

# APLICANDO METODOLOGIAS ÁGEIS EM PROJETOS DE INTEGRAÇÃO DE SISTEMAS

## Applying agile methodologies in systems integration projects

Demitrius Ruan Quadros<sup>1</sup>

**Resumo:** Integração de aplicações corporativas são os meios computacionais pelo qual se possibilita a integração de aplicações empresariais, de forma a suportar todos os processos integrados. Este processo é realizado das mais variadas formas, em que se possibilita que um sistema converse com o outro. Porém, organizar o desenvolvimento de integrações em meio a projetos de implantação de *software* requer controles rápidos e eficazes, que permitam aos gestores do projeto controlar de forma eficiente sua equipe, para que as integrações saiam completamente aderentes ao processo diário da empresa. A melhor maneira de controlar esses desenvolvimentos é por meio de metodologias ágeis, como o SCRUM, em conjunto a ferramentas como o KANBAN, que permitem por meio de seus processos controlar a execução dos projetos de integração e gerando uma documentação concisa sobre os desenvolvimentos, em prazos curtos, em que o mais importante é o resultado final do produto de *software* funcionando totalmente aderente ao processo.

Palavras-chave: SCRUM. Integração. Processo. Informação.

**Abstract:** Integration of enterprise applications, are the computational means by which it is possible to integrate enterprise applications, in order to support all the integrated processes. This process is performed in a variety of ways, where it is possible for one system to talk to the other, abstracting the user that they are two heterogeneous bases. But organizing the development of integrations in the midst of *software* deployment projects requires fast, effective controls that allow project managers to efficiently control their team so that integrations are completely adherent to the day-to-day business process. The best way to control these developments is through agile methodologies such as SCRUM in conjunction with tools such as KANBAN, which allow through their processes to control the execution of the integration projects and thus generating concise documentation on developments, within deadlines Short where the most important is the end result of the *software* product running fully adherent to the process.

Keywords: SCRUM. Integration. Process. Information.

## Introdução

Com o crescimento dos *softwares* de gestão dentro das organizações, que cada vez mais adquirem *softwares* de serviços e *softwares* especialistas para otimizar os seus processos, percebe-se a crescente necessidade de manter estes sistemas com as informações pertinentes integradas ao seu negócio, como podemos identificar na publicação de Marcus Garcia de Almeida, da Universidade Federal do Paraná (UFPR):

O acesso a informações relevantes é de valor inestimável para os gestores das organizações e está baseado em sistemas de informação. Para gerar estas informações é necessário combinar dados armazenados em vários sistemas que precisam integrar-se para que informações novas e relevantes sejam geradas (ALMEIDA, 2010).

Esta crescente demanda de manter dois ou mais sistemas com informações integradas obrigou a indústria de sistemas da informação a desenvolver técnicas, *softwares* e conceitos que estão diretamente ligados à integração de sistemas.

---

<sup>1</sup> Centro Universitário Leonardo Da Vinci – UNIASSSELVI – Rodovia BR 470 – KM 71 – nº 1.040 – Bairro Benedito – Caixa Postal 191 – 89130-000 – Indaial/SC Fone (47) 3281-9000 – Fax (47) 3281-9090 – E-mail: demitriusquadros@gmail.com

---

Para as empresas, ter um ambiente com integrações eficazes é de extrema importância, pois é necessário que as integrações mantenham os sistemas atualizados, com uma base de dados íntegra e que executem de forma mais performática possível, seja ela via importação de arquivos ou integrações via *software* em bases heterogêneas.

Muitas destas integrações são criadas pelas empresas que adquirem o *software*, na maioria das vezes por haver desenvolvimento interno de sistemas legados. Assim, estas integrações trafegam as informações entre os sistemas, mantendo a base de dados íntegra dos dois lados.

Utilizando conceitos de SCRUM e KANBAN, pretende-se exemplificar como aplicar metodologias ágeis de desenvolvimento de *softwares* para projetos de integração de sistemas, exemplificando sua aplicação em um *case* real, com base em projetos dos quais o autor deste artigo participou.

Muitas das integrações estão incorporadas a projetos de implantação de sistemas dentro de corporações, desta forma, para aplicar os conceitos de metodologias ágeis a integrações de sistemas devemos considerá-las um projeto à parte do principal, mas que é necessário ser executado para que as entregas do projeto em questão não sofram nenhuma interferência.

### **Integração de Aplicações Corporativas**

Integração de aplicações corporativas (EAI – do inglês *Enterprise Application Integration*), são os meios computacionais pelo qual se possibilita a integração de aplicações empresariais, de forma a suportar todos os processos de negócio de forma integrada. Este processo de integração de aplicações pode ocorrer de várias formas. Base a base por meio de outro *software* que faça a validação e a adaptação da informação, ou seja, cria-se a comunicação entre bases heterogêneas.

