Disponível em
http://www.anpad.org.br/tac

TAC, Rio de Janeiro, v. 4, n. 1, art. 4,
pp. 46-58, Jan./Jun. 2014



# Projeto de *Software* no Setor Bancário: *Scrum* ou Modelo V

## A Banking-Sector Software Project: Scrum or V-Model

**Marco Alexandre Terlizzi**
E-mail: materlizzi@outlook.com
Universidade Nove de Julho - Uninove
Uninove, Av. Dr. Adolpho Pinto, 109, 01156-050, São Paulo, SP, Brasil.

**César Augusto Biancolino**
E-mail: biancolino@gmail.com
Universidade Nove de Julho - Uninove
Uninove, Av. Dr. Adolpho Pinto, 109, 01156-050, São Paulo, SP, Brasil.

**Artigo recebido em 31.10.2013. Aprovado em 27.02.2014.**

## Resumo

O desenvolvimento de *software* para o setor bancário é um processo cada vez mais complexo que exige uma metodologia de desenvolvimento de *software* estável e previsível. Considerando-se o *Scrum* e o Modelo V, o objetivo deste estudo é analisar qual destas duas metodologias seria a mais adequada para o desenvolvimento de *software* em um projeto implantado em maio/2012 no setor bancário. Enquanto o *Scrum* é uma metodologia ágil que aplica uma abordagem incremental e iterativa para melhorar a previsibilidade, o Modelo V é um processo prescritivo e tradicional que fornece um roteiro com atividades preestabelecidas. Trata-se de uma pesquisa exploratória que utiliza a pesquisa bibliográfica como estratégia, a pesquisa documental e entrevistas como métodos de coleta de dados, bem como um dos maiores bancos brasileiros como unidade de análise. Os resultados demonstram que as características do projeto atenderam aos critérios definidos neste estudo para o Modelo V e não atenderam aos critérios definidos para o *Scrum*. Ao final, propõe-se um *check-list* com cinco questões que podem ajudar outras equipes ao decidir entre o *Scrum* ou o Modelo V.

**Palavras-chave**: metodologia ágil *scrum*; modelo V; metodologia de desenvolvimento de *software*.

## Abstract

Banking sector software development is an increasingly complex process that requires a stable and predictable methodology. Considering the Scrum and V-Model, this study's objective is to analyze which of these two methodologies would be most suitable for software development in a banking project implemented in May/2012 in Brazil. While Scrum is an agile methodology that applies an incremental and iterative approach to improve predictability, the V-Model is a prescriptive and traditional process that provides a roadmap with predetermined activities. This is an exploratory research where data collection was done through documentary research and interviews. The unit of analysis is one of Brazil's biggest banks. The results shows that project characteristics met the criteria defined in this study for the V-Model and did not meet the criteria defined for Scrum. The paper includes a five-question checklist in order to help other teams decide between using Scrum or V-Model in their applications.

**Key words**: agile scrum methodology; V-model; software development methodology.

 

## Introdução

O sistema bancário brasileiro é um dos mais desenvolvidos do mundo tecnologicamente. Na comparação com a penetração de caixas eletrônicos na sociedade, por exemplo, observa-se que o índice brasileiro está muito mais próximo aos Estados Unidos, Japão e Europa do que de países como Índia e México. Nos últimos quatro anos, os investimentos dos bancos em tecnologia cresceram de R$14 bilhões, em 2009, para R$20,1 bilhões, em 2012 (crescimento de 43,5%), e foram distribuídos da seguinte forma: R$8 bilhões em *hardware*, R$7,5 bilhões em *software*, R$4,2 bilhões em telecomunicações e R$400 milhões em outras tecnologias (Federação Brasileira de Bancos [Febraban], 2012a, 2012b).

O desenvolvimento de *software* para o setor bancário é um processo cada vez mais complexo, pois o processo de integração entre distintas plataformas com diferentes canais de comunicação demanda um processo cada vez mais robusto, consistente e previsível. Para acompanhar esta evolução tecnológica, o setor bancário brasileiro necessita construir sistemas com alto desempenho, disponibilidade e segurança, para tanto, utilizando-se de uma metodologia de desenvolvimento de *software* adequada.

