

MÉTODOS ÁGEIS NO DESENVOLVIMENTO DISTRIBUÍDO DE SOFTWARE

Robson de Sousa Martins

FACULDADE DE INFORMÁTICA E ADMINISTRAÇÃO PAULISTA (FIAP)

MBA em Desenvolvimento de Sistemas Corporativos em Java / SOA

Turma 16SCJ

Resumo. *Devido à globalização, muitas organizações possuem desenvolvedores alocados fisicamente em diferentes cidades, estados ou mesmo países, trabalhando colaborativamente em um mesmo projeto. Por outro lado, diversas empresas têm utilizado os métodos ágeis para melhorar o desenvolvimento de software, diminuindo custos e aumentando a qualidade do produto. Como aplicar os métodos ágeis em projetos desenvolvidos por times geograficamente distribuídos, unindo esses dois cenários? Este trabalho pretende discutir algumas vantagens e desafios inerentes a essa abordagem.*

Palavras-chave: Métodos Ágeis. Desenvolvimento Distribuído de Software.

1. INTRODUÇÃO

Há atualmente diversas empresas, atuando nos mais diferentes segmentos, que estão distribuídas geograficamente em várias cidades, estados ou países.

Muitas dessas companhias possuem equipes voltadas ao desenvolvimento de *software*, algumas para suprir suas necessidades internas (sistemas internos), outras para fornecer sistemas aos seus clientes (*software* como a atividade fim).

Algumas empresas optam por manter o time de desenvolvimento atuando em um determinado projeto numa mesma filial ou regional, de modo a não espalhar os seus integrantes em locais físicos diferentes, evitando os problemas relacionados ao desenvolvimento distribuído (SUTHERLAND, 2008).

Outras empresas, por questões de custo, disponibilidade de mão-de-obra e outras razões (como por exemplo, a possibilidade de oferecer trabalho na modalidade *home office*, etc.), preferem manter o time distribuído, com integrantes alocados em diferentes regiões geográficas, mas atuando conjuntamente em um

mesmo projeto.

De outro lado, muitas empresas estão preocupadas com a melhoria do processo de desenvolvimento de *software*, visando redução de custos, aumento na qualidade do produto e a satisfação do cliente. Para materializar esses benefícios, várias dessas organizações têm buscado a implantação de uma nova metodologia de desenvolvimento, denominada metodologia ágil (HIGHSMITH, 2004).

Tendo em vista as vantagens proporcionadas pelos métodos ágeis, diversas empresas que possuem times distribuídos estão dispostas a aplicar suas práticas e princípios ao desenvolvimento de *software*, porém se deparam com diversos problemas e dificuldades causados pela união desses dois cenários.

Este trabalho objetiva trazer à discussão as vantagens, os problemas e as possíveis soluções na utilização de métodos ágeis no desenvolvimento distribuído de *software*, e está organizado nas seguintes seções: “Visão Geral do Desenvolvimento Distribuído de Software”, que aborda os conceitos e os problemas relacionados à distribuição geográfica dos times de desenvolvimento; “Visão Geral sobre Métodos Ágeis”, que explana, de maneira sucinta, as características dos métodos ágeis; “Aplicação de Métodos Ágeis em Times Distribuídos”, que apresenta as consequências da aplicação desses métodos nesses times; e por fim, a “Conclusão”, que traz algumas considerações finais sobre essa abordagem.

2. VISÃO GERAL DO DESENVOLVIMENTO DISTRIBUÍDO DE SOFTWARE

O desenvolvimento distribuído de *software* (também chamado de DDS), surgiu como uma necessidade em empresas que possuem filiais ou regionais espalhadas entre diferentes cidades, estados ou países.

Segundo Carmel (1999), alguns fatores contribuíram para fomentar essa necessidade:

- A vantagem de estar próximo a cada um dos mercados locais, permitindo maior relacionamento com os clientes, independente da sua localização.
- Redução de custos no deslocamento ou contratação de pessoal, permitindo a colaboração dos especialistas seja onde estiverem alocados.
- Incentivos fiscais adotados por governos que querem favorecer a implantação de fábricas e/ou desenvolvimento tecnológico dentro de seus limites ou

fronteiras.

- Aproveitamento das diferenças de fuso horário entre países para aumentar o tempo disponível para desenvolvimento e atendimento das demandas (*follow-the-sun*).