### **A história do EAI**

Para entendermos a história do EAI, recuamos para o início da informatização empresarial, onde as informações eram processadas pelos mainframes e os dados e informações trafegavam por ambientes homogêneos.

Seguindo pela evolução tecnológica, surgiram as plataformas pessoais com sistemas operacionais (Windows, Unix, Linux, Mac OS), o que possibilitou um novo paradigma de programação. Estes ambientes gráficos produzidos vieram futuramente a ser as aplicações distribuídas.

Como os *mainframes* não acompanharam a evolução das plataformas e não conseguiram resolver as demandas departamentais, além do seu alto custo, obrigaram muitas empresas a desenvolverem suas ferramentas proprietárias para atender ao negócio de cada área específica.

Isto fez com que as empresas adquirissem *softwares* que atenderiam todo o processo de sua empresa (*Softwares* ERP), estes *softwares* que em muitos casos não atendem em sua plenitude todo o processo de negócio de uma empresa. Desta forma, obriga-se a empresa a trabalhar com outros *softwares* mais específicos para atender em plenitude cada área de negócio, este *software* com atributos diferentes e que armazenam em sua maioria a mesma informação replicada do *software* centralizador (ERP).

Esta replicação de dados em questão é o que chamamos de EAI, e é desenvolvida justamente para realizar esta replicação de dados entre bases.

---

## Empresas que necessitam de EAI

As empresas que geralmente necessitam de EAI são, em sua maioria, corporações com processos complexos, e que para terem seus processos atendidos em sua plenitude utilizam mais de um *software* de gestão para atender ao fluxo de processos de suas áreas específicas.

Com base no artigo *Enterprise Application Integration*, de Ananias Laftsidis (2016), da IBM Sweden, pode-se observar que em uma empresa que normalmente precisa ser integrada contém a seguinte estrutura de sistemas, ou passou por estes passos tecnológicos:

- **Sistemas legados:** estes são sistemas que foram herdados em organizações com linguagens, plataformas e técnicas anteriores à tecnologia atual. A maioria deles são baseados em *mainframes*, embora alguns deles também possam ser encontrados em plataformas UNIX. Estes são sistemas que fornecem processamento centralizado em terminais.
- **Sistemas distribuídos:** são os sucessores dos sistemas legados. Estes são sistemas que foram desenvolvidos para os departamentos, enquanto buscavam um processamento mais rápido e de menor custo. Sistemas distribuídos são encontrados nas mais diversas plataformas, embora sistemas de *mainframes* modernos também os suportem. Aplicações distribuídas são capazes de fornecer interfaces gráficas com maior usabilidade e aceitação do usuário do que as telas dos terminais. Eles são baseados em tecnologias mais recentes, como orientação a objeto.
- **Pacotes de aplicativos:** ao contrário dos sistemas distribuídos que são desenvolvidos pelas empresas, pacotes de aplicações são normalmente adquiridos. Estas aplicações contêm reutilizações de negócio em seus processos, que podem ser usados em uma organização depois de algum processo de customização. Existem muitos tipos diferentes de aplicações embaladas. Um dos tipos mais comuns é o ERP. Outros tipos podem ser controle de estoque ou aplicações de automação de vendas (LAFTSIDIS, 2016).

Uma empresa que passou por todas estas transições de *software*, com certeza teve a sua informação replicada entre bases várias e várias vezes, isto faz com que alguns departamentos da empresa funcionem com um determinado *software*, e outro departamento com outro *software*, porém compartilhando a mesma base de informações.

## O que é SCRUM?

SCRUM é um *framework* de desenvolvimento iterativo incremental para desenvolvimento de projetos de *software* ágil. Utilizado em trabalhos complexos nos quais é impossível prever todos os acontecimentos.

Scrum é semelhante aos sistemas autocorretivos, evolucionários e adaptativos. Desde o começo, a estrutura do Scrum se tornou a forma de o setor tecnológico criar novas aplicações de *software* e produtos. Contudo, embora ele tenha se tornado muito bem-sucedido no gerenciamento de projetos de *software* e *hardware* no Vale do Silício, ainda permanece pouco conhecido em outros setores de negócios (SUTHERLAND, 2014, p. 5).

