ambientes tradicionais, os dados de teste necessários geralmente não são especificados primeiro como parte de um documento de especificação de teste. Antes que as ferramentas ou funcionalidades necessárias estejam disponíveis, os dados do teste são criados imediatamente para permitir um início quase imediato da execução dos testes manuais. No entanto, para testes automatizados, os dados geralmente precisam ser especificados antecipadamente.

A rastreabilidade entre os requisitos, condições e testes deve ser estabelecida e mantida. As equipes precisam deixar claro que cobriram as várias histórias de usuários e critérios de aceite como parte de seus testes. Embora uma ferramenta formal de gerenciamento de requisitos possa não ser necessária, a equipe precisa de suporte para ter seus requisitos organizados e gerenciados de maneira a suportar a atribuição de identificadores para cada requisito, para que esses identificadores possam ser usados para garantir que os testes sejam concluídos de acordo com o acordado. critério. Em muitas organizações ágeis, as histórias de usuários são a base para o desenvolvimento de um conjunto de critérios de aceite relacionados e testes subsequentes. Uma vez estabelecidos esses testes, os próprios testes geralmente se tornam os requisitos detalhados acordados. O uso dessa abordagem pode ser suficiente para atingir a intenção da prática específica do TMMi no gerenciamento de requisitos relacionados à rastreabilidade horizontal.

2.4.2 SG2 Executar a implementação do teste

A implementação do teste consiste em colocar tudo o que é necessário para iniciar a execução dos testes. Normalmente, o desenvolvimento da documentação de teste, por exemplo, procedimentos de teste, para suportar a execução do teste é minimizado. Os scripts de teste automatizados (de regressão) são desenvolvidos e priorizados. Também a preparação do teste de regressão começa o mais rápido possível, paralelamente a outras atividades de teste.

Em um ambiente Agile, procedimentos detalhados de teste não são uma prática comum. Trabalhando em uma equipe experiente, os testes provavelmente serão documentados em um nível de abstração muito mais alto. Isso será suficiente para aqueles que executam os testes, pois é esperado que eles tenham o nível necessário de domínio e conhecimento técnico para realizar os testes. Aqueles que executam os testes trabalham dentro da equipe, para que compreendam melhor o que e como ele é codificado e como a funcionalidade é adequada ao objetivo. Como o nível de mudança dentro e entre as iterações é geralmente alto, o desenvolvimento de procedimentos de teste detalhados também criaria muito trabalho de manutenção. Normalmente, muitos testes são realizados usando testes exploratórios. Nos testes exploratórios, nenhum procedimento de teste detalhado é desenvolvido, mas cartas de teste de uma página de alto nível descrevendo idéias e orientações para o testador. No entanto, dados de teste específicos necessários para a realização de testes precisam, obviamente, ser criados antecipadamente.

Muitas equipes ágeis usam testes automatizados. As abordagens típicas utilizadas são o Desenvolvimento Orientado a Testes (TDD - *Test-Driven Development*), Desenvolvimento Orientado a Comportamentos (BDD - *Behavior-Driven Development*) e Desenvolvimento Orientado a Testes de Aceite (ATDD - *Acceptance Test-Driven Development*). Usando uma ou mais dessas abordagens, *scripts* de teste automatizados serão criados como parte da atividade de implementação de teste.

A integração contínua é uma prática essencial para projetos Agile. O teste é uma atividade paralela e totalmente integrada a ser executada pela equipe. Não é uma atividade ou fase separada independente. A prática específica 2.3 Especificar o procedimento de teste de ingestão, portanto, torna-se irrelevante dentro de uma equipe Agile. Não

há necessidade e, na verdade, não faz sentido ter um teste formal específico de ingestão sendo especificado e executado. No entanto, caso algum teste, por exemplo, teste de integração do sistema ou teste de integração de hardware / software, esteja sendo executado fora do escopo da equipe Agile, eles podem especificar um teste de admissão para avaliar a qualidade da saída da equipe Agile e determinar se o produto que foi entregue está pronto para o teste.

Em ambientes Agile, nenhum cronograma de execução de teste específico será criado e, portanto, a prática específica 2.4 Desenvolver o cronograma de execução de teste provavelmente não é relevante. Há um alto nível de flexibilidade em relação à ordem em que o teste será executado ao longo de uma iteração, embora isso seja guiado pela prioridade das várias histórias de usuários. Os vários testes a serem executados serão identificados durante o planejamento da iteração e serão gerenciados como tarefas de teste através do painel de tarefas Agile.

2.4.3 SG3 Executar os Teste

Os itens de software que estão sendo desenvolvidos na iteração precisam, obviamente, serem testados. Após o manifesto Agile, assim como os outros objetivos nessa área de processo, isso normalmente é feito com muito menos intensidade de documentação do que em um projeto sequencial tradicional. Frequentemente, nenhum procedimento de teste detalhado e logs de teste são produzidos. Os testes, especialmente os testes de regressão, devem, tanto quanto possível e possível, ser automatizados.

Em um projeto Agile, o software é entregue, no mínimo, diariamente e o teste é parte integrante do processo de desenvolvimento Agile, no qual o trabalho é realizado em cooperação mútua para fornecer um produto de qualidade. Não há equipe de desenvolvimento e teste dedicada e separada. Portanto, um teste de ingestão específico, conforme definido na prática específica 3.1. Realizar um teste de admissão provavelmente não é relevante em um ambiente Agile puro. No entanto, quando alguns testes são realizados fora da equipe do Agile (como geralmente é o caso), um teste de admissão geralmente é realizado pela equipe de teste responsável por testes fora da equipe do Agile. Lembre-se de que há atividades que devem ser concluídas antes do início do teste de aceite de uma história de usuário específica dentro da equipe do Agile. Essas atividades geralmente fazem parte da Definição de Concluído, por exemplo, a quantidade acordada de testes de unidade foi concluída e aprovada.

A execução do teste é realizada de acordo com as prioridades definidas durante o planejamento da iteração. Alguns testes podem ser executados usando um procedimento de teste documentado, mas geralmente muitos testes são executados usando testes exploratórios e baseados em sessões como sua estrutura. Nos testes exploratórios, a modelagem e a execução do teste ocorrem ao mesmo tempo, guiados por uma carta de teste preparada. Um termo de teste fornece objetivos e condições de teste para cobrir durante uma sessão de teste com intervalo de tempo. Durante os testes exploratórios, os resultados dos testes mais recentes orientam os próximos testes.

Com o desenvolvimento iterativo, há uma necessidade crescente de organizar e estruturar testes de regressão. Às vezes, isso é feito manualmente, mas de preferência é feito usando testes de regressão automatizados e ferramentas de suporte. O teste de regressão, especialmente o teste de unidade e integração, geralmente faz parte de um processo de integração contínuo. A integração contínua é uma prática importante no Desenvolvimento Ágil de Software. É basicamente um processo automatizado de criação e teste que ocorre pelo menos diariamente e detecta cedo e rapidamente. A integração contínua permite a execução regular de testes de regressão automatizados e o envio de *feedback* rápido para a equipe sobre a qualidade do código e a cobertura do código alcançada.

Os incidentes encontrados durante o teste podem ser registrados e relatados. Normalmente, há uma discussão nos projetos Agile se todos os incidentes encontrados devem realmente ser registrados. Particularmente no caso em que a equipe está localizada, os testadores precisam conversar com os desenvolvedores e os incidentes / defeitos que podem ser corrigidos imediatamente e incluídos na próxima compilação podem não precisar ser registrados, apenas precisam ser corrigidos. No Agile, geralmente nem todos os incidentes encontrados serão registrados e gerenciados de acordo. Como afirmado, muitos são criados informalmente por um membro da equipe (que pode ser um testador) e são corrigidos imediatamente pelo desenvolvedor que injetou o defeito. Alguns dados são perdidos, mas a sobrecarga ao mesmo tempo é reduzida. Algumas equipes registram apenas incidentes que escapam às iterações, algumas registram se não puderem ser corrigidas hoje, outras registram apenas incidentes de alta prioridade. Se nem todos os incidentes encontrados forem registrados, os critérios deverão estar disponíveis para determinar quais incidentes devem ser registrados e quais não. Lembre-se de que a intenção é corrigir o defeito corretamente, não registrar um incidente. Um exemplo desse critério: “Se um defeito puder ser resolvido antes da próxima espera diária e construção diária, ele não precisará ser documentado. Qualquer defeito levantado que não possa ser fechado dentro de um dia ou que afecte uma parte interessada fora da equipe deve ser documentado e capturado.”

Às vezes, os incidentes são registrados no painel de tarefas do Agile, como um adesivo em uma tarefa ou como uma tarefa separada, visualizando que ele está impedindo que uma história e suas tarefas sejam concluídas. É aceitável usar notas post-it em um painel de tarefas para gerenciar seus defeitos e esse é geralmente o método escolhido para as equipes localizadas. A equipe do Agile trabalhará posteriormente nos incidentes e os gerenciará até o encerramento (consulte a meta específica 4 desta área de processo a seguir). Alguns optam por registrar e documentar incidentes encontrados em uma ferramenta, por exemplo, gerenciamento de defeitos ou ferramenta de rastreamento de defeitos. Essa ferramenta deve ser usada da maneira mais enxuta possível e não forçar a apresentação de quaisquer elementos dos relatórios de incidentes de teste que tenham pouco ou nenhum valor agregado.

Também é considerado uma boa prática nas equipes do Agile registrar dados durante a execução do teste para determinar se os itens testados atendem aos critérios de aceite definidos e podem realmente ser rotulados como "concluídos". Isso também é verdade ao aplicar técnicas baseadas em experiências, como testes exploratórios. Exemplos de informações que podem ser úteis para documentar são a cobertura do teste (quanto foi coberto e quanto resta a ser testado), observações durante o teste, por exemplo, se o sistema e a história do usuário sob teste parecem estáveis, lista de riscos (quais riscos foram cobertos e quais permanecem entre os mais importantes), defeitos encontrados e outros problemas e possíveis questões em aberto.

As informações registradas devem ser capturadas ou resumidas em alguma forma de ferramenta de gerenciamento de status (p.ex.: ferramentas de gerenciamento de teste, ferramentas de gerenciamento de tarefas, quadro de tarefas), de uma maneira que facilite à equipe e às partes interessadas o entendimento do status atual para todos os testes que foram realizados. Obviamente, não é apenas o status das tarefas de teste isoladas em que a equipe precisa de um status, mas sim o status geral da história do usuário. Nada está completo do ponto de vista da história, a menos que todas as atividades relacionadas estejam concluídas, não apenas testando.

2.4.4 SG4 Gerenciar os incidentes de teste até o encerramento

Como mencionado acima, em ambientes ágeis, muitas vezes nem todos os incidentes encontrados são registrados. Esse objetivo específico se aplica apenas aos incidentes registrados e, portanto, precisam ser gerenciados para serem encerrados. Em princípio, gerenciar incidentes no Agile é simples. No caso de o incidente ser registrado no painel de tarefas do Agile, como um adesivo em uma tarefa ou como uma tarefa separada, ele é visualizado como um defeito que impede uma história e suas tarefas de serem concluídas. O modo de trabalho Agile padrão será aplicado como qualquer outro impedimento ou tarefa que esteja bloqueando o progresso. Será discutido durante as reuniões de *stand-up* e designado para ser resolvido pela equipe, momento em que desaparecerá do quadro de tarefas.

Caso seja tomada a decisão de adiar o incidente encontrado para outra iteração, eles se tornarão apenas outra opção (ou alteração) desejada para o produto. Adicione-os ao *backlog* e priorize de acordo. Quando a prioridade é alta o suficiente, eles serão escolhidos pela equipe na próxima iteração. Observe que a prioridade geralmente é definida pela empresa, mas às vezes é preciso dar prioridade a incidentes relacionados à chamada dívida técnica. Algumas equipes têm iterações dedicadas para limpar incidentes ou defeitos relacionados à dívida técnica. No entanto, essa não é uma prática padrão recomendada pelo Agile, pois a dívida deve ser paga regularmente.

Como um exemplo de como os incidentes podem ser tratados em um ambiente Agile, primeiro imagine que a equipe esteja no meio de uma iteração. Enquanto trabalha em uma das histórias de usuário, a equipe determina que não é possível "concluir", porque um dos critérios de aceite não é atendido, por exemplo, devido a um defeito introduzido durante o desenvolvimento. As equipes ágeis tendem a corrigir esses tipos de defeitos o mais rápido possível na iteração atual, caso contrário, eles continuariam com o trabalho desfeito (equivalente ao inventário enxuto). Esses tipos de incidentes serão visualizados - e gerenciados através do quadro de tarefas.