Existem alguns diferentes cenários para o desenvolvimento distribuído:

- **Equipes com grupos espalhados em cada local:** Nesse cenário, cada filial ou regional possui um agrupamento de especialistas. O conjunto dos diversos grupos forma a equipe do projeto. Esta modalidade é muito comum nas empresas onde o desenvolvimento distribuído de *software* já foi adotado há algum tempo, ou em corporações que adquiriram empresas em outras localidades e as incorporaram em sua estrutura.
- **Alguns membros de uma equipe em outro local:** Uma equipe possui a maioria dos seus membros em uma localização, alguns poucos membros estão alocados em locais distintos. Essa forma de trabalho pode também ser aplicada a *home office* (teletrabalho), onde alguns dos membros da equipe podem trabalhar em casa, em determinados momentos.
- **Todos os membros de uma equipe distribuídos geograficamente:** Cada um dos membros da equipe está em uma localização distinta, ou unidos em pequenos grupos (duas, três pessoas), que sozinhos, não poderiam compor uma equipe independente dos outros. Esse modelo é adotado no desenvolvimento de *software open source*, por exemplo, onde cada participante pode estar num país diferente e completamente distante um do outro (RAYMOND, 1999) .

Independente do cenário, o desenvolvimento distribuído de *software* pode promover alguns desafios (CARMEL, 1999; HERBSLEB et. al., 2001; SUTHERLAND, 2008):

- Dispersão geográfica (distância física) – implica na dificuldade de comunicação entre as pessoas, o distanciamento pessoal e a falta de socialização entre elas.
- Dispersão temporal (diferença de fuso horário) – implica na dificuldade de sincronismo entre as pessoas. Dessa forma, uma possibilidade de telefonema ou reunião *real time* remota entre os participantes se torna difícil, pois as

jornadas de trabalho podem não coincidir.

- Dispersão cultural (diferenças culturais ou regionais) – provoca dificuldades de comunicação (idioma diferente, por exemplo) ou conflitos entre os participantes (exemplo: uma forma de cumprimento em um país pode ser interpretada como uma ofensa em outro).
- Dificuldades de implantação de uma infraestrutura capaz de integrar eficientemente os produtos de trabalho de cada membro da equipe.
- Gestão de conhecimento diluída e descentralizada – causa dificuldades no armazenamento, processamento e disseminação das informações produzidas e adquiridas ao longo do desenvolvimento do produto.

As consequências desses problemas podem ser enumeradas: produção de *software* sem qualidade, grande número de retrabalhos e manutenção, falta de cumprimento dos prazos, insatisfação do cliente, perda de oportunidades de negócio, baixa coesão e desmotivação da equipe, dentre outras.

Desta forma, para que o desenvolvimento distribuído de *software* possa ser viável, torna-se necessário observar alguns pontos:

- Adotar uma infraestrutura tecnológica que suporte trabalho colaborativo à distância;
- Prover eficientes meios de comunicação entre os membros da equipe, mesmo que estejam a quilômetros uns dos outros;
- Minimizar as diferenças de idioma e de cultura através da adoção de um idioma comum, realização de treinamentos, conscientização, intercâmbios culturais e outros meios de integração social.

3. VISÃO GERAL SOBRE MÉTODOS ÁGEIS

O desenvolvimento de *software* é uma atividade complexa, pois lida com muitos elementos lógicos e estados, além de sofrer possíveis mudanças de escopo e de requisitos ao longo do seu ciclo de vida (BROOKS, 1987).

Devido a essa complexidade, é comum que projetos de desenvolvimento terminem fracassados ou tenham seu desempenho comprometido, devido ao excesso de manutenção, retrabalhos, aumento de custos, descumprimento dos prazos e insatisfação do cliente.

Nesse cenário, surgiram as metodologias ágeis, como uma forma alternativa de lidar com os problemas intrínsecos ao desenvolvimento de *software*.

Segundo os princípios descritos no Manifesto Ágil (BECK, et. al., 2001):

Nossa maior prioridade é satisfazer o cliente, através da entrega adiantada e contínua de software de valor.

Aceitar mudanças de requisitos, mesmo no fim do desenvolvimento.

Processos ágeis se adequam a mudanças, para que o cliente possa tirar vantagens competitivas.

Desta forma, seguindo uma abordagem adaptativa, os métodos ágeis permitem oferecer maior valor ao cliente, no menor tempo e na maior qualidade possível.