Hoje é amplamente entendido que a utilização de uma adequada metodologia de desenvolvimento de *software* (MDS) melhora a eficiência e eficácia do produto final. Ao longo dos últimos 25 anos, muitas metodologias foram propostas (Cascata, Modelo V, Espiral, *Scrum, XP*, etc.), no entanto cada projeto tem uma característica única e isso significa que uma MDS pode ser adequada para um projeto e não ser adequada para outro. Escolher a MDS errada pode atrasar o cronograma do projeto e gerar retrabalho desnecessário. Portanto, muitas vezes, a equipe do projeto enfrenta a difícil tarefa de selecionar a MDS mais adequada (Guntamukkala, Wen, & Tarn, 2006).

A empresa objeto de estudo será chamada de Banco Beta, pois não foi possível coletar a autorização necessária para divulgar seu nome. A área de tecnologia do Banco Beta possui uma Área de Metodologia e Qualidade de *Software* que homologou e adotou como padrão de desenvolvimento de *software* somente as metodologias *Scrum* e Modelo V, tanto para as equipes de desenvolvimento internas quanto externas (consultorias contratadas). O projeto estudado utilizou o Modelo V aplicado de forma incremental e resultou em um sistema computacional que foi implantado em maio de 2012 e, em menos de um ano, gerou uma receita líquida de R$146 milhões para o Banco Beta. A equipe do projeto, ao invés de utilizar critérios assertivos para decidir qual metodologia utilizar, simplesmente, considerou o fato de não conhecer o *Scrum*.

O objetivo da pesquisa é analisar qual das metodologias seria a mais adequada para o desenvolvimento de *software* em um projeto implantado no setor bancário. Desse modo, este documento contém uma breve introdução, metodologias de desenvolvimento de *software*, método utilizado, caracterização da organização e projeto, mecanismos adotados na solução do problema e conclusões.

## Metodologias de Desenvolvimento de *Software*

Não existe um critério empírico estabelecido a fim de selecionar a MDS mais adequada para conduzir um projeto ao sucesso. As equipes de projeto preferem as metodologias Cascata e Modelo V quando: (a) os requisitos são claros; e (b) é esperada manutenção futura frequente do *software*. Por sua vez, as equipes de projeto preferem as metodologias *Scrum* e *XP* quando: (a) os requisitos mudam constantemente; (b) a equipe é pequena; (c) os riscos são desconhecidos; e (d) o prazo é restrito (Guntamukkala *et al.*, 2006).

A escolha de qual MDS utilizar deve ocorrer logo na fase de planejamento do projeto, pois tal decisão pode impactar o custo, cronograma e alocação de recursos. Na Tabela 1, encontra-se uma breve descrição de algumas das principais metodologias de desenvolvimento de *software*.