Em segundo lugar, suponha que todas as histórias da iteração sejam concluídas de acordo com a Definição de Concluído até o final da iteração. Porém, se durante a reunião de revisão, ao demonstrar uma história, houver algo que não se comporte da maneira que o cliente gostaria. Considere que o comportamento inesperado não foi realmente discutido junto com os negócios antes do início da iteração. Nesse caso, a funcionalidade do software produzida está funcionando e o que provavelmente precisa ser feito é converter o incidente em uma história adicional e adicioná-lo à lista de pendências do produto.

É uma boa prática monitorar o número de defeitos encontrados e resolvidos durante uma iteração, bem como o número de defeitos não resolvidos, possivelmente se tornando parte do backlog da próxima iteração durante a reunião diária (lançamento) e também em reuniões retrospectivas.

Em um ambiente tradicional de ciclo de vida sequencial, geralmente há um CCB (Painel de Controle de Configuração) que atende, revisa e decide a disposição do incidente. Esse papel no Agile é assumido pela equipe com poderes. É claro que o representante comercial, por exemplo, proprietário do produto, tem um papel importante

a desempenhar quando se trata de decidir sobre a disposição de um incidente. Lidar com incidentes é uma sobrecarga, está tirando o foco e o esforço de outras coisas. A sobrecarga aumenta à medida que tentamos formalizar um processo para listá-los, qualificá-los, solicitá-los e, por fim, planejá-los. Um processo complexo de gerenciamento de incidentes como esse é um simples desperdício. Melhor descrever alguns princípios básicos sobre como os incidentes são tratados e gerenciados pelas equipes Agile e agir de acordo.

2.5 Área de Processo 2.5 Ambiente de Teste

O objetivo do ambiente de teste é estabelecer e manter um ambiente adequado, incluindo dados de teste, nos quais é possível executar os testes de maneira gerenciável e repetível.

Com a área de processo Ambiente de teste, um ambiente de teste adequado é estabelecido e mantido, incluindo dados genéricos de teste, nos quais é possível executar os testes de maneira gerenciável e repetível. Obviamente, também em um projeto de desenvolvimento de software que usa o ciclo de vida do Agile, um ambiente de teste adequado é indispensável. Devido aos curtos tempos de ciclo de uma iteração, o ambiente de teste precisa ser altamente estável e disponível. Os problemas no ambiente de teste sempre terão um impacto imediato no progresso e nos resultados da iteração. Gerenciar adequadamente a configuração e as alterações no ambiente de teste e nos dados de teste também é da maior importância.

Os ambientes de teste são variados, dependendo, entre outros, do sistema em teste, mas geralmente a especificação, implementação e gerenciamento do ambiente de teste são feitos paralelamente ao projeto de Desenvolvimento Ágil de Software. A implementação de um ambiente de teste geralmente leva muito tempo e pode ser altamente complexa, tornando quase impossível executar essa tarefa dentro do tempo de ciclo limitado de uma iteração [Van der Aalst e Davis]. Também envolve diferentes partes interessadas e engenheiros que normalmente formam uma equipe Agile. As metodologias Agile foram desenvolvidas para dar suporte ao desenvolvimento de software, não para ambiente de teste ou desenvolvimento e gerenciamento de infraestrutura. É apenas menos adequado e a maneira como o ambiente de teste ou o desenvolvimento e gerenciamento da infraestrutura é realizada permanece praticamente a mesma de um ambiente tradicional, embora possa haver algumas mudanças no processo quando o Agile está sendo praticado.

2.5.1 SG1 Desenvolver requisitos do ambiente de teste

A especificação dos requisitos do ambiente de teste é realizada no início do projeto. A especificação de requisitos é revisada para garantir sua correção, adequação, viabilidade e representação precisa de um ambiente operacional de 'vida real'. A especificação antecipada de requisitos tem a vantagem de fornecer mais tempo para adquirir ou desenvolver o ambiente de teste e os componentes necessários, como simuladores, controladores ou drivers.

Em alguns projetos de desenvolvimento Agile, a chamada iteração inicial (iteração 0) é aplicada onde a elicitação e especificação dos requisitos do ambiente de teste são realizadas.

2.5.2 SG2 Executar a implementação do ambiente de teste

Às vezes (parte) da implementação ocorre na iteração inicial, além de outras atividades, por exemplo, treinamento, avaliação de alto nível de risco do produto e estabelecimento da definição de concluído. Dessa maneira, busca-se alcançar que mesmo as primeiras iterações de desenvolvimento de software já tenham um ambiente de teste que funciona parcialmente. Ações ou problemas técnicos ambientais podem fazer parte do backlog do produto.

Definir o ambiente de teste (incluindo dados genéricos de teste) poderia iniciar um ciclo de vida sequencial por meio de um plano completo, mas geralmente é melhor começar com a implementação o mais rápido possível e ter uma versão inicial do ambiente de teste disponível. no início da primeira iteração.

Na verdade, é difícil ser genérico em ambientes de teste. Em alguns domínios, controladores e drivers estão sendo rapidamente substituídos pela virtualização de serviço, geralmente permitindo uma implementação muito mais rápida do ambiente de teste. Em alguns projetos Agile, o provisionamento de ambientes de teste é altamente automatizado. Os ambientes de teste podem ser comissionados em minutos ou horas, e não em dias. Atualmente, há muito mais uso da virtualização de máquinas, servidores e serviços. Além disso, a nuvem pode ajudar a fornecer os ambientes necessários, acelerando novamente o acesso.

2.5.3 SG3 Gerenciar e controlar ambientes de teste

O gerenciamento e o controle do ambiente de teste ocorrerão durante um projeto Agile. Às vezes, a equipe do Agile precisa configurar e manter o ambiente de teste, mas na maioria das vezes isso é feito por uma unidade separada fora da equipe do Agile. Quando as atividades são realizadas pela equipe Agile, é claro que isso implica que a equipe possui conhecimento técnico suficiente para poder executar essas tarefas. Isso também significa que as tarefas se tornam parte do processo Agile, por exemplo, precisam ser abordadas em uma sessão de planejamento, colocadas no quadro de tarefas e possivelmente discutidas durante as reuniões de lançamento diárias em caso de problemas.

Como principal conclusão para o ambiente de teste da área de processo, os objetivos e práticas específicos ainda são aplicáveis e não mudam em essência. O que muda é o tempo no ciclo de vida.

3 TMMi Nível 3 Definido

3.1 Área de Processo 3.1 Organização de Teste

O objetivo da área de processo da Organização de Teste é identificar e organizar um grupo de pessoas altamente qualificadas, responsáveis pelos testes. Além do teste, o grupo de teste também gerencia melhorias no processo de teste da organização e nos ativos do processo de teste com base em um entendimento completo dos pontos fortes e fracos do processo atual de teste da organização e dos ativos do processo de teste.

3.1.1 SG 1 Estabelecer uma organização de teste

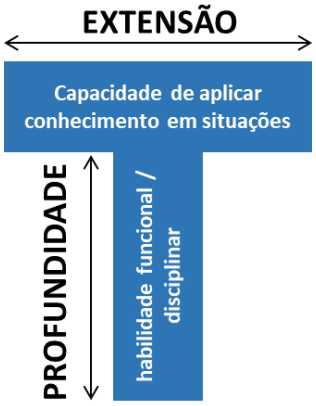
A organização de teste é uma área de processo muitas vezes incompreendida. Muitos veem dessa forma porque o TMMi requer um grupo de testes independente ou mesmo um departamento que faça testes independentes. Por mais que isso seja possível, também existem outros modelos organizacionais que atendem aos requisitos do TMMi. Além da organização de teste independente que prepara e executa testes, o centro de competência de teste também é uma opção. As tarefas típicas e responsabilidades de um centro de competência de teste, estão estabelecendo, gerenciando e aprimorando os processos de teste, e fornecendo os recursos para projetos com experiência, conhecimento e habilidades em teste. Um centro de competência de teste geralmente possui propriedade sobre metodologias e processos de teste. Observe que um testador no dia-a-dia geralmente faz parte de uma equipe Agile, mas, além disso, é comum que um testador pertença a uma organização de teste, como o centro de competência de teste ou a corporação de teste.

Uma organização de teste exige liderança em áreas relacionadas aos problemas de teste e qualidade. Os membros da equipe desse grupo são chamados de especialistas em teste. A opção do centro de competência de teste se ajusta bem ao Agile e garante que os testes sejam preservados como disciplina e tratados com seriedade. É claro que existem líderes de pensamento ágeis que não são a favor de manter disciplinas especiais. Embora essa possa ser uma opção preferida para alguns, fazer o TMMi exige tratar e manter os testes como uma disciplina. Escusado será dizer que isso só deve ser feito se fizer sentido do ponto de vista comercial e, portanto, quando isso não faz sentido, o TMMi provavelmente não é o caminho de melhoria preferido.

No Agile, a organização de teste pode assumir o formato de um grupo de teste. Um grupo de teste geralmente é um grupo informal de autogerenciadores, que são membros de diferentes equipes Agile. As atividades importantes de um grupo de teste podem ser o compartilhamento de conhecimento, o aprendizado em grupo, a observação das tendências etc. A associação a um grupo é voluntária. Na prática, um grupo de teste geralmente se reúne duas vezes por semana. O grupo também pode receber tarefas mais formais, como estabelecer trajetórias de carreira de teste, determinar e organizar treinamentos, planejar e implementar melhorias no processo de teste. É importante garantir que a associação de teste permaneça amplamente informal e autogerenciada. Uma restrição importante para que um grupo de teste seja bem-sucedida é que seus membros recebam tempo para gastar nas várias atividades da guilda de teste a serem executadas.

Testadores independentes costumam ser mais eficazes em encontrar defeitos. Equilibrar essa independência com o Agile é um desafio. Algumas equipes do Agile mantêm equipes de teste totalmente independentes e separadas e atribuem testadores sob demanda durante diariamente aos finais de cada sprint. Isso pode preservar a independência e esses testadores podem fornecer uma avaliação objetiva e imparcial do software. No entanto, pressões de tempo, falta de entendimento dos novos recursos do produto e problemas de relacionamento com os stakeholders e desenvolvedores de negócios geralmente levam a problemas com essa abordagem. Outra opção é ter uma organização de teste independente e segregada, na qual os testadores são designados para as equipes Agile a longo prazo, no início do projeto, permitindo que eles mantenham sua independência enquanto obtêm um bom entendimento do produto e fortes relacionamentos com outros membros da equipe. Além disso, a organização independente de teste pode conter testadores especializados fora das equipes Agile para trabalhar em atividades de longo prazo ou independentes de iteração, como desenvolvimento de ferramentas de teste automatizadas, realização de testes não funcionais, criação e suporte de ambientes de teste e dados e execução de níveis de teste que podem não se encaixar bem em um sprint (p.ex.: teste de integração de sistemas). Observe que também há possibilidades de praticar testes independentes em um a equipe do Agile. Desde que alguém que não seja o criador de ativos ou código também teste o produto, isso ainda é independência. Portanto, os desenvolvedores podem testar o código um do outro por meio de análises de código ou teste de unidade, ou analistas de negócios que contribuem para o teste da perspectiva de um usuário usando, por exemplo, diferentes personalidade. Por último, mas não menos importante, a independência pode ser organizada através de uma estrutura organizacional e de responsabilidade, mas provavelmente está acima de tudo tendo a mentalidade crítica correta.

3.1.2 SG 2 Estabelecer as funções de teste para especialistas em teste



O teste é considerado uma valiosa profissão e disciplina. O conhecimento e as habilidades de teste são necessários em uma equipe do Agile. O especialista em testes os fornecerá e também treinará outros membros da equipe durante a execução das atividades de teste. As funções de teste e os planos de carreira de teste são definidos e suportados por um programa de treinamento de teste. O conhecimento e as habilidades necessárias para um típico testador Agile são diferentes dos de uma organização tradicional. Obviamente, o testador ainda precisará de conhecimentos e habilidades "tradicionais", mas também conhecimentos e habilidades para automação de testes e para poder executar tarefas fora dos testes, por exemplo, scripts ou engenharia de requisitos. Como um testador Agile está trabalhando em equipe, as habilidades sociais são igualmente importantes para aumentar as habilidades técnicas. O testador ágil é o chamado testador de forma em T, que deve ser refletido na descrição das funções de teste. A organização de teste é composta por pessoas que têm a habilidade e motivação para serem bons testadores em um contexto Agile. Eles são atribuídos a uma função específica de teste. Especialmente em um contexto ágil, é importante, uma vez que toda a equipe executará atividades de teste, para elaborar outras descrições de funções específicas que não sejam de teste, com expectativa em relação às atividades de teste, funções e conhecimentos e habilidades de teste necessários.