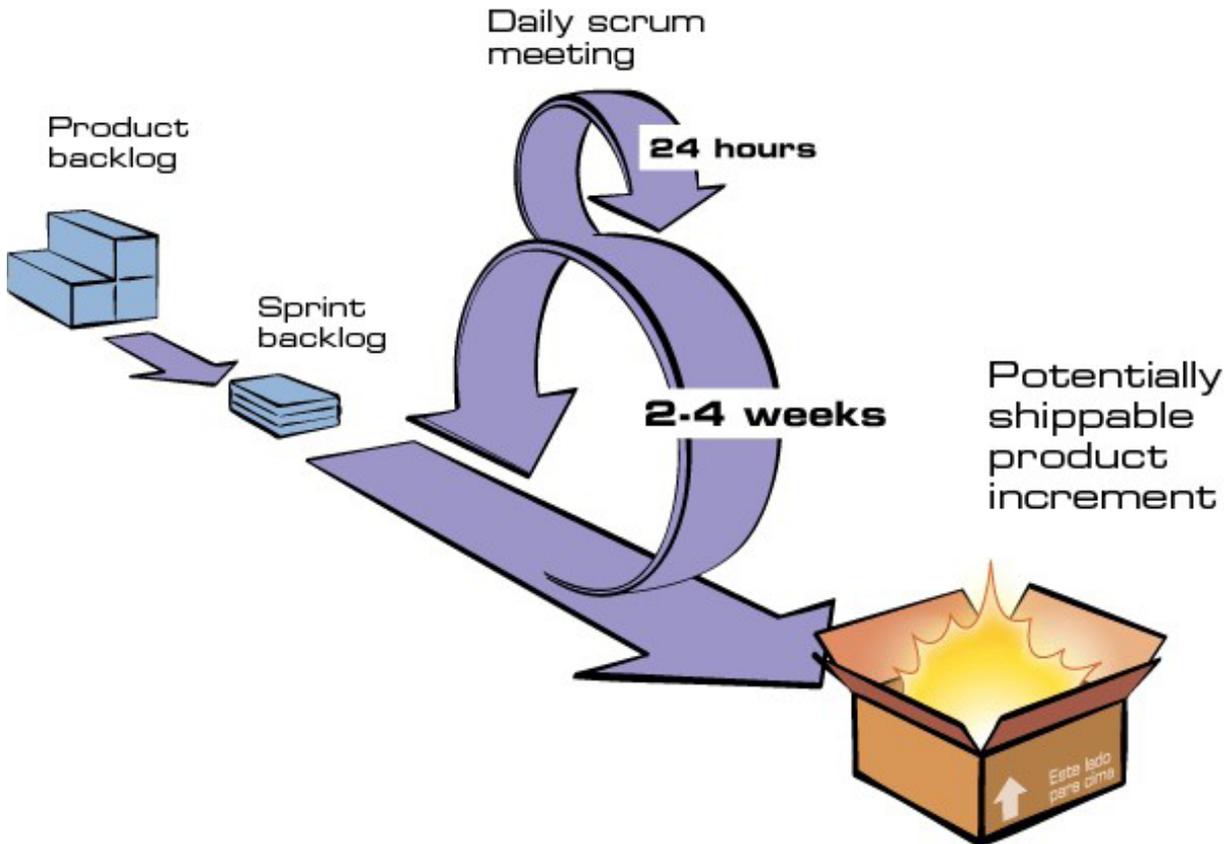
SCRUM é muito utilizado por empresas, pois facilita com que as mesmas alcancem seus objetivos de forma ágil e clara. Também é uma ótima ferramenta quando se têm prazos curtos e objetivos bem definidos.

## Artefatos do SCRUM

Os artefatos do SCRUM são os documentos e outros elementos utilizados durante a aplicação do SCRUM dentro do projeto.

O SCRUM disponibiliza as seguintes ferramentas em sua *framework*, e elas são: *Product Backlog*, *Sprints*, *Sprint Planning*, *Daily Scrum*, *Definition of Done*, *Sprint Review*, *Sprint Retrospective*, conforme representado na figura abaixo:

**Figura 1.** Ferramentas disponibilizadas pelo SCRUM



Fonte: Disponível em: <<http://www.desenvolvimentoagil.com.br/scrum/>>. Acesso em: 29 maio 2016.

Para melhor compreender cada artefato, foi desenvolvido o quadro abaixo, com breve descrição de cada um deles:

**Quadro 1.** Resumo Artefato X Descrição

Artefato	Descrição
PRODUCT BACKLOG	<i>Product backlog</i> é uma lista de todas as prioridades que devem ser desenvolvidas no projeto, ordenada por ordem de prioridade.
SPRINTS	<i>Sprint</i> é uma iteração ou ciclo de trabalho que deve ser realizado. O trabalho resultante de cada <i>Sprint</i> deve ser algo de valor para o produto e para o cliente final.
DAILY SCRUM	Reunião diária que acontece geralmente no mesmo horário, todos os dias. Com tempo definido de 15 minutos e a equipe de pé responde a perguntas definidas.