Ainda segundo o Manifesto Ágil (BECK, et. al., 2001), os principais valores são:

1. Indivíduos e interação entre eles, mais que processos e ferramentas.
2. *Software* em funcionamento, mais que documentação abrangente.
3. Colaboração com o cliente, mais que negociação de contratos.
4. Responder a mudanças, mais que seguir um plano.

Existem diferentes métodos ágeis aplicáveis em desenvolvimento de *software*. Dois dos mais conhecidos são o *Scrum* e o *Extreme Programming* (XP).

Além disso, muitas organizações utilizam a combinação de diversas “práticas ágeis” em seu próprio processo de desenvolvimento, em diferentes níveis, como *Test Driven Development* (TDD), programação em pares, integração contínua, e outras.

Essencialmente, os métodos ágeis valorizam curtas iterações, com duração pré-definida, onde o trabalho de desenvolvimento resulta em várias entregas de um produto funcional, completamente testado e de grande valor de negócio ao cliente.

Diferentemente de um processo cascata convencional (ROYCE, 1970), um processo iterativo ágil permite o refinamento constante dos requisitos do produto e do planejamento do projeto, incrementando o resultado final com o que foi realizado em cada iteração.

4. APLICAÇÃO DE MÉTODOS ÁGEIS EM TIMES DISTRIBUÍDOS

Alguns dos princípios descritos no Manifesto Ágil (BECK, et. al., 2001):

Pessoas relacionadas a negócios e desenvolvedores devem trabalhar em conjunto e diariamente, durante todo o curso do projeto.

Construir projetos ao redor de indivíduos motivados, dando a eles o ambiente e suporte necessário, e confiar que farão seu trabalho.

O método mais eficiente e eficaz de transmitir informações para, e dentro de um time de desenvolvimento, é através de uma conversa cara a cara.

Analisando os principais desafios proporcionados pelo desenvolvimento distribuído de *software* e os princípios dos métodos ágeis, pode-se observar que há alguns pontos críticos comuns às duas abordagens:

- Necessidade de comunicação
- Meios para promover o envolvimento, motivação e coesão da equipe
- Ambiente e suporte para favorecer o trabalho colaborativo

Desta forma, os problemas que podem ocorrer em um desenvolvimento distribuído de *software*, usando uma metodologia ágil, podem ser determinantes para o insucesso do projeto.

Fowler (2006) aborda em seu trabalho algumas maneiras para reduzir esses pontos críticos, e possibilitar o desenvolvimento distribuído de *software*, desfrutando das vantagens oferecidas pelas metodologias ágeis.

A seguir, serão explorados alguns tópicos relevantes à aplicação de métodos ágeis no desenvolvimento distribuído de *software*.

4.1. Distribuição e agrupamento da equipe

Como elucidado anteriormente, os métodos ágeis primam por uma comunicação direta e eficiente entre os integrantes de toda a equipe.

Como os integrantes podem estar geograficamente distribuídos em grupos, uma abordagem mais eficiente seria a de ter especialistas de diferentes áreas dentro de todos os grupos (quando isso for possível).

Por exemplo, se um desenvolvedor precisar de uma conversa com um especialista em banco de dados, a comunicação será mais rápida e direta se os dois estiverem no mesmo local físico.

Uma outra questão é relacionada ao *scrum master* (que, na metodologia

Scrum, é o responsável por garantir as condições favoráveis ao trabalho da equipe): como os integrantes de uma equipe estão alocados em grupos geograficamente distribuídos, para agilizar a comunicação entre estes e o *scrum master*, pode ser necessária a existência de várias pessoas com esse papel, uma alocada em cada grupo de desenvolvimento. Esses “*scrum masters* locais” garantiriam as necessidades dos seus grupos regionais, podendo ser auxiliados ainda por um “*scrum master* principal” do projeto (*scrum of scrums*), alocado na sede, por exemplo.

Essa abordagem pode ser adotada para permitir respostas rápidas aos impedimentos e atendimento às necessidades locais de cada um dos grupos da equipe, pois somente os impedimentos mais globais seriam tratados remotamente pelo “*scrum master* principal”.

4.2. Infraestrutura / Ferramentas para comunicação

Comunicação é um dos pontos mais importantes das metodologias ágeis. Desta forma, uma infraestrutura capaz de garantir a comunicação entre os membros distribuídos da equipe se torna extremamente essencial.