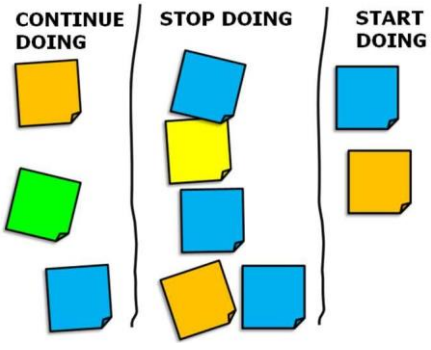
3.1.3 SG 3 Estabelecer os planos das carreira de teste

As carreiras de teste são estabelecidas para permitir que os profissionais aprimorem seus conhecimentos, habilidades, status e recompensas e, assim, permitam que eles avancem em suas carreiras. Com base nas trajetórias de carreira de teste definidas, os planos pessoais de ascensão na carreira de teste são desenvolvidos e mantidos para todos os membros da organização de teste. Esse objetivo específico e as práticas específicas que o acompanham também são totalmente aplicáveis em um contexto Agile. Com a ênfase do Agile no aspecto de pessoas, esse é claramente um objetivo específico importante. A equipe Agile precisa de testadores que entendam seu trabalho desafiador, possam treinar outros membros da equipe e estar motivados. Muitos deles são abordados com planos de carreira de teste.

No entanto, a real definição das carreiras de teste normalmente será um pouco diferente em uma organização Agile comparando-se as de uma organização que segue um ciclo de vida sequencial. Enquanto na organização tradicional os planos de carreira de teste geralmente têm foco no crescimento vertical (do testador ao gerente de teste), os planos de carreira de teste em uma organização Agile tendem a se concentrar no crescimento horizontal (do testador júnior ao testador sênior e talvez treinador de teste). Normalmente, há menos necessidade de várias funções e funções hierárquicas, como gerenciamento de testes e uma maior necessidade de crescer ao longo das funções de um dono do produto ou *scrum master*. Observe que muitas das tarefas tradicionais de gerenciamento de teste ainda são relevantes, mas são assumidas por outras pessoas (p.ex.: *scrum master*, equipe Agile).

3.1.4 SG 4 Determinar, planejar e implementar as melhorias no processo de teste

O processo de melhoria descrito por esse objetivo específico do TMMi e suas práticas específicas são amplamente abordados pela reunião retrospectiva realizada por uma equipe do Agile. A retrospectiva da iteração é uma oportunidade para a equipe do Agile se inspecionar e definir ações para melhorias a serem implementadas durante a próxima iteração. Uma retrospectiva geralmente ocorre após a revisão da iteração e antes do próximo planejamento da iteração.



Durante a retrospectiva, a equipe discute o que correu bem na iteração, o que poderia ser melhorado e o que estará comprometido em melhorar na próxima iteração. Durante cada retrospectiva, a equipe do Agile identifica e planeja maneiras de aumentar a qualidade do produto, por exemplo, melhorando os processos de trabalho ou adaptando a Definição de Concluído, se apropriado. Retrospectivas podem resultar em decisões de melhoria relacionadas a testes focadas na eficácia, produtividade e qualidade dos testes. Eles também podem abordar a testabilidade dos aplicativos, histórias de usuários, recursos ou interfaces do sistema. A análise da causa raiz de defeitos pode impulsionar melhorias de teste e desenvolvimento. Os testadores desempenham um papel importante nas retrospectivas. Eles fazem parte da equipe e trazem uma perspectiva única.

Todos os membros da equipe, testadores e não testadores, podem fornecer informações sobre as atividades de teste e não-teste.

No final da retrospectiva, a equipe do Agile deve ter identificado os aprimoramentos que serão implementados na próxima iteração. A implementação dessas melhorias na próxima iteração é a adaptação à inspeção da própria equipe do Agile. A equipe vota nas melhorias mais importantes para a próxima iteração, para que as melhorias (teste) possam permanecer na lista de pendências por mais de uma iteração, mas elas serão implementadas. É importante que a equipe faça apenas melhorias que possam caber na próxima iteração, portanto, provavelmente, no máximo duas ou três por iteração. Os testadores devem se apropriar das melhorias de teste para garantir que elas sejam implementadas dentro da equipe. A retrospectiva é um mecanismo importante que permite que uma equipe evolua e melhore continuamente ao longo da vida de um projeto.

Como o escopo geralmente é limitado ao ciclo ou iteração anterior, são feitas pequenas, mas frequentes melhorias que se concentram principalmente na solução de problemas específicos do projeto. O foco dessas melhorias geralmente não é o aprendizado entre projetos e a institucionalização de melhorias. Observando como a melhoria do processo (teste) é organizada e gerenciada, é provável que haja menos foco em um grupo de processos (teste) no nível organizacional e mais ênfase no autogerenciamento das equipes no projeto. Embora isso não seja ruim, é importante também criar um mecanismo pelo qual as melhorias locais de teste bem-sucedidas sejam compartilhadas com o restante da organização e as áreas de melhoria que estão além do poder da equipe Agile também podem ser abordadas. Uma solução possível seria fazer com que um representante da organização de teste participasse de reuniões retrospectivas locais para garantir que as melhorias fossem compartilhadas e institucionalizadas em toda a organização, quando apropriado. O melhorador do processo de teste também pode remover problemas mais amplos ou fundamentais que afetam os testes em toda a organização.

3.1.5 SG 5 Implantar o processo de teste organizacional e incorporar as lições aprendidas

Também para um projeto Agile, é importante ter acesso e reutilizar o processo de teste padrão organizacional e os ativos do processo de teste, conforme necessário e com valor agregado. A organização de teste garantirá que eles sejam implantados em todos os projetos. (Consulte a área de processo Testar ciclo de vida e integração para obter mais informações sobre o processo de teste padrão organizacional e os ativos do processo de teste e como isso se relaciona aos projetos Agile.)

A implementação do processo de teste padrão da organização e o uso de ativos do processo de teste em projetos Agile podem ser monitorados pelas próprias equipes auto-organizadas, por exemplo, abordando isso em retrospectivas. Com um representante da organização de teste, por exemplo, um melhorador de processo de teste, estando presente nessas reuniões, essa pessoa pode acompanhar o status da implementação e resolver problemas entre projetos. Essa pessoa também pode tirar lições valiosas aprendidas e testar melhorias da equipe Agile de volta à organização de teste. As lições aprendidas e a melhoria do teste podem ser subsequentemente enviadas como propostas de melhoria do processo de teste e gerenciadas de acordo. As lições aprendidas dos testes com a equipe Agile são, portanto, se apropriadas, incorporadas ao processo do padrão organizacional (Agile) (teste) e aos ativos do processo de teste.

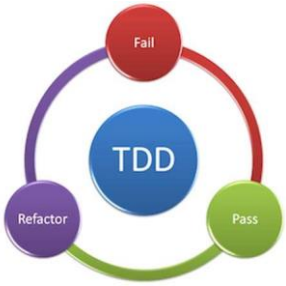
3.2 Área de Processo 3.2 Programa de Treinamento em Teste

O objetivo da área de processo do Programa de Treinamento em Testes é desenvolver um programa de treinamento que facilite o desenvolvimento do conhecimento e das habilidades das pessoas, para que tarefas e funções de teste possam ser executadas de maneira eficaz e eficiente.

O principal objetivo do Programa de Treinamento em Testes é fornecer o treinamento necessário aos testadores e outros membros da equipe. Como resultado da declaração do manifesto Agile “indivíduos e interações sobre processos e ferramentas”, os processos em um ambiente Agile geralmente são documentados com pouca carga e não abordam todos os cenários possíveis. O treinamento de pessoas, no entanto, torna-se um fator crítico de sucesso que precisa de foco e é quase como uma contramedida para os processos leves. Um programa de treinamento de qualidade garante que os envolvidos nos testes continuem melhorando seus conhecimentos e habilidades em testes e adquiram o conhecimento de domínio necessário e outros conhecimentos e habilidades relacionados aos testes. Como a qualidade e o teste agora são responsabilidades da equipe, espera-se que o treinamento relacionado ao teste não seja fornecido apenas para profissionais de teste, mas para toda a equipe do Agile.

3.2.1 SG1 Estabelecer um recurso de treinamento em testes organizacionais

O processo começa com a identificação das necessidades de treinamento em teste estratégico da organização. Além das necessidades tradicionais de treinamento para testadores, por exemplo, em técnicas de modelagem de teste e medidas de cobertura, agora também é necessário adquirir conhecimentos e habilidades sobre as especificidades do teste Agile. Os exemplos incluem o manifesto Agile e seus princípios, estruturas como Scrum, Kanban e XP, conceitos de teste Agile (p.ex.: pirâmide de teste e quadrantes de teste), *Test Driven Development* (TDD), *Behavior Driven Development* (BDD), *Behavior Driven Development* (BDD), *Acceptance TestDriven Development* (ATDD), desenvolvimento da história do usuário, incluindo suporte à definição de critérios de aceite e revisão dos mesmos quanto à testabilidade, automação de teste e teste exploratório. Além disso, algumas áreas típicas de conhecimento e habilidades de gerenciamento de testes, como planejamento, estimativa (p.ex.: *Poker Planning*), monitoramento e análise de risco (p.ex.: *Poker Risk*) agora se tornarão áreas que precisam fazer parte do conjunto de conhecimentos e habilidades para a maioria dos testadores que trabalham em uma equipe ágil e não apenas para o gerente de teste. A identificação das necessidades estratégicas de treinamento em testes não se limita apenas aos testadores, é de extrema importância especificar também as necessidades de treinamento relacionadas a testes para outros membros das equipes Agile.



Assim como em qualquer outra organização, as necessidades de treinamento em teste serão traduzidas em um plano de treinamento (leve), possivelmente também abordando algumas necessidades específicas de treinamento em teste de projeto. O plano de treinamento precisa ser revisto e os compromissos precisam ser estabelecidos. Observe que, especialmente no Agile, algumas habilidades são transmitidas de maneira eficaz e eficiente por meio de veículos informais (p.ex.: treinamento no trabalho, sessões de treinamento no almoço, treinamento e orientação), enquanto outras habilidades requerem treinamento formal. A abordagem apropriada para satisfazer necessidades específicas de treinamento em testes deve ser determinada por área de conhecimento e habilidades, por exemplo, com o treinamento exploratório em testes no trabalho normalmente sendo um aspecto importante.

3.2.2 SG2 Prover o treinamento necessário em teste

Em termos gerais, as práticas específicas para o SG2 são as mesmas em uma organização tradicional e ágil. Com base no plano de treinamento organizacional, são identificados os membros da equipe que precisam receber o treinamento necessário para poder desempenhar sua função de teste com eficiência. O treinamento é subsequentemente agendado, incluindo os recursos necessários, e realizado. Parece uma afirmação óbvia, mas algumas organizações ágeis lutam para encontrar tempo para o treinamento. As equipes ágeis estão sempre totalmente engajadas e tentando cumprir o prazo final de iteração. É claro que o tempo para o treinamento precisa ser tratado no planejamento da iteração, pelo qual o participante estará menos disponível para a próxima iteração.

Conforme mencionado acima, os veículos de treinamento informal são frequentemente usados em uma organização Agile. Por exemplo, além do treinamento formal, a orientação e o treinamento no trabalho geralmente são continuamente incentivados e empregados. De fato, o aconselhamento é responsabilidade de todos e esperado em todos os níveis de uma organização. Trabalhar em pares é outro veículo comumente usado pelas equipes Agile para aperfeiçoamento de habilidades e transferência de conhecimento. Ao usar o *Scrum*, é responsabilidade do scrum master treinar a equipe nas práticas do *Scrum*.

Os registros do treinamento realizado em teste, formal ou informal, serão criados e a eficácia do treinamento será avaliada. Um treinamento assistido também pode ser avaliado como parte de uma reunião de retrospectiva, na qual ocorre uma discussão sobre se os conhecimentos e habilidades dos participantes agora são mais adequados para a execução das tarefas de teste.

3.3 Área de Processo 3.3 Ciclo de Vida e Integração de Testes

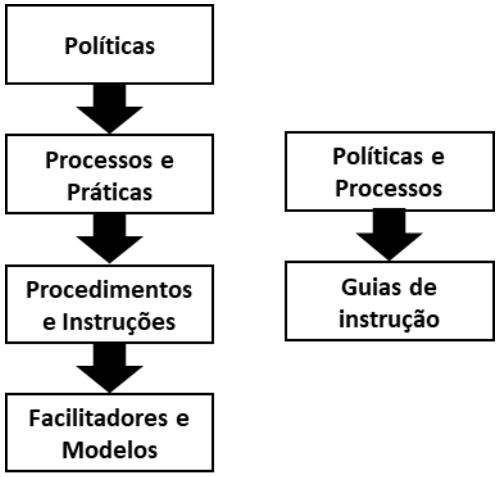
O objetivo do Ciclo de Vida e Integração de Testes é estabelecer e manter um conjunto utilizável de ativos do processo de teste organizacional (p.ex.: um ciclo de vida de teste padrão) e padrões do ambiente de trabalho e integrar e sincronizar o ciclo de vida de teste com o ciclo de vida de desenvolvimento. O ciclo de vida integrado garante o envolvimento precoce dos testes em um projeto. O objetivo do Ciclo de Vida do Teste e da Integração também é definir uma abordagem de teste coerente em vários níveis de teste, com base nos riscos identificados e na estratégia de teste definida, e fornecer um plano de teste geral, com base no ciclo de vida do teste definido.