DEFINITION OF DONE	<i>Definition of Done</i> é um documento que define quando e como o resultado de um <i>Sprint</i> é dito como pronto.
SPRINT REVIEW	O <i>Sprint review</i> reúne todos os participantes do projeto, em que é verificado o produto construído, que é apresentado geralmente em forma de demonstração.
SPRINT RETROSPECTIVE	O <i>Sprint Retrospective</i> serve para identificar as adaptações necessárias no produto, e desta forma verificar o que pode ser melhorado, o que funcionou bem e o quais são as ações futuras.

Fonte: Disponível em: <<http://www.desenvolvimentoagil.com.br/scrum/>>. Acesso em: 29 maio 2016.

### Papéis e responsabilidades

Existem três papéis principais que compõem a *framework* do SCRUM, são eles: *Product Owner*, *Scrum Master* e *Development Team*.

Os papéis servem para definir quais serão os participantes do projeto, estes participantes não deverão ter contato com o mundo externo ao projeto quando trabalhando nele, e devem comunicar-se sobre o projeto entre si.

Para melhor exemplificar, pode-se ver uma breve descrição de cada papel no quadro abaixo:

**Quadro 2.** Aprendendo um pouco mais sobre o Framework Scrum

PAPEL	DEFINIÇÃO
PRODUCT OWNER	Pessoa que exerce a liderança sobre o projeto que está em desenvolvimento. Define o que precisa ou não ser executado em relação ao produto.
SCRUM MASTER	Conhece SCRUM como ninguém, em casos de dúvidas, deve orientar os membros da equipe como agir em cada situação. Em analogia aos esportes, o Scrum Master é o treinador do time.
DEVELOPMENT TEAM	A equipe de desenvolvimento faz o trabalho de desenvolver as funcionalidades do produto, incluindo a concepção, desenvolvimento, integração e testes das funcionalidades desenvolvidas.

Fonte: Adaptado de SCHWABER (2004).

### KANBAN

KANBAN é um termo japonês que significa cartão ou sinalização, em sua forma literal.

Trata-se da utilização de cartões para controlar processos predefinidos, estes cartões podem ser utilizados tanto na forma de *post-it* ou *softwares* que façam esta funcionalidade.

Para cada local onde o cartão for colocado, são utilizadas indicações que determinam em qual tarefa está determinado cartão, por exemplo: “A fazer”, “Fazendo”, “Terminado”.

Em sua utilização, o KANBAN é responsável por expor as atividades enquanto elas fluem pelo processo, desta forma é possível visualizar os gargalos do processo e identificar onde deve-se agir e colocar um maior esforço.

---

O Kanban, no entanto, vai um passo além e dá transparência ao processo e seu fluxo. O Kanban expõe gargalos, filas, variabilidade e desperdício. Tudo que impacta o desempenho da organização em termos de quantidade de trabalho de valor entregue e o tempo de ciclo necessário para entregá-lo. Proporciona aos membros da equipe e às partes interessadas externas a visibilidade sobre os efeitos de suas ações (ou falta de ações). Sendo assim, os primeiros estudos de caso estão mostrando que o Kanban muda o comportamento e incentiva uma maior colaboração no trabalho (KNIBERG; MATTIAS, 2009, p. 11).

Em suma, sua principal funcionalidade é visualizar em que estágio se encontra cada processo, e isto casa muito bem com a aplicação das técnicas de SCRUM. Pode-se tomar como exemplo a visualização do *Product Backlog*, em que se pode ver a situação de cada um via KANBAN e assim identificar em que estágio está cada item da lista.

### **Aplicando em um *case* real**

Neste passo será exemplificado como aplicar estes conceitos em um *case* real, com as técnicas utilizadas no projeto de implantação de um sistema ERP em uma empresa do ramo têxtil, identificando como foi utilizado os conceitos de SCRUM e KANBAN para organizar as integrações a serem desenvolvidas para o projeto.

### **Metodologias ágeis em projetos de integração**

O desenvolvimento de integrações em projetos, em sua grande maioria, é desenvolvido conforme a demanda, sem nenhum controle ou especificação. Como pode-se observar na citação a seguir, hoje persiste um “padrão” para o desenvolvimento de integrações dentro das mais diversas empresas.

O analista de sistemas responsável pelo novo SI analisa, especifica e gerencia o desenvolvimento das integrações requeridas. Estas são entregues, na maioria das vezes, na forma de *software* que são adicionados à nova aplicação e que estabelecem um forte vínculo com a outra ponta da integração (SORDI; MARINHO, 2007, p. 7).