Devem ser utilizadas ferramentas que facilitem a comunicação remota (tantas quantas forem necessárias): correio eletrônico (*e-mail*), mensageiros instantâneos (*chats*), listas de discussão (fóruns), *wikis*, portais de *intranet*, videoconferência, telefones para audioconferência, etc.

É igualmente imprescindível que os grupos distribuídos tenham *links* de rede eficientes uns com os outros, para agilizar o compartilhamento e troca de informações. Problemas frequentes na infraestrutura de comunicação podem atrasar o andamento do projeto, dificultar a disseminação do conhecimento dentro da equipe ou mesmo desmotivar os seus membros (que poderão se sentir “ilhados”).

4.3. Infraestrutura / Ferramentas para fluxo e controle das informações

As informações de um projeto (código-fonte, itens de *backlog*, documentação ou outros artefatos) deverão estar atualizadas e acessíveis a qualquer integrante da equipe, seja onde ele estiver alocado geograficamente. Sendo assim, uma infraestrutura de gestão e controle de informações se faz necessária.

Ferramentas que facilitam o desenvolvimento colaborativo devem ser usadas: sistema de controle de versões (repositório), sistema de *builds* e integração contínua (compilações automatizadas), *bug tracking* (rastreamento de defeitos), além de

wikis, portais de *intranet*, e outras ferramentas para gestão de conhecimento.

4.4. Integração e coesão da equipe

Outro aspecto importante das metodologias ágeis é a coesão da equipe, que deve trabalhar unida e focada nos objetivos.

Quando a equipe está geograficamente distribuída, a tendência natural é que cada grupo forme uma “ilha”, isolada dos demais grupos.

Para superar essa dificuldade, podem ser usadas algumas estratégias de socialização entre os grupos que compõem a equipe:

- **Presença de um “embaixador”**: um dos integrantes da equipe é temporariamente transferido e alocado em outra localidade, trabalhando em conjunto com o grupo daquela região. Desta forma, os indivíduos daquele grupo terão a nítida sensação de que são importantes para o projeto e, além disso, poderão trocar experiências técnicas e culturais com esse “embaixador”.
- **Visitas programadas (intercâmbio)**: outra possibilidade para integrar os grupos distribuídos é promover intercâmbios entre eles, onde os membros de um grupo estariam realizando uma excursão para trabalhar pessoalmente com outro grupo. Se possível, alguns encontros pessoais para confraternização (fora do escopo do trabalho) também podem ser realizados regularmente.
- **“Janela para o mundo”**: algumas organizações optam por instalar painéis no ambiente de trabalho, que exibem em “tempo real” o que está acontecendo nas outras regionais ou filiais. Seria como uma janela que visualiza o que está do outro lado da sala, mesmo que esse “outro lado” esteja fisicamente a quilômetros de distância.

4.5. Minimização dos desafios da cultura

As metodologias ágeis definem que a equipe deve ser auto-organizada e pró-ativa, não tomando uma postura de sempre aguardar ordens de um chefe ou gerente.

Em alguns países, por exemplo, as pessoas têm dificuldade de compreender essa forma de trabalho, pois estão acostumadas com uma cultura onde há um comandante (que dá as ordens) e vários subordinados (que as executam).

Sendo assim, as dificuldades culturais para a aplicação dos métodos ágeis podem ser significadamente ampliadas.

Desta forma, há uma maior necessidade de realização de treinamentos e cursos para difundir os princípios e os valores da metodologia ágil, e conquistar a confiança dos envolvidos nos grupos que compõem a equipe.

4.6. Adequando as práticas ágeis ao time distribuído

Por questões de distanciamento geográfico e/ou temporal, algumas práticas ágeis devem ser adequadas conforme a realidade da equipe distribuída:

- **Reuniões de *status* (*daily scrum* ou *stand-up meeting*):** as reuniões de *status* poderão ser marcadas em periodicidade maior do que um dia, devido às dificuldades em se reunir diariamente a equipe. Há empresas que realizam essas reuniões somente duas vezes por semana, via videoconferência, por exemplo.
- ***Scripts* de teste para ajudar a entender os requisitos:** como pode existir um distanciamento geográfico com o cliente (*product owner*, ou representante da área de negócios), é importante que os membros remotos da equipe entendam plenamente os requisitos e sejam capazes de trabalhar corretamente com os itens de *backlog*. Para ajudar a garantir esse entendimento, casos de teste e *scripts* podem ser elaborados em conjunto com a definição dos itens de *backlog*, até mesmo incluindo a participação do cliente nessa definição.
- **Documentação:** no caso do desenvolvimento distribuído, a documentação poderá ter mais importância do que num cenário de equipe local. Essa documentação poderá ser produzida dinamicamente, por exemplo, usando *wikis* e fóruns. Contudo, deve-se evitar a burocratização trazida pela obrigatoriedade de produzir artefatos de documentação em excesso.