Tabela 1

**Algumas das Principais Metodologias de Desenvolvimento de *Software***

| Metodologia | Características | Autor(es) |
|---|---|---|
| Codificar e Consertar (*Code and fix*) | No final dos anos de 1960, o desenvolvimento de *software* era considerado uma arte e não existia um processo formalmente estabelecido. Neste modelo, os programadores simplesmente codificavam com base em conversas com usuários e depois se reuniam para testar e consertar os erros em conjunto (Guntamukkala *et al.*, 2006). | Não há |
| Cascata (*Waterfall*) | Nos anos de 1970, com a necessidade de desenvolver grandes sistemas computacionais, o autor propôs a criação de um processo organizado sequencialmente com diversas fases bem definidas, são elas: Requisitos do Sistema, Requisitos do *Software*, Análise, Desenho do Programa, Codificação, Testes e Operação. O autor nunca chamou seu modelo de cascata, este nome foi popularizado posteriormente devido ao formato de cascata de seu diagrama, onde a fase seguinte do processo somente pode ser executada após o término da fase anterior (Royce, 1970). | Dr. Winston W. Royce |
| Modelo V (V-Model) | Este modelo é uma variação do modelo cascata e foi originalmente desenvolvido pelas Forças Armadas da Alemanha, em 1979. Seu objetivo principal é relacionar as atividades de desenvolvimento de *software* com as atividades de garantia da qualidade (Bucanac, 1999). | Forças Armadas da Alemanha |
| Espiral (*Spiral*) | Processo de desenvolvimento de *software* orientado a risco criado em 1988. Seu autor propõe uma espiral onde cada ciclo da espiral representa uma fase distinta do processo. Dessa forma, o ciclo mais interno está preocupado com a Viabilidade do Sistema, o ciclo seguinte com a Definição dos Requisitos, o ciclo seguinte com o Desenho do Sistema e assim por diante. A cada novo ciclo, as atividades de gerenciamento de risco são revisitadas (Boehm, 1988). | Barry W. Boehm |
| Scrum | Concebido inicialmente em 1990, por Jeff Sutherland, evoluiu posteriormente com a contribuição de Ken Schwaber e Mike Beedle. Os princípios do *Scrum* são consistentes com o manifesto ágil e usados para orientar as atividades de desenvolvimento em um processo iterativo, onde cada ciclo do processo é chamado de *Sprint*. Enfatiza o uso de um conjunto de padrões de processos de *software* que provou ser eficaz para projetos com prazos de entrega apertados e requisitos críticos de negócio (Pressman, 2011). | Jeff Sutherland, Ken Schwaber e Mike Beedle |
| RUP – Rational Unified Process (Processo Unificado da Rational) | Modelo proprietário da empresa *Rational*, adquirida pela IBM (International Business Machines Corporation), que foi criado originalmente em 1996, por Ivar Jacobson, e vem sendo continuamente aperfeiçoado ao longo do tempo, por outros autores. É um modelo iterativo com quatro fases distintas: Iniciação, Elaboração, Construção e Transição. No entanto, ao contrário do modelo em cascata, onde as fases estão diretamente relacionadas com a parte técnica, no RUP, as fases estão estreitamente relacionadas aos negócios (Sommerville, 2011). | Ivar Jacobson, Grady Booch, James Rumbaugh e Philippe Kruchten |
| XP – Extreme Programming (Programação Extrema) | O trabalho seminal sobre o tema foi escrito em 1999. Emprega uma abordagem orientada a objetos como seu paradigma de desenvolvimento preferido e envolve um conjunto de regras e práticas constantes no contexto de quatro atividades metodológicas iterativas: Planejamento, Projeto, Codificação e Testes. O autor define um conjunto de cinco valores que estabelece as bases para todo trabalho realizado como parte da XP: Comunicação, Simplicidade, *Feedback*, Coragem e Respeito (Pressman, 2011). | Kent Beck |

**Nota.** Fonte: Elaborado pelos autores.

Como o Banco Beta adotou somente as metodologias *Scrum* e Modelo V, para entender suas características com maior profundidade, essas serão as duas metodologias estudadas como fundamentos teóricos.

## Scrum

Com o objetivo de encontrar melhores maneiras de desenvolver *software*, dezessete renomados desenvolvedores assinaram, em 2001, o Manifesto para o Desenvolvimento Ágil de *Software*. Este documento contempla um conjunto de valores e princípios em que se valorizam mais os indivíduos e as interações do que os processos e as ferramentas, mais os *softwares* funcionais do que a documentação, mais a colaboração com o cliente do que a negociação de um contrato e mais as respostas às mudanças do que um plano formal (Beck, *et al.*, 2001).

O *Scrum* é fundamentado no empirismo, o qual define que o conhecimento vem das experiências e tomadas de decisões baseadas no que é conhecido. O empirismo fundamenta-se em três pilares: transparência, inspeção e adaptação. A transparência requer que os principais aspectos do processo estejam visíveis e padronizados para que os interessados tenham o mesmo entendimento, tal como a definição de uma linguagem comum e o que se entende como algo concluído. A inspeção deve ocorrer frequentemente para detectar indesejáveis variações e precisa ser realizada, de preferência, por um especialista. A adaptação significa que o processo ou produto tem de ser adaptado quando o especialista identificar um desvio ou produto inaceitável (Schwaber & Sutherland, 2011).

O *Scrum* é considerado uma metodologia ágil e é um *framework* estrutural dentro do qual é possível empregar vários processos ou técnicas. O *Scrum* aplica uma abordagem incremental e iterativa para melhorar a previsibilidade e o controle de risco (Schwaber & Sutherland, 2011).

As equipes do *Scrum* são compostas pelo Dono do Produto, pelo *Scrum Master* e pelo Time de Desenvolvimento. O Dono do Produto é a única pessoa responsável pelo gerenciamento do *backlog* do produto, o *Scrum Master* é o responsável por garantir que o *Scrum* seja aplicado corretamente e o Time de Desenvolvimento é uma equipe disciplinar responsável por realizar a entrega com tamanho ideal de três a nove integrantes (Schwaber & Sutherland, 2011).