3.3.1 SG 1 Estabelecer os ativos do processo de teste organizacional

Uma responsabilidade importante da organização de teste é definir, documentar e manter um processo de teste padrão, de acordo com a política e os objetivos da organização. Entre outros, contém uma descrição dos vários tipos e níveis de teste a serem executados. Embora o TMMi não exija processos de teste detalhados, definidos e documentados, modelos frequentemente focados, elaborados com exemplos, são uma ótima abordagem para estabelecer uma maneira comum de trabalhar. Consulte também os três pilares da qualidade e teste Agile de Bob Galen, onde ele identifica os padrões como listas de verificação, modelos e repositórios como um dos pilares dos testes de software Agile [Galen]. Um modelo identifica as atividades necessárias para reunir informações e documentar os resultados implícitos em sua estrutura. Um processo não precisa ser uma sequência estrita de atividades. Quando existem dependências, elas podem ser capturadas por anotações no modelo ou campos em um formulário que não são acessíveis até que outros campos pré-requisitos sejam preenchidos. Os modelos têm valor prático, transmitindo a real intenção de um processo. Os modelos também evitam as ambiguidades comumente encontradas em documentos de processo "prolixos". Como organização Ágil, se você tiver processos eficazes, pode ser necessário apenas documentá-los de maneira leve e talvez adicionar algumas coisas. Algumas diretrizes-chave frequentemente usadas na organização Agile ao estabelecer processos:

- O processo "preciso fazer" é empacotado separadamente das diretrizes
- Nenhum processo tem mais de duas páginas (objetivo, regra suave)
- Os processos não contêm informações de “como fazer” ou informações de ferramentas, a menos que alguém decida ordenar isso em todo o projeto, independentemente do tamanho e escala.
- As diretrizes separadas contêm opções de adequação/planejamento e informações de como fazer. Observe que, com informações "como fazer", também é possível decidir apenas fazer referência a um livro ou papel existente.

Não é incomum em muitas organizações grandes ver quatro níveis de ativos de processo, como políticas, processos/práticas, instruções/procedimentos de trabalho e facilitadores/modelos. Isso não se deve a algo que o TMMi exige, mas à maneira como muitas grandes organizações optaram por implementar seus ativos de processo. Enquanto a escolha dos ativos do processo depende de cada organização, a maioria das organizações Agile descobriu que dois níveis de ativos do processo são suficientes. Isso é feito consolidando uma declaração de política com a descrição do processo associada que encapsula "o que deve ser feito" na execução do processo. O segundo nível contém diretrizes de “como fazer” para executar o processo e adaptá-lo. Esse nível pode ser visto como um auxílio para adaptar o processo, dando autonomia à equipe para decidir exatamente como trabalhar dentro da estrutura definida e geralmente inclui modelos de suporte. Na maioria das organizações Agile, os procedimentos passo a passo são substituídos por guias de ferramentas e treinamento/mentoria. Conforme declarado acima, um modelo, como modelo de carta de teste, modelo de folha de sessão de teste ou modelo de plano de teste, pode servir como um processo com as atividades de processo necessárias implícitas no modelo. Essa é uma técnica comum que está sendo observada para o desenvolvimento de descrições eficazes do processo Agile. [McMahon]



Estrutura de ativos do processo

O conjunto padrão de processos de teste da organização pode ser adaptado para criar seus processos definidos específicos. No contexto do SP 1.3. Estabeleça critérios e diretrizes de adaptação que a maioria das organizações hoje utiliza uma abordagem chamada de adaptação sob medida. Essa abordagem exige que as pessoas se esforcem explicando por que algo não é necessário. Isso adiciona a maior quantidade de trabalho para os projetos mais simples, o que não é uma abordagem amigável ao Agile. Quando você se adapta, até onde pode ir? Existe um mínimo definido? No entanto, se você começar com um conjunto mínimo em que o conjunto mínimo é o que todos devem fazer, você elimina o risco de adaptar um "deve fazer". Uma abordagem de adaptação é muito mais consistente com as abordagens ágeis e completamente compatível com TMMi. A adequação é um mecanismo extremamente poderoso para organizações ágeis que desejam manter sua agilidade ou para organizações altamente maduras (de teste) que buscam aumentar sua agilidade. Um conjunto de critérios simples pode ajudar a fazer as perguntas certas ao adaptar. Sem critérios para ajudar, alguns tendem a adaptar demais.

Para dar suporte à implantação do processo de teste padrão, uma biblioteca de ativos do processo de teste será criada e mantida, geralmente em um wiki, no qual os testadores podem encontrar, por exemplo, modelos, práticas

recomendadas e listas de verificação que os ajudarão em seu trabalho diário. A criação de uma biblioteca de ativos do processo de teste também faz sentido em um contexto Agile, desde que as equipes obtenham um valor agregado ao usar os ativos fornecidos durante suas atividades de teste. Mais ou menos o mesmo raciocínio vale para um banco de dados do processo de teste no qual dados de teste, por exemplo, estimativas, cobertura, medidas de qualidade, são coletados e disponibilizados para as equipes. Garantir que tudo o que é coletado e disponibilizado para permitir o aprendizado entre equipes e o compartilhamento de experiências tenha um valor agregado para as equipes.

Obviamente, também em um contexto Agile, os padrões do ambiente de trabalho são usados para orientar a criação de ambientes de trabalho do projeto.

3.3.2 SG 2 Integrar os modelos de ciclo de vida de teste aos modelos de desenvolvimento

O modelo de ciclo de vida do teste padrão define as principais atividades e produtos para os vários níveis de teste. O modelo de ciclo de vida de teste padrão está alinhado com os modelos de ciclo de vida de desenvolvimento para integrar as atividades de teste em termos de fases, marcos, entregas e atividades. A integração do ciclo de vida é feita de forma a garantir o envolvimento precoce dos testes nos projetos. Tudo isso parece que não há nada a fazer aqui no caso de um ciclo de vida Agile ter sido adotado, tudo já foi resolvido. No Agile, o desenvolvimento e o teste devem ser totalmente integrados ao ciclo de vida, e os testes já devem estar ocorrendo o mais cedo possível.

Por mais que isso seja verdade em muitas organizações que usam o Agile, também existem organizações que lutam com esse aspecto. Sua iteração se assemelha a um mini ciclo de vida do modelo V para desenvolvimento, onde o teste só começa no final da iteração, geralmente é realizado sob pressão de tempo e às vezes até é interrompido. Obviamente, essa não é uma implementação de um ciclo de vida integrado de desenvolvimento e teste do Agile, como deveria ser. No entanto, mostra que, para algumas organizações que são “ágeis”, esse objetivo específico é uma área a ser aprimorada. É importante que um entendimento comum seja criado, definindo como o teste pode ser envolvido e executado no início da iteração.

Portanto, esse objetivo específico também é aplicável a uma organização que desenvolve Desenvolvimento Ágil de Software. Se feito adequadamente, o desenvolvimento e o teste devem ser totalmente integrados e existe um entendimento comum de como isso é feito. Nesses casos, nada mais precisa acontecer. Em outras circunstâncias, como descrito acima, esse objetivo específico é uma área de melhoria muito importante e deve levar a atividades de desenvolvimento e teste alinhadas muito melhores realizadas em uma iteração.

3.3.3 SG 3 Estabelecer um Plano de Teste principal

No TMMi nível 3, o teste se preocupa com o planejamento de teste principal, que aborda a coordenação das tarefas, responsabilidades e abordagem do teste em todos os níveis. Isso evita redundância desnecessária ou omissões de testes entre os vários níveis de teste e pode aumentar significativamente a eficiência e a qualidade de todo o processo de teste. O plano de teste principal descreve a aplicação da estratégia de teste para um projeto específico, incluindo os níveis específicos a serem executados e o relacionamento entre esses níveis. Um planejamento (teste) do TMMi nível 2 já foi abordado no nível da liberação e da iteração como parte da área de processo do Planejamento de Teste. Uma equipe do Agile geralmente executa vários níveis de teste, que devem ser abordados adequadamente como parte do planejamento de liberação e iteração.

No entanto, alguns projetos que estão trabalhando de acordo com um modelo de ciclo de vida Agile adotaram uma abordagem um tanto híbrida, pois além de testar em uma equipe Agile, alguns testes são organizados fora da equipe. Nessas circunstâncias, um plano de teste principal definirá e gerenciará o relacionamento das equipes Agile com os níveis de teste que são executados fora de seu escopo, por exemplo, testes não funcionais específicos (segurança, desempenho), testes de integração de hardware/software, testes de integração de sistemas ou teste beta. Todas essas práticas específicas relacionadas a esse objetivo específico também serão aplicáveis. Especialmente em projetos maiores e à medida que as organizações crescem, a documentação de um plano mestre de teste geral e decisões relacionadas de maneira consistente começa a ter maior valor. Isso é particularmente verdade quando uma organização está usando estruturas Agile em escala.

Caso todos os testes ocorram com as equipes do Agile, um plano mestre de teste específico não possui valor agregado e, portanto, esse objetivo do TMMi provavelmente não será relevante nesses casos. Aqui, o planejamento no nível de release e iteração deve cobrir todos os aspectos de teste necessários.

3.4 Área de Processo 3.4 Testes Não Funcionais

O objetivo da área de processo de testes não funcionais é melhorar os recursos do processo de teste para incluir testes não funcionais durante o planejamento, a modelagem e a execução do teste.

Se o teste não funcional é importante, é independente do ciclo de vida que está sendo aplicado. O produto em desenvolvimento determinará quais aspectos não funcionais são relevantes para o teste. É claro que isso é verdade para os modelos de ciclo de vida sequencial e modelos de desenvolvimento Agile. Portanto, testes não funcionais também são aplicáveis ao Desenvolvimento Ágil de Software.

No entanto, dependendo da natureza dos aspectos não funcionais e da abordagem de teste utilizada, alguns testes não funcionais, por exemplo, desempenho e confiabilidade, quando testados de maneira tradicional, não podem, como tal, ser testados em uma iteração curta. Como esses testes não funcionais podem, dependendo dos métodos e técnicas usados, levar semanas para configurar, organizar e executar, o tempo de processamento é menos aplicável ao batimento cardíaco das iterações. Portanto, muitas vezes é necessária uma abordagem diferente para testar atributos não funcionais. Essa é uma das mudanças fundamentais para melhor com o ágil, pois os atributos de qualidade já são testados desde o início das iterações e não são deixados até o final, como no modelo tradicional de ciclo de vida. Nem todos os componentes/funções podem estar disponíveis na iteração 1 para realizar um teste não-funcional completo. No entanto, os resultados desse teste podem ser usados para fornecer indicações precoces de problemas de qualidade.

As histórias do usuário devem abordar elementos funcionais e não funcionais para garantir que o produto certo seja desenvolvido para usuários e clientes. As características de qualidade ISO 25010 podem ajudar a estruturar os requisitos e os testadores devem considerar esses elementos não funcionais [ISO25010].

3.4.1 SG1 Executar uma avaliação de risco de produto não funcional

As sessões de risco do produto para projetos ágeis, conforme identificadas e descritas na meta específica SG1 Realizar avaliação de risco como parte da área de processo do planejamento de teste, agora serão explicitamente estendidas para incluir também aspectos e riscos não funcionais. No nível da versão, a avaliação de risco, agora também para testes não funcionais, pode ser realizada com base na visão do produto. No nível da iteração, isso é realizado usando as histórias do usuário que definem requisitos não funcionais como uma entrada importante.

De preferência, todos os membros da equipe, incluindo o proprietário do produto, e possivelmente alguns Stakeholders, devem participar das sessões de risco do produto. Para algumas áreas não funcionais, podem ser necessários especialistas para ajudar. As sessões de risco do produto normalmente resultam em uma lista documentada de riscos do produto priorizados (não) funcionais. Conforme declarado na área de processo Planejamento de teste, em projetos ágeis, o processo que está sendo usado e a documentação resultante serão muito mais leves em comparação com um projeto tradicional, seguindo um modelo de ciclo de vida sequencial.