4.7. Ferramentas para facilitar as práticas ágeis em equipes distribuídas

Além das ferramentas já discutidas anteriormente, existem outras que facilitam a execução de práticas ágeis no desenvolvimento distribuído de *software*.

Algumas dessas ferramentas, por exemplo, fazem o gerenciamento do *backlog*, exibindo um quadro de tarefas que pode ser acessado e manipulado remotamente pelos membros da equipe.

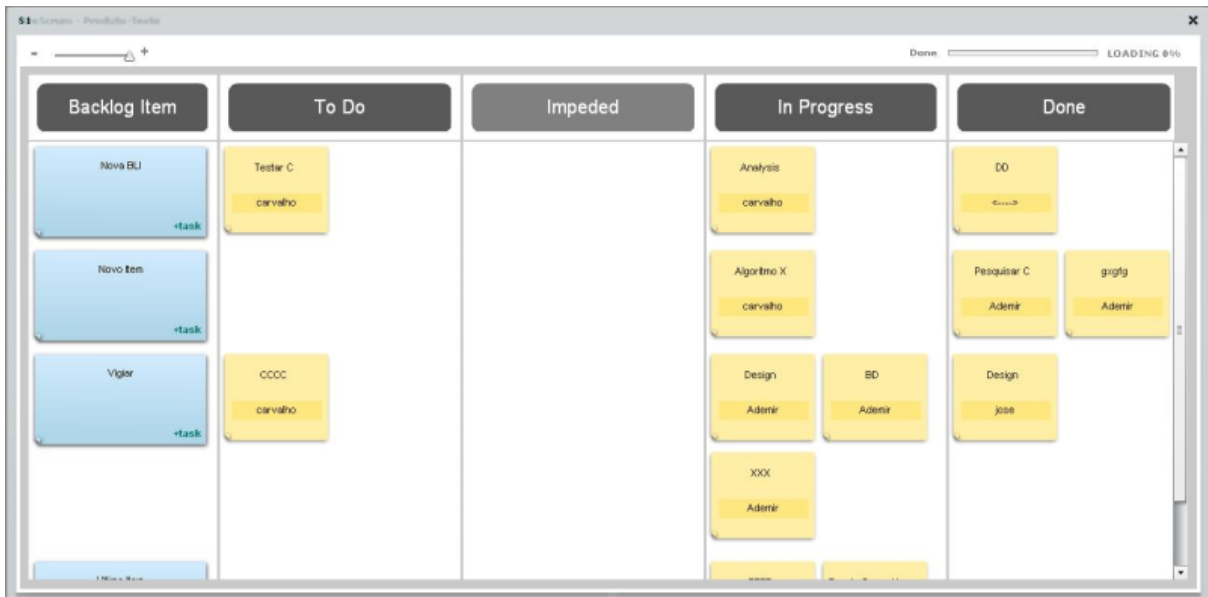


Figura 1 – Quadro de Tarefas em uma ferramenta “ágil” (INES, 2011).

Existem outras ferramentas no mercado que se propõem a gerenciar todo o ciclo de vida de desenvolvimento de uma aplicação de *software* (*Collaborative Application Lifecycle Management – C/ALM*), inclusive usando métodos ágeis, possuindo módulos para comunicação, repositório de informações, planejamento, automação de *builds*, quadro de tarefas, gráficos de *burndown*, etc.