O *Scrum* usa eventos de tempos predefinidos (*time-box*) para criar regularidade e evitar perda de tempo. O coração do *Scrum* é o *Sprint*, um *time-box* de duas a quatro semanas em que uma entrega concluída e utilizável é criada. O *Sprint* consiste de cinco eventos: Planejamento do *Sprint*, *Daily Meeting*, Execução do *Sprint*, Revisão do *Sprint* e Retrospectiva do *Sprint*.

No Planejamento do *Sprint*, que dura aproximadamente oito horas, o Dono do Produto apresenta os itens do *backlog* do produto priorizados de acordo com os objetivos organizacionais. É importante destacar que o *backlog* do produto nunca está pronto e os primeiros desenvolvimentos apenas estabelecem os requisitos inicialmente entendidos e melhor entendidos. Em seguida, o Time de Desenvolvimento estima o escopo que poderá ser desenvolvido no *Sprint* e como o trabalho será executado. Os itens do *backlog* selecionados, em conjunto com o plano para desenvolvê-los, são chamados de *Sprint Backlog*. O *Daily Meeting* é uma reunião diária de 15 minutos para inspecionar a evolução das atividades e criar um plano para as próximas 24 horas. A Figura 1 ilustra resumidamente a visão geral do *framework Scrum* (Schwaber & Sutherland, 2011).


**Figura 1.** Visão Geral do *Framework Scrum*.
Fonte: Adaptado de Schwaber, K., & Sutherland, J. (2011). *The scrum guide.* Recuperado de https://www.scrum.org/Portals/0/Documents/Scrum%20Guides/2013/Scrum-Guide.pdf#zoom=100

Na Execução do *Sprint*, o time de desenvolvimento analisa, codifica e testa os itens selecionados. Na Revisão do *Sprint*, as equipes do *Scrum* e os *stakeholders* se reúnem para verificar o que foi e o que não foi entregue, os desafios enfrentados, as lições aprendidas, bem como redefinir o *backlog* do produto a partir dos itens antigos e dos novos que venham a surgir. A Retrospectiva do *Sprint* é uma reunião feita pela equipe para inspecionar o último *Sprint* e propor adaptações para buscar melhorias no próximo *Sprint* (Schwaber & Sutherland, 2011).

Equipes ágeis funcionam melhor quando alocadas no mesmo espaço físico, pois, além de facilitar a comunicação e colaboração, provou-se ser um meio eficaz de aumentar a produtividade da equipe (Lalsing, Kishnah, & Pudaruth, 2012).

## Modelo V

O Modelo V é um processo prescritivo e tradicional que fornece um roteiro de atividades preestabelecidas razoavelmente eficaz para as equipes de desenvolvimento de *software*. É um método prescritivo porque prescreve um conjunto de artefatos, atividades encadeadas logicamente e mecanismos de controle de mudanças que objetivam garantir a qualidade esperada do produto final. É um método tradicional porque é uma variação do modelo cascata, também conhecido como ciclo de vida clássico, e sugere uma abordagem sequencial e sistemática do início ao fim do projeto de desenvolvimento de *software* (Pressman, 2011).

O Modelo V, representado na Figura 2, foi originalmente desenvolvido pelas Forças Armadas da Alemanha. Seu objetivo principal é relacionar as atividades de desenvolvimento de *software* com as atividades de garantia da qualidade (Bucanac, 1999).

Este tipo de modelo funciona bem quando os requisitos do sistema são bem-desenhados, documentados e compreendidos por todos os envolvidos no projeto. Por outro lado, pode gerar interrupções no processo, pois a fase posterior somente pode ser iniciada após o término da anterior (Pressman, 2011).