3.4.2 SG 2 Estabelecer uma abordagem de teste não funcional

Uma abordagem de teste para as características relevantes da qualidade não funcional é definida para mitigar os riscos não funcionais identificados e priorizados do produto. Para uma iteração específica, os recursos não funcionais a serem testados são identificados durante o planejamento da iteração. A lista priorizada de recursos não funcionais a serem testados geralmente se relaciona às histórias de usuário a serem testadas nesta iteração. Novos riscos não funcionais do produto podem se tornar aparentes durante a iteração, exigindo testes adicionais. Questões como novos riscos não funcionais do produto que exigem testes adicionais geralmente são discutidos em reuniões diárias.

A abordagem de teste não-funcional definida no nível da iteração para mitigar os riscos não-funcionais normalmente abrangerá a identificação de métodos e técnicas de teste não-funcionais apropriados e as técnicas de teste com base no nível e tipo de risco não-funcional. Normalmente, ele também abordará o uso de ferramentas de suporte e a abordagem para testar a automação para testes não funcionais. A abordagem de teste para testes não funcionais no nível do release estará em um nível muito mais alto e deve basear-se na estratégia de teste definida no nível do programa ou da organização e também no plano de teste principal definido, se houver. (Consulte SG3 Estabelecer um plano de teste principal na área de processo Ciclo de vida e integração de testes para justificativa para estabelecer ou não estabelecer um plano de teste principal). Tanto a abordagem de teste não funcional no nível de

versão quanto a iteração fazem parte da abordagem geral de teste e serão mantidas ou exibidas no wiki da equipe / projeto.

Os critérios de saída não-funcionais fazem parte da chamada Definição de Concluído (DoD - Definition of Done). É importante que o DoD tenha critérios específicos relacionados a testes não funcionais, por exemplo, MTBF (Tempo Médio entre Falhas) ou "páginas da Web de front-end foram testadas para a lista dos 10 principais riscos da OWASP". A iteração deve resultar na implementação do conjunto acordado de histórias de usuários não funcionais (incluindo seus critérios de aceite) e atender aos critérios de saída não funcionais (teste), conforme definido no DoD. Observe que os atributos não funcionais também podem fazer parte dos critérios de aceite de histórias de usuários funcionais e precisam necessariamente ser especificados como histórias de usuários não funcionais separadas. Além de existir uma Definição de Concluído no nível da iteração, também existe um DoD no nível do release, abrangendo várias iterações. Além disso, o DoD no nível do release pode ter critérios não funcionais relacionados.

3.4.3 SG 3 Executar a análise e a modelagem de teste não-funcional

Esse objetivo específico segue amplamente as mesmas práticas, mas agora de uma perspectiva não-funcional, como no objetivo específico SG1 Executar análise e modelagem de teste usando técnicas de modelagem de teste na área de processa modelagem e execução de teste. Durante a análise e a modelagem do teste, a abordagem de teste para testes não funcionais é convertida em condições e testes tangíveis. No Agile, a análise e a modelagem de testes e a execução de testes são atividades de suporte mútuo que normalmente são executadas em paralelo durante uma iteração. Isso também se aplica à maioria dos testes não funcionais. A análise de teste não funcional, portanto, não é uma atividade separada explícita, mas sim uma atividade implícita que os testadores desempenham como parte de sua função no desenvolvimento colaborativo de histórias de usuários.

Com base na análise das histórias de usuários não funcionais, são identificadas condições de teste não funcionais. As condições de teste são basicamente uma identificação de "coisas" que precisam ser testadas e cobertas. Desde que os critérios de aceite definidos sejam detalhados e suficientemente claros, eles normalmente assumirão o papel das condições de teste tradicionais. Os critérios de aceite são posteriormente traduzidos em testes não funcionais. Com o teste não-funcional, geralmente é benéfico executar a análise de teste em um nível superior, em vez de apenas histórias de usuários. Por exemplo, analisando um recurso ou épico ou uma coleção de histórias para identificar condições de teste não funcionais que são mais abstratas do que aquelas no nível da história do usuário e também abrangem várias histórias do usuário. Com o princípio do primeiro teste sendo aplicado com o Agile, os testes não funcionais que cobrem o conjunto de condições de teste não funcionais serão identificados (e possivelmente automatizados) antes ou pelo menos em paralelo com o desenvolvimento do código.

Para a maioria dos testes não funcionais manuais, os testes serão identificados / refinados à medida que a equipe avança na execução do teste não-funcional. Os testes geralmente não são documentados com tantos detalhes quanto nos projetos tradicionais, mas em um formato de idéias de teste ao usar o teste exploratório. Para áreas não funcionais complexas e críticas, uma abordagem mais tradicional a modelagem de testes e desenvolvimento de casos de teste usando técnicas formais de modelagem de testes não funcionais pode ser a melhor maneira de cobrir os riscos. No entanto, também com essa abordagem, a quantidade de documentação será limitada em comparação com os testes não funcionais em um ambiente tradicional de ciclo de vida sequencial. A priorização de testes não funcionais normalmente segue a priorização da história do usuário que eles estão cobrindo. No entanto, a priorização também pode ser orientada pelo tempo necessário para preparar e executar um determinado teste não funcional.

Dados de teste específicos necessários para suportar a execução de testes não funcionais são identificados. No Agile, os dados de teste necessários geralmente não são especificados primeiro em um documento de especificação de teste. Quando especificado, geralmente é registrado como parte da história do usuário. No entanto, desde que as ferramentas ou funcionalidades necessárias estejam disponíveis, os dados de teste necessários para suportar os testes não funcionais costumam ser criados instantaneamente para permitir um início imediato da execução de testes manuais não funcionais. Por outro lado, para testes não funcionais automatizados, os dados geralmente precisam ser especificados antecipadamente.

A rastreabilidade entre os requisitos não funcionais, condições e testes deve ser estabelecida e mantida. As equipes precisam deixar claro que cobriram várias histórias de usuários não funcionais e critérios de aceite como parte de seus testes. Em muitas organizações ágeis, as histórias de usuários são a base para o desenvolvimento de um conjunto de critérios de aceite relacionados e testes subsequentes. Usando essa abordagem, a rastreabilidade horizontal dos requisitos ao teste pode ser alcançada.

3.4.4 SG 4 Implementar oteste não funcional

A implementação de teste consiste em colocar tudo o que é necessário para iniciar a execução dos testes. Normalmente, o desenvolvimento da documentação de teste, por exemplo, procedimentos de teste, para suportar a execução do teste é minimizado. Os scripts de teste automatizados (de regressão) são desenvolvidos e priorizados.

A implementação de teste não funcional seguirá muitas das práticas já descritas no objetivo específico SG2 Realizar Implementação de Teste como parte da área de processo de Modelagem e Execução de Teste.

O que especificamente precisa ser feito e como a implementação do teste não-funcional é feita depende muito da abordagem definida, das técnicas utilizadas e das características não-funcionais que precisam ser testadas e em que nível. Por exemplo, testes exploratórios, que exigem menos preparação, podem ser adequados para testes de usabilidade, mas geralmente são menos adequados para testes extensivos de confiabilidade e desempenho.

Para algumas características de qualidade não-funcionais, a disponibilidade dos dados de teste é essencial e precisa ser criada durante a implementação do teste. Isso é basicamente o mesmo que nos projetos tradicionais, além de que talvez eles sejam documentados de maneira breve, permitindo ainda a reutilização para testes de regressão.

Obviamente, a implementação e a preparação do teste serão iniciadas o mais rápido possível, paralelamente a outras atividades (de teste). Não é uma fase separada, mas um conjunto de atividades (listadas no quadro de tarefas) que precisam ser executadas para permitir uma execução de teste eficiente e eficaz.

3.4.5 SG 5 Executar o teste não funcional

Como com o objetivo anterior nesta área de processo, nos referiremos amplamente ao objetivo específico relacionado SG 3 Realizar a execução de teste, parte da área de processo Modelagem e Execução de Teste discutida anteriormente neste documento. As práticas para execução de testes não funcionais, relatórios de incidentes de teste e gravação de registros de teste em um ambiente Agile são basicamente as mesmas que para testes funcionais em um ambiente Agile. Normalmente, isso é feito com muito menos intensidade de documentação do que em um projeto tradicional. Frequentemente, nenhum procedimento de teste detalhado e logs de teste são produzidos.

A execução não funcional do teste é realizada de acordo com as prioridades definidas durante o planejamento da iteração. Alguns testes podem ser executados usando um procedimento de teste documentado, mas geralmente também muitos testes não funcionais serão executados usando testes exploratórios e baseados em sessões como sua estrutura. Com o desenvolvimento iterativo, há uma necessidade crescente de organizar e estruturar testes de regressão. Às vezes, isso é feito manualmente, mas de preferência é feito usando testes de regressão automatizados e ferramentas de suporte. O teste de regressão, é claro, também se aplica aos aspectos não funcionais do sistema que foram identificados como importantes para testar.

Os incidentes não funcionais encontrados durante o teste podem ser registrados e relatados pela equipe. Normalmente, há uma discussão nos projetos Agile se todos os incidentes encontrados devem realmente ser registrados. Algumas equipes registram apenas incidentes que escapam às iterações, outras registram se não puderem ser corrigidas hoje, outras registram apenas incidentes de alta prioridade. Se nem todos os incidentes encontrados forem registrados, os critérios deverão estar disponíveis para determinar quais incidentes devem ser registrados e quais não. Às vezes, os incidentes são registrados no painel de tarefas do Agile, como um adesivo em uma tarefa ou como uma tarefa separada, visualizando que ele está impedindo que uma história e suas tarefas sejam concluídas. Alguns optam por registrar e documentar incidentes encontrados usando uma ferramenta, por exemplo, gerenciamento de defeitos ou ferramenta de rastreamento de defeitos. Essa ferramenta deve ser usada o mais enxuta possível e não forçar a apresentação de quaisquer elementos de relatórios de incidentes de teste não funcionais que tenham pouco ou nenhum valor agregado.

Também é considerado uma boa prática nas equipes do Agile registrar dados durante a execução de testes não funcionais para determinar se os itens testados atendem aos critérios de aceite definidos e podem realmente ser rotulados como "concluídos". As informações registradas devem ser capturadas ou resumidas em alguma forma de ferramenta de gerenciamento de status (p.ex.: ferramentas de gerenciamento de teste, ferramentas de gerenciamento de tarefas, quadro de tarefas), de uma maneira que facilite à equipe e às partes interessadas o entendimento do status atual para todos os testes que foram realizados.

3.5 Área de Processo 3.2 Revisões por Pares

O objetivo da área de processo de Revisão por pares é verificar se os produtos de trabalho atendem aos requisitos especificados e remover defeitos dos produtos de trabalho selecionados com antecedência e eficiência. Um efeito

conclusivo importante é desenvolver uma melhor compreensão dos produtos de trabalho e dos defeitos que podem ser evitados.

3.5.1 SG 1 Estabelecer uma abordagem de revisão por pares

As equipes Agile normalmente não fazem revisões formais por pares, no sentido de que geralmente não têm um tempo definido quando as pessoas se reúnem para fornecer feedback sobre um produto. No entanto, eles alcançam a intenção das Revisões por Pares, realizando revisões por pares menos formais ao longo do desenvolvimento. Essa é uma prática comum em muitas organizações ágeis. No entanto, ainda precisa haver uma disciplina ao realizar essas atividades. Esse objetivo do TMMi abrange as práticas para estabelecer uma abordagem de revisão por pares em um projeto. Uma abordagem de revisão define como, onde e quando as atividades de revisão devem ocorrer e se essas atividades são formais ou informais. O estabelecimento de uma abordagem de revisão por pares também é aplicável a projetos Agile, no entanto, geralmente as técnicas de revisão aplicadas e a organização das revisões são muito diferentes.

Exemplos de revisões por pares geralmente executadas em projetos Agile:

- ter sessões de aprimoramento / preparação com as especificações (p.ex.: histórias de usuários) com a equipe e os stakeholders da empresa regularmente durante uma iteração,
- reuniões diárias com outros membros da equipe para discutir abertamente os produtos de trabalho, por exemplo, código ou testes, sendo desenvolvidos e fornecendo feedback,
- a demonstração de produtos antecipada e frequentemente para os clientes, pelo menos no final de uma iteração durante a revisão da iteração.