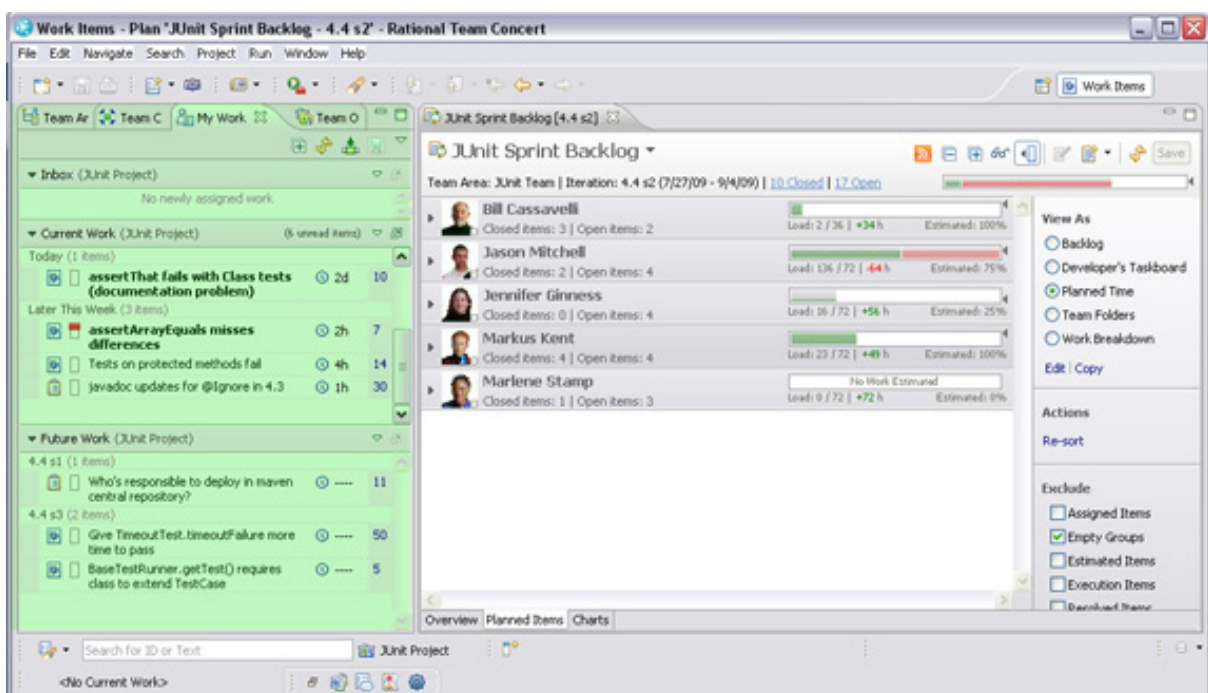


Figura 2 – Visão do progresso de uma iteração em uma ferramenta “C/ALM” (IBM, 2011).