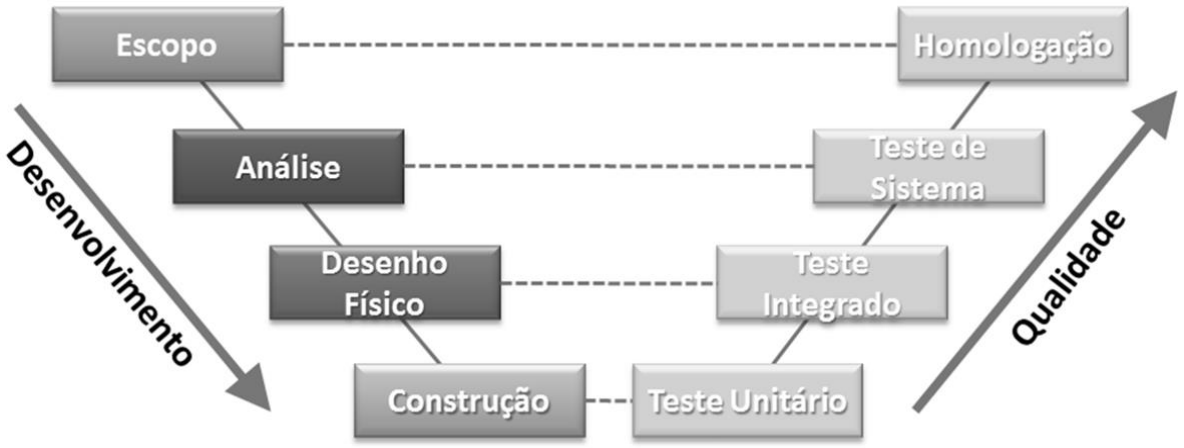


**Figura 2.** Relacionamento entre Atividades no Modelo V.
Fonte: Adaptado de Bucanac, C. (1999). *The V-Model*. Recuperado de http://www.bucanac.com/documents/The_V-Model.pdf

A dinâmica atual dos negócios bancários exige que o desenvolvimento de *software* seja cada vez mais rápido, eficaz e eficiente. Por isso, cabe discutir aqui uma das variações do Modelo V e que consiste em dividir o escopo total do projeto em funcionalidades menores, gerenciáveis e controláveis. O encadeamento dessas funcionalidades permite realizar entregas parciais do *software* de forma incremental, conforme apresentado na Figura 3.

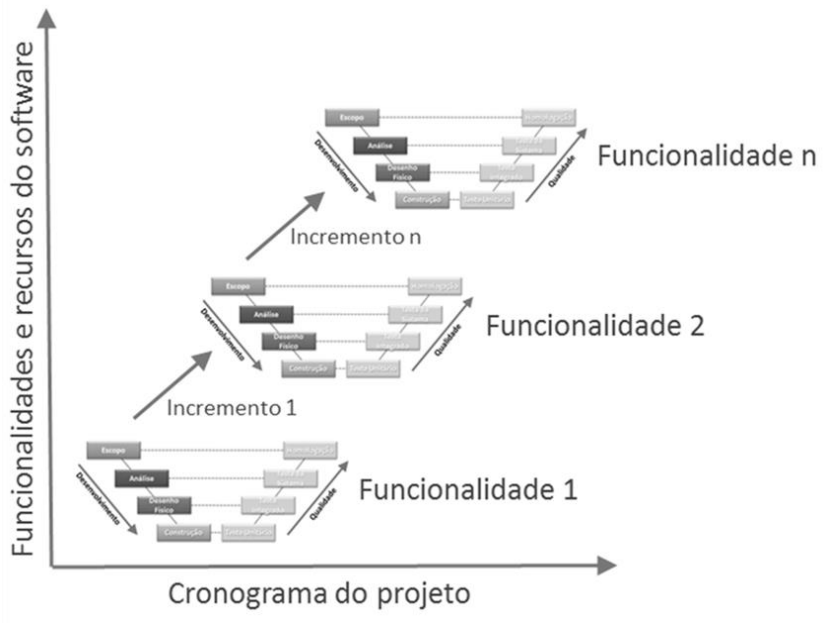


**Figura 3.** Modelo V Aplicado de Forma Incremental.
Fonte: Adaptado de Pressman, R. S. (2011). *Engenharia de Software: uma abordagem profissional* (7a ed.). Porto Alegre: AMGH.

## Método da Produção Técnica

Este estudo é uma abordagem exploratória que utilizou a pesquisa bibliográfica como estratégia de pesquisa. A unidade de análise é um dos maiores bancos brasileiros, o que permitiu um entendimento razoável sobre o segmento de mercado; as técnicas de coleta de dados utilizadas foram a pesquisa documental e entrevistas não estruturadas com três colaboradores da equipe do projeto.

A pesquisa exploratória tem como objetivo aprofundar o conhecimento sobre o problema a fim de torná-lo evidente (Gil, 2010).

A pesquisa bibliográfica utiliza, comumente, fontes secundárias de dados e é fundamental para realizar pesquisas científicas de qualquer natureza, pois busca conhecer, analisar e discutir um assunto ou problema a partir de um referencial teórico (Martins & Theóphilo, 2009). Portanto, como material de consulta, foram utilizados livros e artigos científicos recentes sobre metodologias de desenvolvimento de *software* para entender suas características e uso.