Para melhorar o processo de desenvolvimento das integrações, neste projeto foi definida a utilização de algumas ferramentas de SCRUM e KANBAN para gerar uma documentação concisa dos *softwares* de integração que estavam sendo desenvolvidos.

A escolha da ferramenta de SCRUM permite que sejam feitas análises sobre os produtos entregues, o que permite reavaliar os processos seguintes conforme o projeto está em andamento, melhorando assim os passos seguintes.

O Scrum acolhe a incerteza e a criatividade. Coloca uma estrutura em volta do processo de aprendizagem, permitindo que as equipes avaliem o que já criaram e, o mais importante, de que forma o criaram. A estrutura do Scrum busca aproveitar a maneira como as equipes realmente trabalham, dando a elas as ferramentas para se auto organizar e, o mais importante, aprimorar rapidamente a velocidade e a qualidade de seu trabalho (SUTHERLAND, 2014, p. 13).

---

## Separando os pontos de integração

Primeiramente, identifica-se todos os pontos de integração no projeto que deveriam ser trabalhados, tendo em vista a quantidade de integrações a serem desenvolvidas, juntamente com o tempo a ser investido por parte das empresas para que os sistemas legados e o novo ERP se comunicassem de forma inteligente e *on-line*. Pela grande quantidade de integrações a serem realizadas, trabalhou-se a construção das integrações como um projeto à parte à implantação do *software* ERP, sendo controlado de uma forma separada, porém, sendo um ponto crítico para o sucesso do projeto principal.

Para separar os pontos de integração foi realizada uma reunião, em que todas as integrações foram levantadas e definidas como *Product Backlog's* a serem controladas no projeto.

## Utilizando KANBAN para controlar PRODUCT BACKLOG'S

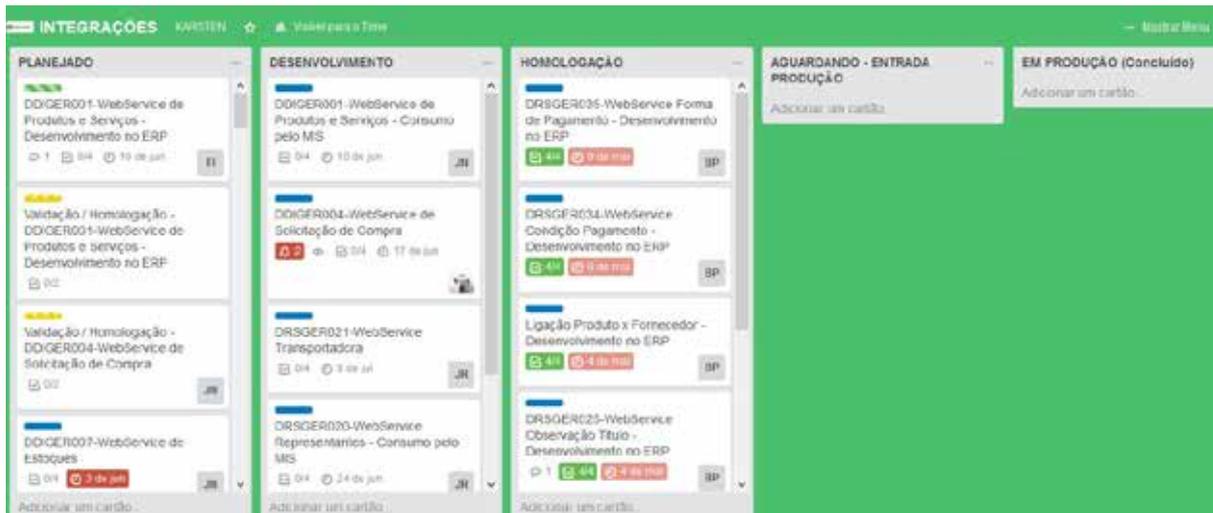
Em um primeiro momento separou-se todos os pontos de integração do projeto (*product backlog's*) e cadastrou-se como quadros na ferramenta gratuita Trello.

O Trello é uma ferramenta que permite a execução e controle de quadros KANBAN, desta forma pontuou-se todos os pontos de integração do projeto e colocou-se os mesmos dentro da ferramenta para que seja possível o controle de cada integração, identificando em que ponto cada integração se encontra dentro do projeto.

Dentro da ferramenta utilizou-se as seguintes divisões para controlar os quadros com cada ponto a ser realizado no projeto:

1. Planejado: todos os cartões dentro desse quadro ainda não foram iniciados, sendo assim, estão aguardando o término de determinado Sprint para iniciar.
2. Desenvolvimento: todos os cartões dentro desse quadro estão em desenvolvimento e dentro de um Sprint.
3. Homologação: todos os cartões dentro desse quadro estão aguardando a homologação do usuário, e serão testados e demonstrados em um Sprint review.
4. Aguardando entrada em produção: todos os cartões dentro desse quadro estão à espera de entrar em produção, dessa forma esses cartões já foram validados, porém podem ser revistos em uma reunião de Sprint Retrospective para identificar se há alguma adaptação necessária.
5. Em produção: neste ponto o cartão está finalizado, ou seja, onde o *product backlog* é entregue, e passou por todos os processos dentro do *Definition of Done* do projeto.