As especificações ruins geralmente são um dos principais motivos da falha do projeto. Os problemas de especificação podem resultar da falta de percepção dos usuários sobre suas reais necessidades, ausência de uma visão global do sistema, recursos redundantes ou contraditórios e outras falhas de comunicação. No desenvolvimento Agile, as histórias de usuários são escritas para capturar requisitos da perspectiva de representantes comerciais, desenvolvedores e testadores. No desenvolvimento sequencial, essa visão compartilhada de um recurso é realizada por meio de revisões formais após a redação dos requisitos, no desenvolvimento Agile, essa visão compartilhada é realizada por meio de revisões informais frequentes enquanto os requisitos estão sendo estabelecidos. Essas sessões informais de revisão geralmente são chamadas de refinamento de lista de pendências ou sessões de preparação de listas de pendências. Durante as reuniões de aprimoramento, o representante comercial e a equipe de desenvolvimento (e os stakeholders, se disponíveis) usam técnicas de revisão para encontrar o nível de detalhe necessário para a implementação e esclarecer questões em aberto.

A revisão quase contínua pela equipe dos produtos de trabalho em desenvolvimento faz parte de toda a abordagem da equipe. Eles são realizados com a intenção de identificar defeitos precocemente e também de identificar oportunidades de melhoria. Métodos e técnicas Agile como XP ou emparelhamento também incluem revisões por pares como uma prática essencial para criar loops de feedback para as equipes. Obviamente, em caso de alta complexidade ou risco, a equipe pode optar por aplicar uma técnica de revisão semi-formal ou formal, por exemplo, inspeção. Nesses casos, há uma justificativa clara para gastar esforços em uma revisão mais formal e aplicar uma maneira mais disciplinada de trabalhar. Uma parte dos critérios da abordagem de revisão pode ser definida quando usar uma técnica de revisão mais formal.

Enquanto a engenharia de requisitos enfatiza a validação de requisitos por meio de métodos como revisões informais, orientações, inspeções ou leituras baseadas em perspectiva, os métodos Agile se esforçam para validar os requisitos por meio de feedback frequente e precoce para implementar rapidamente incrementos valiosos do produto. A necessidade de validação formal antecipada é reduzida, mostrando resultados rápidos na forma de incrementos de produtos integrados. Se o incremento não atender completamente aos requisitos dos stakeholders, o delta será colocado novamente na lista de pendências do produto na forma de novos requisitos e priorizado com todos os outros itens da lista de pendências. Uma demonstração do que realmente foi construído durante uma iteração é simplesmente uma maneira muito eficiente de energizar uma conversa baseada em validação em torno de algo concreto. Nada fornece foco para a conversa como ser capaz de realmente ver como algo funciona. A demonstração é uma atividade que é executada durante a revisão da iteração. Os recursos são demonstrados e discutidos com os stakeholders, as adaptações necessárias são feitas no backlog do produto ou no plano de liberação para refletir os novos aprendizados da discussão.

3.5.2 SG 2 Realizar as revisões por pres

Obviamente, como qualquer abordagem, ela deve não apenas existir como uma abordagem definida e acordada, mas deve ser respeitada. A equipe precisa gastar uma quantidade substancial de tempo em sessões de aprimoramento da lista de pendências durante uma iteração, aplicando revisões formais informais (e possivelmente) como parte de sua rotina diária e ter demonstrações dos stakeholders regularmente. Demonstrações aos stakeholders / clientes são esperadas pelo menos no final de cada iteração.

Os critérios de entrada geralmente são aplicáveis apenas com revisões formais e, portanto, provavelmente serão menos ou não relevantes nos projetos Agile. No entanto, o uso de critérios de saída é aplicável e agregou valor. INVEST [Wake] é um exemplo de um conjunto de critérios de saída comumente mencionados no projeto Agile ao revisar e atualizar histórias de usuários. INVEST é um acrônimo que abrange os seguintes conceitos que compõem uma boa história de usuário:

- Independente
- Negociável
- Valioso
- Estimável
- Pequeno
- Testável.

Certamente existem outros critérios de saída relacionados à qualidade e podem ser usados também.

É importante que o testador, fazendo parte da equipe Agile, participe das sessões de revisão. Isso é especialmente verdadeiro para as sessões nas quais os produtos de trabalho são discutidos e revisados, usados como base para testes em toda a iteração, por exemplo, nas sessões de refinamento de pendências. Recomenda-se a presença de pelo menos um desenvolvedor e um testador ao refinar a lista de pendências, para garantir a presença de pontos de vista alternativos do sistema. Normalmente, a perspectiva exclusiva do testador melhorará a história do usuário, identificando detalhes ausentes ou requisitos não funcionais. Um testador pode apoiar especialmente a identificação e definição de critérios de aceite para uma determinada história do usuário. Um testador contribui perguntando aos representantes comerciais perguntas "e se?" Abertas sobre a história do usuário, propondo maneiras de testar a história do usuário e confirmando os critérios de aceite.

Prática específica 2.3 A análise dos dados da revisão por pares é especialmente relevante nas revisões formais. Com as revisões formais, muito esforço é gasto, para garantir que o esforço seja gasto de maneira eficiente e eficaz, os dados da revisão por pares são reunidos e comunicados à equipe para aprender e ajustar o processo de revisão. Conforme declarado, revisões formais, por exemplo, inspeção, são menos comuns em projetos Agile. Embora a coleta de dados seja uma parte essencial das revisões formais, é muito menos comum na revisão informal. Como resultado, pode-se argumentar que a coleta, análise e comunicação detalhadas de dados de revisão por pares são menos ou mesmo irrelevantes em um contexto Agile. No contexto de decidir coletar dados ao fazer análises por pares, faça as seguintes perguntas:

- Quem usará esses dados se estiverem sendo coletados?
- Como esses dados se relacionam com nossos objetivos de negócios?

Se ninguém usar os dados de maneira significativa, não desperdice recursos valiosos para coletá-los. Como conclusão, para a maioria das práticas específicas do projeto Agile 2.3. Analisar os Dados de Revisão por Pares será considerado provavelmente não relevante. Observe que alguns dados básicos de revisão valiosos ainda serão coletados e usados no contexto das práticas genéricas, por exemplo, GP 2.8 Monitorar e Controlar o Processo.

4 TMMi Nível 4 Medido

Conforme declarado anteriormente neste documento, é importante lembrar que há uma ruptura natural entre o TMMi nível 3 e o TMMi níveis 4 e 5. Continua a haver controvérsia significativa sobre o valor de mudar uma organização para o TMMi nível 4 e 5. Especialmente Agile. Recomenda-se que as organizações escolham criticamente e pratiquem apenas as práticas nos níveis 4 e 5 do TMMi que são importantes e que agregam valor. Embora o TMMi seja abrangente, para ter sucesso, as organizações devem identificar as principais práticas de teste e melhorias que requerem foco. Interessante, Jeff Sutherland, co-fundador do *Scrum*, publicou um artigo sobre o valor do uso do *Scrum* junto com práticas de níveis mais altos de maturidade [*Sutherland et al*]. A seguir, serão fornecidas informações sobre os níveis 4 e 5 do TMMi sobre como se pode obter valor desses níveis mais altos de TMMi e suas práticas, usando-os menos formalmente, juntamente com uma abordagem Agile.

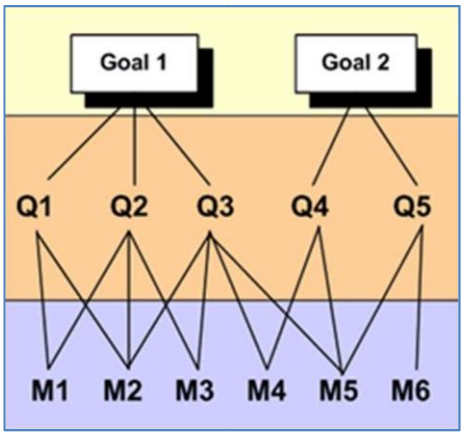
Em primeiro lugar, contrastando um pouco com o que foi afirmado, pode-se considerar também o uso seletivo de práticas de nível 4 e 5 do TMMi com técnicas ágeis para abordar os principais objetivos de negócios, enquanto tenta atingir o nível 2 ou 3. do TMMi. Você não precisa esperar e não deve. Seja flexível e não fique preso às práticas de teste TMMi nível 2 e 3. Use o modelo com base em seus direcionadores de negócios e use áreas, objetivos e práticas de processo que tenham mais valor. Dessa forma, use o TMMi em um modo mais contínuo até certo ponto e não use os níveis de maturidade muito estritamente.

4.1 Área de Processo 4.1 Medição de Teste

O objetivo da Medição de Teste é identificar, coletar, analisar e aplicar medições para apoiar uma organização na avaliação objetiva da eficácia e eficiência do processo de teste, a produtividade de sua equipe de teste, a qualidade do produto resultante e os resultados da melhoria do teste. Como tal, a organização de teste desenvolverá e manterá um recurso de medição de teste usado para apoiar as necessidades de informações de gerenciamento.

As abordagens Agile ajudam a obter o comprometimento daqueles que precisam executar e medir com precisão o trabalho em projetos. No entanto, as abordagens Agile não tendem a fornecer muito suporte quando se trata de recuar para garantir que as medidas corretas sejam usadas em toda a organização e garantir que elas as medidas estão alinhadas com os objetivos de medição derivados das necessidades de negócios da organização. Esses são os pontos fortes que a área de processo de medição de teste TMMi pode trazer para as organizações Agile.

4.1.1 SG 1 Alinhar as atividades de análise e medição de teste



Com demasiada frequência, os programas de medição parecem ter como coletar o máximo de dados possível. Um programa de medição deve ser altamente focado, apenas coisas importantes devem ser coletadas. Comece definindo objetivos claros para o programa de medição e revise-os regularmente, por exemplo, "Ele deve permitir que tenhamos mais sucesso nos testes em nossos projetos". Em seguida, discuta com os stakeholders quais métricas de teste realmente apoiarão esses objetivos. Com base nesse brainstorm, um conjunto limitado de métricas principais será definido.

O objetivo-pergunta-métrica (GQM) [Van Solingen e Berghout] é um método altamente prático que apoiará essa abordagem. São necessárias medidas relevantes ao contexto específico com base nas necessidades do negócio. As métricas, padrão da empresa, geralmente são insuficientes para aprimoramento real do processo (teste). O objetivo das medições é orientar as decisões. Use pequenas equipes capacitadas para obter medidas significativas e, em seguida, revise e refine-as em ciclos curtos.

Com um programa de medição de teste focado a partir do valor agregado (comercial), a área de processo de Medição de Teste TMMi está totalmente alinhada com o princípio Agile de simplicidade. Verifique se há uma estratégia de medição de teste e coleta de dados enxuta dentro da organização. Vale ressaltar que é possível evoluir um programa de medição de teste começando com apenas alguns objetivos de medição focados e algumas medidas-chave e expandi-las ou modificá-las mais tarde, conforme as necessidades dos negócios evoluam e mudam.

Observe que, no Agile, o foco será mais baseado em equipe e pensamento sistêmico. Isso pode resultar em uma ampliação correspondente de algumas das métricas para a equipe e para o sistema geral, em vez de se limitar apenas às especificidades do próprio teste. Isso pode levar a mais desafios na fase de análise de medição.

É claro que as métricas que serão definidas e para quem os dados serão coletados dependerão dos objetivos da medição, mas alguns exemplos de métricas de teste típicas para equipes Agile são:

- Tempo de ciclo do defeito; As equipes Agile devem se esforçar para corrigir os defeitos o mais rápido possível. De fato, um dos principais objetivos da abordagem colaborativa Agile é corrigir defeitos mais rapidamente, para que o software seja lançado mais cedo.
- Transbordamento de defeitos (número de defeitos adiados para uma versão futura); As equipes Agile visam produzir software de trabalho a cada iteração. A propagação de defeitos mede os defeitos que não são corrigidos durante uma determinada iteração ou sprint simplesmente contando os defeitos restantes no final de cada sprint ou iteração. Em algumas organizações ágeis, isso é chamado de dívida técnica.
- Número de defeitos encontrados durante uma iteração
- Número de defeitos encontrados após liberação para produção / clientes

- Número de defeitos encontrados por pessoas fora da equipe
- Número de solicitações de suporte ao cliente
- Porcentagem de cobertura de teste automatizada
- Porcentagem de cobertura de código.

4.1.2 SG 2 Fornecer resultados da medição do teste

Depois que as medidas relevantes são definidas com base nas necessidades dos negócios, as práticas específicas nesse objetivo específico são geralmente aplicáveis da mesma maneira que em uma organização tradicional. A única diferença é que isso será feito com uma mentalidade diferente, mantendo as práticas o mais enxutas possível.

Os dados de medição de teste necessários são coletados e verificados quanto à integridade e integridade, posteriormente, os dados são analisados conforme o planejado e os resultados são comunicados a todos os stakeholders relevantes.

4.2 Área de Processo 4.2 Avaliação da Qualidade do Produto

O objetivo da Avaliação da Qualidade do Produto é desenvolver um entendimento quantitativo da qualidade dos produtos e, assim, apoiar as conquistas das metas de qualidade do produto de projetos específicos.