A pesquisa documental utiliza fontes primárias de dados, tais como documentos e bases de dados de entidades privadas, com o objetivo de contribuir para a análise dos problemas (Martins & Theóphilo, 2009). A coleta de dados foi realizada a partir de documentos internos do projeto e baseada na experiência profissional do pesquisador, que atuou desde o início até a conclusão do projeto na função de gerente e interagiu diretamente com todas as partes interessadas (patrocinador, executivo de conta, escritório de projetos, analistas de negócio, gestores funcionais, analistas de sistema e usuários).

Martins e Theóphilo (2009) assim explicam as entrevistas não estruturadas como técnica de coleta de dados:

> trata-se de uma técnica de pesquisa para coleta de informações, dados e evidências cujo objetivo básico é entender e compreender o significado que entrevistados atribuem a questões e situações, em contextos que não foram estruturados anteriormente. Na condução de uma entrevista não estruturada o entrevistador busca obter informações, dados, opiniões e evidências por meio de uma conversação livre. (p. 88)

Foram entrevistados três participantes da equipe do projeto para corroborar e complementar o entendimento do pesquisador sobre a situação vivenciada durante o período de planejamento do projeto estudado.

Foi utilizado o seguinte roteiro para a construção deste trabalho: (a) elaboração da questão de pesquisa com base em um problema enfrentado pelo autor na organização em estudo; (b) descrição das metodologias de desenvolvimento de *software* que suportam o estudo, principalmente por meio de pesquisa bibliográfica; (c) coleta de dados, informações e evidências, dos principais artefatos elaborados durante todo o ciclo de vida do projeto: plano de projeto, estrutura analítica do projeto, requisitos, lista de riscos, protótipos, roteiros de testes, evidências de testes, diário de bordo, relatórios de *status* e lições aprendidas; (d) entrevistas não estruturadas com três participantes da equipe do projeto; (e) análise dos dados coletados, em conjunto com as metodologias de desenvolvimento de *software*, com o objetivo de solucionar o problema de pesquisa; (f) conclusão do trabalho.

## Contexto do Projeto

O projeto de desenvolvimento de *software* aqui relatado ocorreu numa empresa do setor bancário, uma das líderes em seu ramo de atuação.

## Caracterização da organização

O estudo foi realizado em uma empresa privada, multinacional de capital nacional, atuante no setor bancário de varejo e atacado desde 1944, que possui atualmente 97 mil colaboradores, valor de mercado estimado em US$78 bilhões e conta com 5 mil agências e PABs (Postos de Atendimento Bancário) no Brasil e exterior.

A área de tecnologia do Banco Beta possui 6 mil colaboradores, um Escritório de Projetos de Tecnologia implantado e uma Área de Metodologia e Qualidade de *Software* que determina os padrões de processo a serem seguidos pelas equipes de desenvolvimento de *software*. As metodologias de desenvolvimento de *software* disponíveis tanto para as equipes de desenvolvimento internas quanto externas é o *Scrum* e o Modelo V.

## Caracterização do projeto analisado

O projeto estudado foi nominado de Seguro Prestamista PJ, pois se trata do desenvolvimento de um *software* para gerenciar a comercialização de um novo produto da Seguradora do Banco Beta. Este produto é um seguro para empresas que quitam as operações de empréstimo em caso de imprevistos com os sócios (falecimento ou invalidez). O projeto foi concluído em maio de 2012, consumiu 20 mil horas/homem (R$3,5 milhões) em oito meses e envolveu a integração entre dez sistemas corporativos independentes. Foi considerado um caso de sucesso, pois, em menos de um ano (abril de 2013) de comercialização do produto, gerou uma receita líquida de R$146 milhões para o Banco Beta.

As características peculiares do produto permitiram segregar o desenvolvimento do *software* em três grandes funcionalidades e implantá-las de forma incremental. Quando ocorre o sinistro (falecimento ou invalidez de um dos sócios), inicia-se o processo de solicitação e aprovação da quitação do empréstimo que leva sessenta dias. Dessa forma, foi possível que a funcionalidade de controle do sinistro fosse implantada dois meses após o início das vendas do produto sem risco para o negócio. Conforme demonstrado na Figura 4, a Estrutura Analítica do Projeto (EAP) foi subdividida em: (a) venda da operação de empréstimo com seguro, (b) manutenção da operação; e (c) sinistro.