Figura 2. Integrações



Fonte: Disponível em: <<https://trello.com/b/t9OZFbQ0/integracoes>>. Acesso em: 29 maio 2016.

### Artefatos do SCRUM utilizados no projeto

Aqui será exemplificado como cada artefato do SCRUM foi utilizado dentro do projeto, e como foi conciliado ao controle via KANBAN.

### Product backlog's

Os *product backlog's* definidos no projeto foram criados em uma reunião, em que cada integração a ser entregue foi definida como um *product backlog* a ser entregue no projeto.

Cada *product backlog* foi documentado gerando um documento central chamado de macro fluxo de integração, onde cada *product backlog* estava descrito informando suas responsabilidades e processos dentro do funcionamento dos sistemas a serem integrados.

### Sprints

Para os *Sprints* foi definida uma periodicidade mensal, cada *Sprint* é composto por um conjunto de integrações a serem realizadas, o seu resultado é algum processo integrado funcionando completamente.

Conforme pode-se ver na citação a seguir, um *Sprint* é composto por vários *Product Backlog's*, esse conjunto de *Product Backlog's* denomina-se *Sprint Backlog*.

O *Sprint Backlog* é composto por uma lista de tarefas extraídas do *Product Backlog*, com as quais a equipe se compromete a fazer durante uma *sprint*, ou seja, a equipe determina a quantidade de itens do *Product Backlog* que serão executados, pois ela estará responsável em se comprometer por sua implementação e entrega de suas respectivas funcionalidades (SCHWABER, 2004).

Pode-se tomar como exemplo a seguinte situação: o processo de compras será controlado dentro do ERP que está sendo implantado, porém as solicitações de compra a serem cadastradas no novo ERP partiram do cadastro do sistema legado, que continuará controlando a parte produtiva da empresa.

---

Um Sprint pode ser definido com todos os *product backlog's* necessários para controlar o processo de compras da empresa, dessa forma, ao final desse Sprint o processo de compras é entregue funcionando plenamente integrado nos dois sistemas.

### **Daily SCRUM**

No Daily SCRUM utilizava-se a reunião para discutir o andamento das integrações, em que se definia o que havia sido feito, quais os problemas encontrados e quais os pontos podiam ser melhorados. Dentro do Daily SCRUM também se utilizava o tempo para atualização dos quadros KABAN, tendo assim uma visão diária da posição do projeto.

### **Definition of done**

Para o documento de definição, adaptou-se a sua utilização e criou-se um novo documento, que foi utilizado para mais opções.

O documento *Definition of Done* define os passos para obter um resultado tangível ao fim de um *Sprint*. Porém, para a utilização, decidiu-se criar um documento para descrever o processo para desenvolvimento de cada integração, o que não representa exatamente a conclusão de um *Sprint*, pois cada *Sprint* tem como resultado um processo funcionando e é composto pelo desenvolvimento de várias integrações.

Este documento de desenvolvimento foi denominado de MDS (Metodologia de Desenvolvimento de *Software*), ele contém todos os passos para desenvolver determinada integração, passando por todas as etapas de desenvolvimento de *software*, como diagrama de classe, caso de uso, fluxograma de processo e suas descrições.

Pode-se dizer, então que, na realidade, o documento *Definition of Done* era um conjunto de documentos que determinava como cada integração deveria funcionar para cada *Sprint*.

### **Sprint Review**

Para o *Sprint Review* ao término de cada *Sprint* reunia-se todos os envolvidos no resultado deste *Sprint* para realizar os testes práticos dos sistemas funcionando em conjunto.

Exemplificando, junta-se usuários que irão realizar o processo, consultores de implantação do sistema ERP e a equipe de TI da empresa para testar as rotinas que foram desenvolvidas, nesta etapa executa-se o processo em base de teste dos sistemas para que se possa visualizar o sistema funcionando com as integrações e deixar de forma clara como funciona cada processo após a implantação do sistema.

Também, no *Sprint Review*, identifica-se os pontos de ajuste das rotinas de integração, se necessário realiza-se algum ajuste que demande alguma alteração processual, este é identificado no momento dos testes integrados da reunião *Sprint Review*.