4.2.1 SG 1 Estabelecer metas mensuráveis de projeto para a qualidade do produto e suas prioridades

Em essência, não há diferença real na identificação e definição de metas priorizadas mensuráveis para a qualidade do produto em um contexto Agile, em comparação com isso em um ambiente tradicional. O objetivo geral é contribuir para satisfazer as necessidades e desejos dos clientes e usuários finais por produtos de qualidade. Existem várias opções para fazer isso em um contexto Agile. Alguns usam a Definição de Concluído para definir objetivos mensuráveis para a qualidade do produto, outros usam recursos, épicos ou histórias de usuários para defini-los, onde também usarão critérios de aceite para torná-los mensuráveis. A escolha (forma de trabalhar) depende, entre outras coisas, da natureza da meta de qualidade do produto. Consulte [ISO25010] para uma visão geral dos atributos de qualidade.

O Definição de Concluído (DoD - Definition of Done) é uma lista de verificação abrangente das atividades necessárias da equipe ou produtos de trabalho que garantem que apenas histórias de usuários realmente feitas sejam entregues, não apenas em termos de funcionalidade, mas também em termos de qualidade do produto. A vantagem de indicar as metas de qualidade do produto no DoD é que elas as mantêm visíveis e, portanto, acionáveis por toda a equipe, à medida que os incrementos são desenvolvidos. É óbvio que as metas de qualidade precisarão ser globalmente aplicáveis ao aplicativo para que ele seja mencionado no DoD. Se os objetivos de qualidade não são aplicáveis globalmente, provavelmente a melhor maneira alternativa de trabalhar é defini-los como um recurso ou épico.

Às vezes, inicialmente "apenas", uma lista de alto nível de atributos relevantes da qualidade do produto é especificada no nível da liberação. Eles serão elaborados gradualmente durante a sessão de aprimoramento como um recurso, épico ou história do usuário. Posteriormente, no início de uma iteração, a equipe considerará se um ou mais dos objetivos de qualidade do produto especificados farão parte do objetivo dessa iteração específica.

4.2.2 SG 2 quantificar e gerenciar o progresso real para alcançar as metas de qualidade do produto do projeto

Uma vez definidas as metas quantitativas de qualidade do produto, elas serão monitoradas, controladas e possivelmente ajustadas quando necessário. Como a qualidade do produto será definida como parte do DoD ou como recursos específicos, épicos ou histórias de usuários com critérios de aceite mensuráveis, os mesmos métodos e técnicas podem ser usados para rastrear e gerenciar o progresso para alcançar as metas de qualidade do produto, conforme explicado anteriormente. e descrito na área de processo 2.3 Monitoramento e controle de teste no TMMi Nível 2. O progresso será medido diariamente e discutido através da reunioes diárias relatando o progresso, qualidade e risco. Além disso, a avaliação da qualidade do produto também é normalmente suportada pela área de processo Medição de teste, que fornece uma infraestrutura de medição.

4.3 Área de Processo 4.3 Revisões Avançadas

O objetivo das Revisões Avançadas, com base nas práticas da Revisão por Pares da área de processo do nível 3 do TMMi, é medir a qualidade do produto no início do ciclo de vida e aprimorar a estratégia e a abordagem de teste alinhando as revisões por pares (teste estático) com o teste dinâmico.

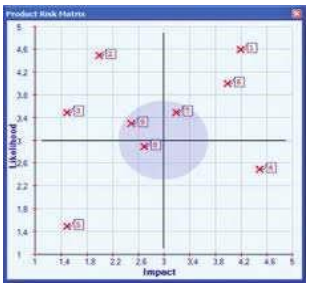
4.3.1 SG 1 Coordenar a abordagem de revisão por pares com a de teste dinâmico

Enquanto nos projetos tradicionais o teste estático, por exemplo, revisões, geralmente não está vinculado ao teste dinâmico sendo realizado por uma equipe de teste independente, espera-se que seja totalmente diferente em um contexto Agile. Em um ambiente ágil, a abordagem para garantir a qualidade do produto deve ser aquela em que todas as atividades relacionadas ao teste, estáticas e dinâmicas, são organizadas como uma abordagem coerente e coordenada. É claro que é responsabilidade da equipe Agile integrar testes estáticos e dinâmicos e encontrar a combinação ideal. Espera-se que a equipe Agile assuma a liderança no estabelecimento de uma abordagem integrada coerente e eficaz. A abordagem para testes estáticos (revisão por pares) será, assim, alinhada e coordenada com a abordagem para testes dinâmicos.

4.3.2 SG 2 Avaliar a qualidade do produto no início do ciclo de vida por meio de Revisão por pares

No nível 4 do TMMi, a organização define metas quantitativas para produtos de software e produtos de trabalho relacionados. As revisões pelos pares desempenham um papel essencial na consecução desses objetivos. Enquanto no TMMi, as revisões por pares do nível 3 são realizadas principalmente para encontrar defeitos, a ênfase está agora na medição da qualidade do produto ou documento para controlar a sua qualidade no início do ciclo de vida. As práticas de revisão são aprimoradas para incluir práticas como amostragem, aplicação de critérios de saída e regras de prescrição. Na verdade, isso é o que Tom Gilb chama de inspeção ágil, trata-se de mudar a ênfase da limpeza para a amostragem, medição, motivação e prevenção de defeitos [Gilb]. Essa é claramente uma prática que se ajustará a organizações Agile de alta maturidade. Outro exemplo seria usar as regras do INVEST [Wake] para histórias de usuário como uma Definição de Pronto (DoR) e medir a qualidade das histórias de usuário, analisando-as em relação a essas regras. Ter uma definição de pronto significa que as histórias devem ser imediatamente acionáveis.

4.3.3 SG 3 Ajustar a abordagem de teste com base nos resultados da revisão no início do ciclo de vida



Quando os resultados da revisão por pares e os testes dinâmicos são coordenados, os resultados e os dados da revisão antecipada podem influenciar os riscos do produto e a abordagem do teste. Conforme já declarado no objetivo específico 1 desta área de processo, no Agile, a qualidade é um esforço da equipe e os resultados das atividades baseadas em verificação e validação serão discutidos nas reuniões da equipe. O princípio de usar os resultados dos testes iniciais para influenciar atividades sucessivas de teste, portanto, tem uma correspondência quase perfeita com a maneira de trabalhar nas equipes e projetos Agile.

5 TMMi Nível 5 Otimização

Os objetivos e práticas no TMMi nível 5 são derivados de abordagens de gerenciamento de qualidade de alta maturidade e melhoria de processos. Nesse alto nível de maturidade, as metas e práticas são quase independentes do ciclo de vida do software que está sendo aplicado. É claro que uma organização enxuta baseada no Agile abordará as coisas de maneira diferente das organizações tradicionais hierárquicas, mas as práticas essenciais em torno do controle de qualidade, otimização de processos e prevenção de defeitos permanecerão praticamente inalteradas. A justificativa para isso é que essas práticas são consideradas práticas de gerenciamento da qualidade, geralmente realizadas em grande parte em um nível superior ao nível de desenvolvimento do produto operacional. Portanto, essas práticas não são diretamente relacionadas às atividades principais de desenvolvimento ou teste de software em um fluxo de entrega de projeto ou produto.

5.1 Área de Processo 5.1 Prevenção de Defeitos

O objetivo da Prevenção de Defeitos é identificar e analisar causas comuns de defeitos ao longo do ciclo de vida do desenvolvimento e definir ações para impedir que defeitos semelhantes ocorram no futuro.

As organizações Agile geralmente já se concentram na prevenção de defeitos, que é uma base importante sobre a qual os objetivos e práticas específicas dessa área de processo podem ser construídos. Organizações Agile eficientes focam na prevenção de defeitos, e não apenas na descoberta de defeitos. Uma cultura de prevenção de defeitos inclui a separação de defeitos "em processo", de defeitos que escapam além da iteração na qual as histórias de usuários são desenvolvidas.

A retrospectiva Agile é uma prática para as equipes refletirem, aprenderem e se tornarem continuamente melhores no que fazem. Embora as retrospectivas sejam usadas com mais frequência para explorar a maneira atual de trabalhar, elas também podem ser usadas para investigar problemas de qualidade ou para acordar ações que possam melhorar a qualidade do software entregue.

A área de processo Prevenção de Defeitos aborda as práticas para identificar e analisar causas comuns de defeitos em equipes, produtos e fluxos de valor e definir ações específicas para remover as causas comuns desses tipos de defeitos no futuro. Todos os defeitos, sejam encontrados durante o desenvolvimento, teste ou em campo, estão dentro do escopo da área de processo. Como a prevenção de defeitos nesse nível precisa de dados de medição, a prevenção de defeitos se baseia nas práticas de medição do nível 4 do TMMi e nos dados de medição disponíveis sobre desenvolvimento, testes e qualidade do produto.

Existem basicamente dois tipos distintos de atividades de prevenção de defeitos nas organizações Agile: nível de equipe e nível de equipe cruzada. As atividades de prevenção de defeitos relacionadas à equipe são de responsabilidade da equipe e as que já foram abordadas no TMMi nível 3, as atividades de prevenção de defeitos entre equipes são executadas por um Grupo de Processo de Teste ou guilda de teste e esses são os que são especificamente abordados por esta área de processo.

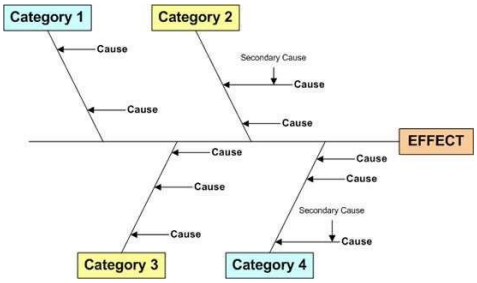
5.1.1 SG 1 Determinar causas comuns de defeitos

As melhorias no Agile, incluindo melhorias no processo de teste, geralmente ocorrem com base em loops de feedback frequentes. Como o escopo geralmente é limitado à iteração anterior, são feitas pequenas, mas frequentes melhorias que se concentram principalmente na solução de problemas locais específicos. O foco dessas melhorias geralmente não é o aprendizado entre equipes e a institucionalização de melhorias. A solução de problemas apenas no nível da equipe também poderia facilmente levar à sub-otimização e à perda de contato com o cenário geral. É aqui que esta área de processo do TMMi agrega valor, com base nas práticas Agile existentes, a prevenção de defeitos agora também é aplicada no nível organizacional (entre equipes, produtos e fluxos de valor) e geralmente gerenciada por um Grupo de Processos (Teste) ou associação de teste (consulte ponto 3.1.1).

A seleção dos defeitos a serem analisados deve ser baseada em vários fatores, incluindo risco e valor do negócio. É necessário focar nas áreas em que a prevenção de defeitos tem o maior valor agregado (geralmente em termos de custo ou risco reduzido) ou onde os defeitos são mais críticos. Embora a área do processo de prevenção de defeitos seja comumente usada para defeitos, ela também pode ser usada para problemas como problemas de velocidade.

Embora um Grupo de Processos de Teste possa coordenar as atividades de prevenção de defeitos, isso deve ser feito por uma equipe multidisciplinar, por exemplo, com representantes da engenharia de requisitos, engenharia de sistemas ou desenvolvimento de software, pois as ações de melhoria geralmente afetam outras disciplinas. Obviamente, isso já é comum nos métodos e equipes de entrega Agile. Garantir, no entanto, que todas as disciplinas necessárias estejam envolvidas, conforme as ações de melhoria.

Os métodos analíticos para avaliar as causas principais dos defeitos, como diagramas de causa-efeito, diagramas de espinha de peixe ou os 5 motivos, geralmente são usados para prevenção de defeitos e práticas já ágeis comuns.



Todas as três práticas específicas (1.1 Definir parâmetros de seleção de defeitos, 1.2 Selecionar defeitos para análise e 1.3 Analisar causas principais e comuns de defeitos selecionados), conforme definido pelo modelo TMMi para esse objetivo específico, também são aplicáveis a uma organização Agile.

5.1.2 SG 2 Priorizar e definir as ações para eliminar sistematicamente as causas comuns de defeitos

Não há diferenças reais na execução de práticas específicas sob esse objetivo específico, seja praticado em um ciclo de vida ágil ou sequencial. Obviamente, a solução real proposta e a proposta de melhoria subsequente serão diferentes, pois precisam se encaixar na maneira de trabalhar com Agile, por exemplo, criando itens adicionais de lista de pendências em vez de propostas de ação formalizadas com aprovações. Ações apropriadas são tomadas para minimizar a probabilidade do problema recorrente. Geralmente, isso significa combinar a melhoria do processo com orientação ou treinamento adicional para os membros da equipe e outros funcionários. É preciso abordar questões de pessoas e processos de maneira integrada, em vez de tentar separá-las artificialmente.