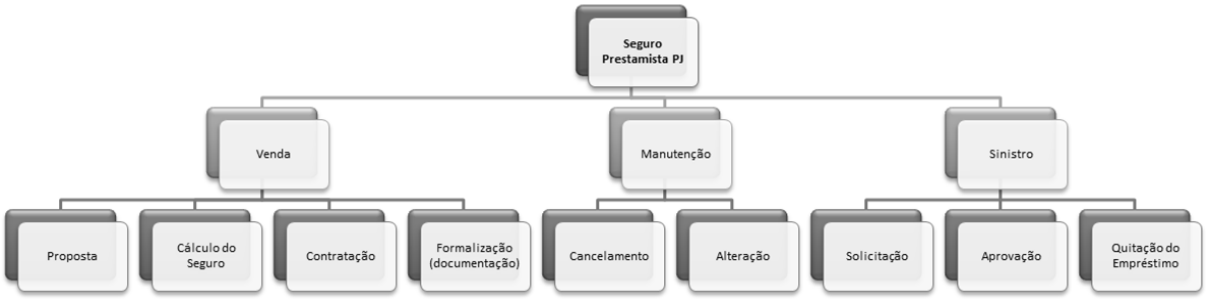


**Figura 4.** EAP do Projeto Seguro Prestamista PJ.
Fonte: Elaborado pelos autores.

O desenvolvimento do novo *software* foi terceirizado para uma consultoria homologada pelo Banco Beta. Entretanto a manutenção dos sistemas corporativos (seguros, empréstimos, limite de crédito, garantias, contabilidade, conta corrente e correios) e canais de comunicação (agências e internet *banking*) foi realizada internamente.

Este produto, depois de implantado, utiliza toda a capilaridade de vendas do Banco Beta e é comercializado pelas agências e pela internet em todo o território nacional. Os benefícios esperados para o produto foram atingidos: (a) proteção para o cliente em caso de infortúnios com os sócios; (b) aumento da receita da seguradora com a venda de um novo produto e expansão da carteira de clientes; e (c) redução do risco de crédito para o banco, pois estudos internos indicavam que a morte ou invalidez dos sócios estava diretamente relacionada com o aumento da inadimplência. Entre as metodologias de desenvolvimento de *software* homologadas pelo Banco Beta, optou-se pelo Modelo V aplicado de forma incremental.

## Tipo de Intervenção e Mecanismos Adotados

A decisão quanto à MDS a ser adotada no projeto em questão foi tomada pelo gerente com apoio da equipe na reunião de identificação e análise dos riscos durante a fase de planejamento. Foram considerados os

seguintes aspectos: (a) ausência de conhecimento e experiência em *Scrum* pela equipe do projeto, (b) havia restrição da data de implantação imposta pela área executiva do Banco Beta, e (c) o projeto era classificado como estratégico pelo Escritório de Projetos. A equipe entendeu que seria um risco ao projeto aplicar uma metodologia desconhecida em um projeto com tais características e decidiu adotar o Modelo V para eliminar o risco.

Durante o planejamento, esta foi a única análise realizada para decidir qual MDS deveria ser utilizada e pareceu ser a mais assertiva na época. O gerente, junto com a equipe, tinha a prerrogativa de tomar a decisão e não era necessário justificar o motivo de tal decisão para o Escritório de Projetos. Entretanto a decisão foi tomada com base na conveniência e experiência profissional da equipe do projeto, ou seja, somente com base em critérios subjetivos.

Ao utilizar o Modelo V, o projeto foi desenvolvido e implantado dentro do custo e prazo planejados, além de atender as expectativas do cliente. Apesar do bom desempenho do projeto, a equipe do projeto, conscientemente, pulou uma etapa importante do processo, que seria verificar a existência de um procedimento definido com critérios assertivos para decidir qual MDS utilizar, sendo este o principal motivador da elaboração deste estudo.

## Resultados Obtidos e Análise

Com base nas metodologias de desenvolvimento de *software* estudadas, foram estabelecidos os critérios para aplicação do *Scrum* ou Modelo V (primeira coluna das Tabelas 2 e 3). E, a partir da pesquisa documental dos artefatos do projeto, entrevistas com os colaboradores da equipe e a observação do autor que participou como gerente do projeto, foram analisadas as características do projeto para determinar sua aderência a cada um dos critérios estabelecidos.