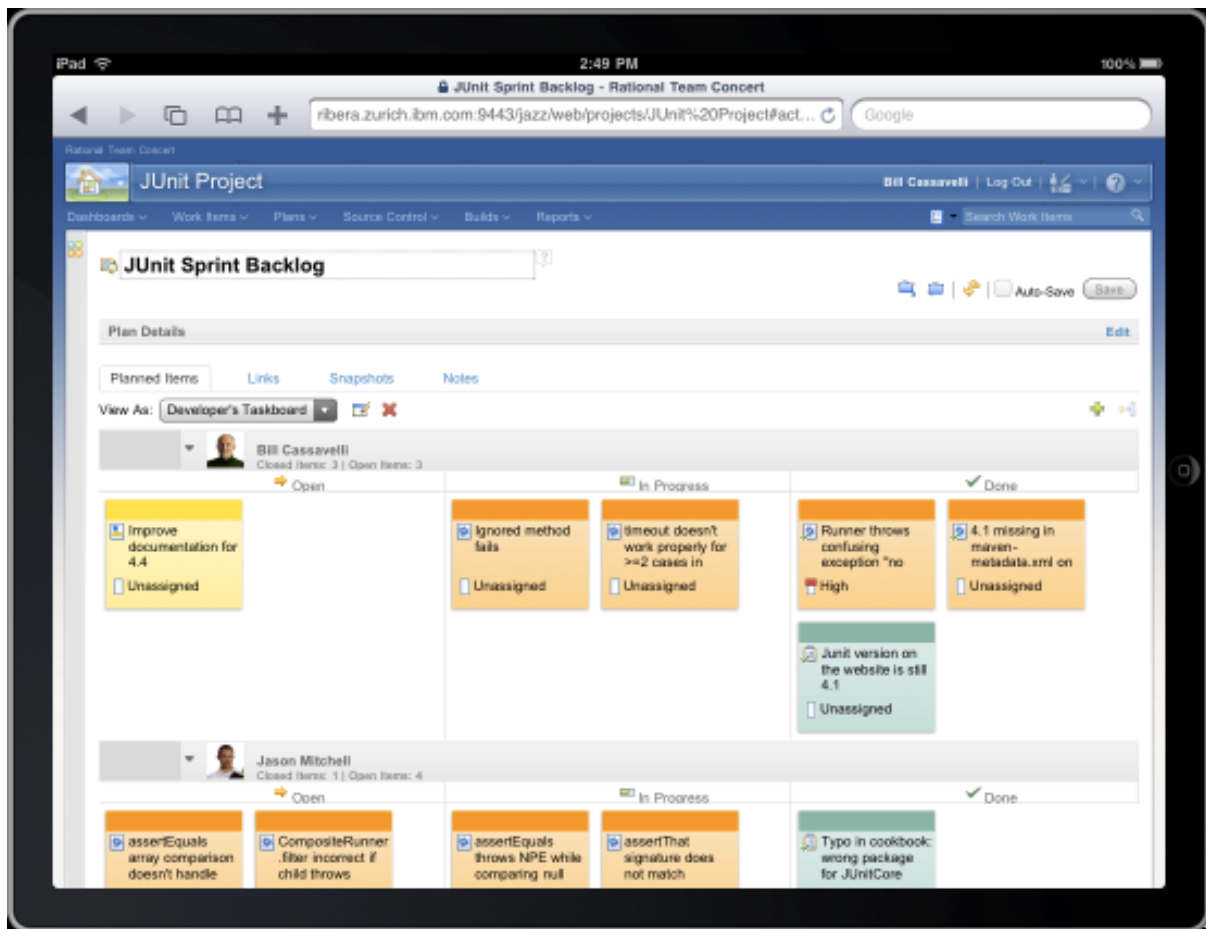


Figura 3 – Quadro de Tarefas em uma ferramenta “C/ALM” (IBM, 2011).



Figura 4 – Gráfico de *burndown* em uma ferramenta “C/ALM” (IBM, 2011).

5. CONCLUSÃO

Tendo em vista os aspectos apresentados, pode-se concluir que a aplicação de metodologias ágeis no desenvolvimento distribuído de *software* produz alguns importantes desafios.

Esses desafios podem ser superados com a ajuda de algumas ferramentas e procedimentos, favorecendo a comunicação, o trabalho em equipe e o respeito mútuo entre os indivíduos.

Uma vez que esses desafios sejam superados, a metodologia ágil torna-se viável e vantajosa, independente da distribuição geográfica dos membros da equipe de desenvolvimento de *software*.

6. REFERÊNCIAS

- BECK, et. al. **Manifesto for Agile Software Development**. 2001. Disponível em <http://www.agilemanifesto.org/>. Acesso em 29 dez. 2011.
- BROOKS, Frederick P. **No silver bullet**: essences and accidents of Software Engineering. EUA: IEEE. Computer, v. 20, n. 4, 1987. p. 10-19.
- CARMEL, E. **Global Software Teams** – Collaborating Across Borders and Time-Zones. EUA: Prentice Hall, 1999.
- FOWLER, M. **Using an Agile Software Process with Offshore Development**. 2006. Disponível em <http://www.martinfowler.com/articles/agileOffshore.html>. Acesso em: 29 dez. 2011.
- HERBSLEB, J. D., MOITRA, D. **Global Software Development**. EUA: IEEE Software, March/April 2001.
- HIGHSMITH, J. **Agile Project Management** – Creating Innovative Products. EUA: AddisonWesley, 2004.
- IBM. **Rational Team Concert**. 2011. Disponível em <http://www-01.ibm.com/software/rational/products/rtc/index.html>. Acesso em 30. dez. 2011.
- INSTITUTO NACIONAL DE CIÊNCIA E TECNOLOGIA PARA ENGENHARIA DE SOFTWARE (INES). **FireScrum**: The Open Source Scrum Tool. 2011. Disponível em <http://sourceforge.net/projects/firescrum/develop>. Acesso em: 30 dez. 2011.
- RAYMOND, E. S. **The Cathedral and the Bazaar**. EUA: O'Reilly, 1999.

ROYCE, W. **Managing the Development of Large Software Systems.** EUA: IEEE WESCON, 1970.

SUTHERLAND, J. **Fully Distributed Scrum:** The Secret Sauce for Hyperproductive Offshored Development Teams. EUA: IEEE Software, 2008.