### **Sprint retrospective**

No *Sprint Retrospective* analisa-se juntamente com a equipe de desenvolvimento e da consultoria de implantação do sistema ERP o resultado das integrações. Nesta reunião a equipe tenta identificar o que pode ser melhorado em cada aspecto da integração, se existe uma forma melhor de realizar este processo, além de identificar a real funcionalidade desta integração dentro do projeto, se a performance da rotina é adequada e se os dados que são replicados entre os sistemas estão adequados.

---

## Por que não utilizar somente uma técnica?

Utilizar o SCRUM, para a equipe do projeto, não era o suficiente, pois existem vários riscos ao implantar um *software* ERP em empresas com sistemas legados. Um desses problemas era perder membros da equipe de SCRUM para problemas do cotidiano, pois os mesmos interferem diretamente nos resultados dos *Sprints*, e como de costume, *Sprints* tem interações curtas (de uma semana a um mês) e um ou dois dias perdidos no desenvolvimento das integrações interferem diretamente no resultado final do *Sprint*.

Com base em um desses problemas, surgem dúvidas: como controlar o processo de SCRUM e saber em que estado está cada *Product Backlog* após aplicar as suas ferramentas?

Esta resposta é com o KANBAN, ele permite que seja realizada a visualização imediata da posição de determinado cartão em meio a um processo. Utilizando o KANBAN em conjunto ao SCRUM pode-se visualizar diretamente em que ponto está cada integração e quais serão os gargalos para a entrega de determinado *Sprint*, ou até para a entrega do projeto como um todo.

Como pode-se observar na citação abaixo, o KANBAN é capaz de auxiliar rapidamente uma equipe que está com um gargalo.

Três meses depois de introduzir o Kanban, a equipe do sistema de administração ganhou da Direção o título de “equipe com a melhor performance” no departamento de TI. Ao mesmo tempo, a equipe foi votada como umas das 3 melhores experiências na retrospectiva da companhia. A “retrospectiva da companhia” é um evento de toda a empresa que acontece a cada 6 semanas, e esta foi a primeira vez que uma equipe entrou na lista das 3 melhores! E, apenas 3 meses antes, essa equipe havia recebido diversos tipos de reclamações devido a pontos de gargalo (KNIBERG et al., 2009, p. 117).

Por motivos como este optou-se por uma mistura de utilização das duas técnicas, permitindo que o processo seja mais flexível e aderente a um projeto de implantação de um *software* ERP.

## Metodologia

Para levantar o conteúdo deste trabalho foram realizadas pesquisas em livros e *sites* na internet sobre desenvolvimento ágil, EAI, SCRUM e KANBAN. Foram levantados os dados sobre origem do assunto e como aplicá-los.

Referente à aplicação do assunto em um *case* real, foi feita a coleta dos dados enquanto o autor participava de um projeto real, assim podia visualizar a aplicação do assunto em meio a um ambiente real.

## Resultados e discussão

Os resultados obtidos, aplicando as metodologias aqui apresentadas em um *case* real, foram:

1. Maior agilidade no desenvolvimento das integrações, evitando o retrabalho. Como a utilização do SCRUM permite que os processos sejam revistos a partir do *Sprint Review* e do *Daily Scrum*, evita-se que os erros cometidos no desenvolvimento do primeiro *Sprint* sejam repetidos no segundo, e assim sucessivamente, através dos *Sprints* que compõem o projeto.

- 
2. Criação de uma documentação padrão para as integrações. Com a utilização de uma metodologia define-se vários padrões que irão documentar os produtos do projeto, gerando uma documentação que ficará armazenada e de posse da empresa, não permitindo que o conhecimento seja perdido com a saída ou substituição de alguém dentro da equipe.
  3. Possibilidade de visualizar o andamento do projeto de integrações. Utilizando o KANBAN permite-se uma visualização imediata do ponto em que o projeto está, é possível também utilizar ferramentas, como gráficos, para medir o andamento do projeto com base na posição dos cartões e suas respectivas datas de entrega.
  4. Possibilidade de agir em problemas encontrados antecipadamente. Por meio das ferramentas disponibilizadas é necessária uma validação antecipada do produto, dessa forma é possível visualizar erros antes da entrada do produto em ambiente de produção, antecipando problemas que a empresa teria após o *go-live* do projeto principal.
  5. Padronização do desenvolvimento das integrações. Com a criação de novos documentos, e adaptando algumas das ferramentas do SCRUM, foi possível definir um processo padrão para o desenvolvimento de integrações, com métricas definidas, fazendo com que toda integração desenvolvida fosse padronizada.
  6. Maior comunicação da equipe. Com as reuniões diárias a equipe passou a se comunicar de forma mais ativa e de forma mais criativa, fazendo com que os laços no relacionamento de trabalho se estreitassem mais, quanto mais a equipe tinha maturidade para discutir o projeto, maior ficava a qualidade dos produtos finais.
  7. Todos os membros da equipe tinham total conhecimento sobre os temas. Como todos os membros da equipe participavam ativamente de todas as definições de processo, todos tinham conhecimento do que era necessário para a continuidade do projeto, não deixando muitos espaços de dúvidas a serem preenchidos na hora de aplicar o que havia sido planejado.