As organizações Agile também são construídas com equipes de autogerenciamento. Isso implica que as equipes podem escolher quais melhorias implementar ou rejeitar. Assim, durante as ações de priorização e definição, é necessário envolver uma representação adequada das equipes para discutir se essas melhorias realmente agregam valor a elas.

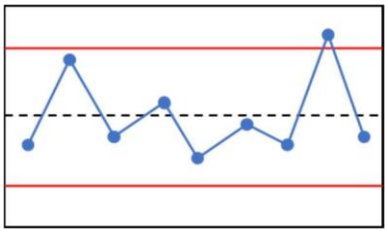
Além disso, a maneira como as melhorias são implantadas será diferente, será menos descendente e mais parecida com uma oferta voluntária às equipes de autogerenciamento. Mais informações sobre a implantação de aprimoramentos de teste podem ser encontradas na área de processo, abaixo.

5.2 Área de Processo 5.2 Controle de Qualidade

O objetivo do controle de qualidade é gerenciar e controlar estatisticamente o processo de teste. Nesse nível, o desempenho do processo de teste é totalmente previsível e estabilizado dentro de limites aceitáveis. Os testes no nível do projeto são realizados usando métodos estatísticos baseados em amostras representativas para prever a qualidade do produto e tornar o teste mais eficiente.

5.2.1 SG 1 Estabelecer um processo de teste estatisticamente controlado

Os princípios e fundamentos para o controle de qualidade são os mesmos, independentemente do método do ciclo de vida que está sendo aplicado. O controle de qualidade do processo envolve o estabelecimento de objetivos para o desempenho do processo de teste padrão. Com base nas medições feitas no desempenho do processo de teste pelas equipes, são realizadas análises e são feitos ajustes para manter o desempenho do processo de teste dentro de limites aceitáveis. Quando o desempenho do processo de teste é estabilizado dentro de limites aceitáveis, o processo de teste definido, as medições associadas e os limites aceitáveis para medições são estabelecidos como linha de base e usados para controlar estatisticamente o desempenho do processo de teste. A capacidade do processo de teste do processo de teste padrão da organização, ou seja, o desempenho do processo de teste que uma equipe pode esperar alcançar, agora é totalmente compreendida e conhecida. Como resultado, os desvios dessas expectativas podem ser atendidos em uma equipe com antecedência e consistência, para garantir que eles tenham um desempenho dentro dos limites aceitáveis.



É importante selecionar subprocessos para o controle estatístico que são críticos para os negócios e que tiveram problemas no passado. A experiência indica que, na maioria das vezes, os subprocessos que melhor atendem a esses critérios não são isolados para equipes individuais, mas cruzam várias equipes, projetos e fluxos de entrega de produtos. Ao usar métodos e técnicas Agile, o trabalho é particionado em iterações curtas, nas quais o trabalho concluído em cada iteração envolve a execução de várias atividades de desenvolvimento e teste. Portanto, como resultado, o desenvolvimento Agile suporta, mais efetivamente do que o desenvolvimento tradicional, o monitoramento dos subprocessos mais valiosos para controle estatístico, pois os ciclos são muito mais curtos, levando a um feedback quase instantâneo sobre o desempenho do processo de teste. O feedback rápido geralmente é altamente valioso para as equipes e um importante princípio do Agile.

5.2.2 SG 2 O teste é realizado usando métodos estatísticos

O controle de qualidade do produto baseia-se em perfis operacionais [Musa] e modelos de uso do produto em seu ambiente pretendido para fazer inferências estatisticamente válidas, resultando em uma amostra representativa de testes. Essa abordagem, especialmente útil no nível do sistema, usa métodos estatísticos de teste para prever a qualidade do produto com base nessa amostra representativa. Em outras palavras, ao testar um subconjunto de todos os usos possíveis, conforme representado pelo perfil de uso ou operacional, os resultados do teste podem servir de base para conclusões sobre o desempenho geral do produto.

Essa é basicamente uma abordagem que se ajusta bem ao Agile, porque é enxuta por natureza e é impulsionada pelo feedback do uso operacional e das sessões de revisão sobre o produto. No entanto, exige que uma organização Agile tenha o nível certo de maturidade para poder aplicar essa abordagem. Também constrói fortemente e depende de medições e dados coletados e disponíveis.

Uma organização que usa métodos estatísticos em testes é capaz de quantificar níveis de confiança e confiabilidade (consulte a estrutura do TMMi para obter mais informações sobre esses termos). Tanto o nível de confiança quanto a confiabilidade são normalmente usados como Definição de Concluído ao aplicar testes estatísticos.

5.3 Área de Processo 5.3 Otimização do Processo de Teste

O objetivo da Otimização do Processo de Teste é melhorar continuamente os processos de teste existentes usados na organização e identificar novas tecnologias de teste (p.ex.: ferramentas ou métodos de teste) que possam ser apropriadas e fazer a transição para a organização de maneira ordenada. A Otimização de Processos de Teste também suporta a reutilização de ativos de teste em toda a organização. As melhorias suportam a qualidade do produto da organização e os objetivos de desempenho do processo de teste, derivados dos objetivos de negócios da organização.

5.3.1 SG 1 Selecionar as melhorias no processo de teste

As propostas de melhoria de teste são coletadas e analisadas. As propostas são de fontes internas e, geralmente, entre equipes, projetos e fluxo de produtos. As fontes internas incluem questões e ideias provenientes de atividades de prevenção de defeitos, revisões de equipes, retrospectivas, capítulos e associações. Após uma análise de custos e benefícios, as interessantes propostas de melhoria de teste são testadas para avaliar mudanças importantes novas e não comprovadas antes de serem implementadas em toda a organização nas equipes, conforme apropriado. Finalmente, é feita uma seleção das melhorias para implantação em toda a organização.

5.3.2 SG 2 Novas tecnologias de teste são avaliadas para determinar seu impacto no processo de teste

Como parte da otimização do processo de teste, o grupo de processos de teste organizado centralmente (ou guilda de testes) busca ativamente no mercado novas tecnologias, como ferramentas, métodos, técnicas ou inovações técnicas, que podem melhorar a capacidade de teste das equipes Agile. Ao manter um conhecimento das inovações tecnológicas relacionadas ao teste e avaliar e experimentar sistematicamente com elas, a organização, em consulta com os representantes da equipe, seleciona tecnologias de teste apropriadas para melhorar a qualidade de seus produtos e a produtividade de suas atividades de teste. Os pilotos são executados dentro das equipes para avaliar tecnologias de teste novas e não comprovadas antes de serem incorporadas à prática padrão.

5.3.3 SG 3 Implantar as melhorias de teste

As melhorias e novas tecnologias de teste apropriadas são implantadas nas equipes Agile para aprimorar os testes. Seus benefícios são medidos e as informações sobre novas inovações são disseminadas pelas organizações. Obviamente, há uma diferença entre a implantação em uma organização tradicional e uma organização Agile. Em uma organização tradicional, a implantação geralmente resulta em novas práticas que devem ser aplicadas por todos os envolvidos. Em uma organização Agile, a equipe tem mais autonomia e é auto gerenciada. Isso também implica que o processo de implantação é diferente. Aprimoramentos de teste e novas tecnologias de teste apropriadas são incorporadas aos ativos do processo de teste organizacional, conforme apropriado, treinamento e suporte são organizados e oferecidos. No entanto, no final, a própria equipe Agile deve decidir se a melhoria do teste se tornará ou não parte de sua maneira de trabalhar.

5.3.4 SG 4 Estabelecer a reutilização de ativos de teste de alta qualidade

No TMMi nível 3, a reutilização de testware entre equipes, projetos e fluxos de produtos já pode ocorrer, no entanto, a reutilização de ativos se torna um objetivo importante no TMMi nível 5. A equipe não deve reinventar a roda, mas sim aproveitar a experiência e os ativos já disponíveis e, assim, economizar esforço e tempo. Os ativos de alta qualidade que provavelmente podem ser reutilizados em toda a organização são identificados, por exemplo, em reuniões de retrospectivas, sessões de lições aprendidas, discussão na associação de teste ou revisões de teste. Os ativos de teste reutilizáveis subsequentemente são selecionados e adicionados ao repositório central de testes em um formato modificável para reutilização em projetos e entrega de produtos futuros.

6 Visão geral, aplicabilidade, objetivos e práticas específicas do TMMi

6.1 Avaliações TMMi

Para ajudar os principais avaliadores e avaliadores em suas avaliações, é fornecida uma visão geral neste capítulo da aplicabilidade das áreas de processo, objetivos específicos e práticas específicas em um contexto Agile. Nos precedentes capítulos, identificamos um número limitado de práticas específicas que se espera serem menos ou talvez até não relevantes. Essas práticas específicas são listadas a seguir. Durante uma avaliação do TMMi, um avaliador líder ou avaliador precisará avaliar explicitamente esses itens quanto à sua relevância no contexto Agile específico. Caso se decida que um item é de fato menos ou mesmo não relevante, espera-se que seja claramente documentado com justificativa tanto nos processos da organização quanto no relatório de avaliação.

6.2 TMMi Nível 2 Gerenciado

Área de processo 2.2 Planejamento de teste

SG2 Desenvolver a Abordagem de Teste	Aplicável
• SP2.3 Definir critérios de entrada	Menos relevante ou irrelevante
• SP2.5 Definir critérios de suspensão e retomada	Menos relevante ou irrelevante
SG3 Estabelecer Estimativas de Teste	Aplicável
• S3.2 Definir o ciclo de vida do teste	Menos relevante ou irrelevante
SG5 Obter compromisso com o Plano de Teste	Aplicável
• SP5.2 Reconciliar níveis de trabalho e recursos	Menos relevante ou irrelevante

Área de Processo 2.3 Monitoramento e Controle de Teste

SG2 Monitorar a qualidade do produto em relação ao plano e expectativas	Aplicável
• SP2.3 Monitorar critérios de entrada	Menos relevante ou irrelevante
• SP2.5 Monitorar critérios de suspensão e retomada	Menos relevante ou irrelevante

Área de Processo 2.4 Modelagem e Execução de Teste

SG2 Executar Implementação de Teste	Aplicável
• SP2.3 Specify intake test	Menos relevante ou irrelevante
• SP2.4 Desenvolver cronograma de execução de teste	Menos relevante ou irrelevante
SG3 Realizar execução de teste	Aplicável
• SP3.1 Executar teste de entrada	Menos relevante ou irrelevante

6.3 TMMi Nível 3 Definido

Área de Processo 3.5 Revisões por Pares

SG2 Realizar revisões por pares	Aplilcável
• SP2.3 Analisar dados de revisão por pares	Menos relevante ou irrelevante

Referências

[Black, Van Veenendaal] Rex Black, Erik van Veenendaal and Dorothy Graham (2012), Foundations of Software Testing ISTQB Certification (3rd edition), Cengage

[Cohn] M. Cohn (2009), Succeeding with Agile: Software Development using Scrum, Addison-Wesley

[Galen] R. Galen (2015), Three Pillars of Agile Quality and Testing: Achieving Balanced Results in your Journey Towards Agile Quality, RGCG, LLC

[Gilb] T. Gilb (2004), Agile Specification Quality Control, International Council on Systems Engineering (INCOSE)

[ISO25010] ISO/IEC 25010 (2011), Systems and Software Engineering – Systems and Software Quality Requirements and Evaluation (SQuaRE) System and Software Quality Models, International Organization for Standardization

[Sutherland et al] C.R. Jakobsen, K. Johnson and J. Sutherland (2007), Scrum and CMMI level 5: The Magic Portion for Code Warriors in: IEEE Agile 2007 Conference, Washington, D.C.

[McMahon] Paul. E. McMahon (2011), Integrating CMMI with Agile Development – Case Studies and Proven Techniques for Faster Performance Improvement, Addison Wesley

[Musa] J. Musa (1998), Software Reliability Engineering Testing, McGraw-Hill Education

[Van der Aalst and Davis] L. van der Aalst and Cecile Davis (2013), TMap NEXT in scrum – Effective testing in agile projects, Sogeti Nederland B.V.

[Van Solingen and Berghout] R. van Solingen and E. Berghout (1999), The Goal/Question/Metric method, McGrawHill

[van Veenendaal] E. van Veenendaal (2014), PRISMA: Product Risk Assessment for Agile Projects, in: Testing Experience, Issue 04/12, December 2012

[Wake] B. Wake (2003), INVEST in Good Stories, and SMART Tasks, xp123.com/articles/invest-in-good-stories-andsmart-tasks