Como pode-se analisar, aplicar metodologias ágeis em projetos de integração de sistema é uma das melhores formas de controlar os objetivos, pois é de fácil visualização o progresso do trabalho.

Como os envolvidos no trabalho já estavam familiarizados com integrações de sistemas, esta foi uma parte de fácil controle no projeto, porém ocorreram problemas, como falta de equipe em momentos cruciais do projeto por parte da empresa que estava adquirindo o *software* ERP, pois em vários momentos a equipe era redirecionada para suportar os processos do sistema legado, impactando diretamente na entrega final do projeto.

### **Considerações finais**

Após pesquisar e analisar sobre o assunto, conclui-se que a melhor forma de lidar com integrações de sistemas em projetos de implantação de *softwares* é por meio de metodologias ágeis.

Antes de implantar um padrão para o projeto, as integrações de sistema no qual o autor participou não eram controladas, de forma que não havia uma documentação sobre o processo, ficando o conhecimento somente na mão de pessoas que participaram do projeto e futuramente, com o passar do tempo, este conhecimento se perdia com a saída de pessoas chaves ao projeto, perdendo uma referência sobre o processo, que por vez é fundamental à empresa.

---

Como *softwares* de integração de sistemas são *softwares* pequenos, que por sua vez possuem uma funcionalidade específica, não há necessidade de aplicar um processo complexo para seu desenvolvimento, porém há a necessidade de se aplicar algum processo para gerar uma documentação sobre o ocorrido, para que se possa, no futuro, visualizar o processo com mais clareza e permitir com que os processos saiam com maior assertividade após a conclusão do projeto.

## Referências

ALMEIDA, Marcus Garcia de. **Integração de sistemas de informação: uma proposta metodológica.** Curitiba: UFPR, 2010. Disponível em: <<http://www.ppcgi.ufpr.br/publicacoes/12-integracao-de-sistemas-de-informacao-uma-proposta-metodologica.html>>. Acesso em: 20 mar. 2016.

FERREIRA, Manuela Klanovicz et al. **Um modelo de integração entre sistemas de informação na Universidade Federal do Rio Grande do Sul: eventos e repositório digital.** [s.d.]. Disponível em: <<https://www.lume.ufrgs.br/bitstream/handle/10183/55392/000858805.pdf?sequence=1>>. Acesso em: 9 jun. 2016.

GUTIERREZ, Felipe. **Para especialistas, maior risco de computação em nuvem é erro humano.** 2015. Disponível em: <<http://www1.folha.uol.com.br/mercado/2015/07/1657359-para-especialistas-maior-risco-de-computacao-em-nuvem-e-erro-humano.shtml>>. Acesso em: 20 nov. 2015.

KNIBERG, Henrik; MATTIAS, Skarin. **Kanban e Scrum: obtendo o melhor de ambos.** United States: C4Media Inc, 2009.

LAFTSIDIS, Ananias. **Enterprise Application Integration.** 2016. Disponível em: <<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.200.6603&rep=rep1&type=pdf>>. Acesso em: 9 jun. 2016.

SCHWABER, Ken. **Agile Project Management with Scrum.** United States: Microsoft Press, 2004.

SORDI, José Osvaldo de; JÚNIOR, Gildo Medeiros. **Abordagem sistêmica para integração entre sistemas de informação e sua importância à gestão da operação: análise do caso GVT.** 2006. Disponível em: <<http://www.scielo.br/pdf/gp/v13n1/29580.pdf>>. Acesso em: 20 mar. 2016.

SORDI, José Osvaldo de; MARINHO, Bernadete de Lourdes. **Análise dos ambientes para integração entre sistemas de informação segundo especialistas.** 2006. Disponível em: <<https://periodicos.ufsc.br/index.php/adm/article/view/1678>>. Acesso em: 20 mar. 2016.

SUTHERLAND, Jeff. **Scrum: a arte de fazer o dobro do trabalho na metade do tempo.** São Paulo: LeYa, 2014.

---

VIEIRA, Denisson. **Scrum**: A Metodologia Ágil Explicada de forma Definitiva. 2014. Disponível em: <<http://www.mindmaster.com.br/scrum/>>. Acesso em: 9 jun.2016.

---

Artigo recebido em 30/05/17. Aceito em 10/07/17.