Tabela 2

**Aplicabilidade do *Scrum* ao Projeto Estudado**

| Critérios para aplicação do *Scrum* | Características do Projeto | | |
|---|---|---|---|
| | Atende | Não atende | Resultado e Análise |
| Projetos executados por um time de desenvolvimento de 3 a 9 pessoas, desconsiderando o *Scrum* Master e o Dono do Produto (Schwaber & Sutherland, 2011). | | X | O time de desenvolvimento do projeto era de 16 pessoas. |
| Projetos que possam ter implantações relevantes em ambiente produtivo a cada 2-4 semanas (Schwaber & Sutherland, 2011). | | X | Foi possível dividir o escopo do projeto em diversas funcionalidades, mas o cronograma de implantação envolvia a sincronização entre diversos sistemas corporativos com datas de implantação predefinidas. |
| Escopo do projeto não está claramente definido (Schwaber & Sutherland, 2011). | | X | O escopo total do projeto estava claramente definido no início do projeto. |
| Equipes alocadas no mesmo ambiente físico (Lalsing *et al.*, 2012). | | X | Conforme o PMBOK® (Project Management Institute, 2013), a estrutura organizacional era matricial balanceada onde as equipes dos sistemas corporativos não se dedicavam em tempo integral ao projeto, pois estavam alocadas simultaneamente em outros projetos. Assim, não podiam ser deslocadas para outro ambiente físico. |
| Equipe estável (Lalsing *et al.*, 2012). | X | | A equipe do projeto era estável, pois a equipe definida no início do projeto participou até o final do projeto. |

**Nota**. Fonte: Elaborado pelos autores.

Tabela 3

**Aplicabilidade do Modelo V de Forma Incremental ao Projeto Estudado**

| Critérios para aplicação do Modelo V de forma incremental | Características do Projeto | | |
|---|---|---|---|
| | Atende | Não atende | Resultado e Análise |
| Projetos previsíveis com requisitos claramente definidos (Pressman, 2011). | X | | Os requisitos do projeto estavam claramente definidos e aprovados pelo patrocinador desde o início do projeto. O Modelo V era utilizado há bastante tempo na organização e suas fases e artefatos, conhecidos pela equipe do projeto. |
| Projeto com entregas incrementais (Pressman, 2011). | X | | Conforme apresentado na EAP do projeto, as funcionalidades foram entregues de forma incremental. |
| Sistema com previsão de manutenção futura frequente (Guntamukkala *et al.*, 2006). | X | | Trata-se da criação de um novo sistema desenvolvido na plataforma *mainframe* (grande porte), com expectativa de manutenção futura para expandir o seguro para outros tipos de empréstimos. |

**Nota**. Fonte: Elaborado pelos autores.

Nas Tabelas 2 e 3, ao confrontar a teoria revisada nas metodologias de desenvolvimento de *software* com as características do projeto é possível avaliar de forma clara e organizada a possível aplicabilidade de cada uma das metodologias ao projeto em questão.

No projeto estudado, verificou-se que o Modelo V, aplicado de forma incremental, atendia melhor os critérios definidos neste estudo, pois: (a) o escopo estava claramente definido, o que permitiu à equipe desenvolver diversas funcionalidades em paralelo e sem a intervenção diária do Dono do Produto; (b) a equipe do projeto era matricial balanceada, pois, como cada sistema corporativo era de responsabilidade de um gerente funcional diferente, o gerente de projeto não tinha autoridade sobre alguns dos 16 desenvolvedores do time; e (c) não era possível alocar todos os recursos no mesmo ambiente físico, o que é uma exigência do *Scrum*.

A Tabela 4 apresenta um resumo comparativo entre as duas metodologias de desenvolvimento de *software* estudadas.

Tabela 4

**Resumo Comparativo o *Scrum* e Modelo V**

| Características | Scrum | Modelo V |
|---|---|---|
| Estrutura organizacional [a] | Projetizada ou matricial forte. | Qualquer tipo de estrutura (funcional, matricial ou projetizada). |
| Ambiente Físico | Membros da equipe do projeto alocados no mesmo ambiente físico. | Membros da equipe do projeto alocados no mesmo ambiente físico ou em ambientes físicos distintos. |
| Dedicação dos membros da equipe do projeto | Tempo integral. | Tempo integral ou parcial. |
| Frequência de intervenção do Gerente/Líder | Diária (daily meeting). | Conforme definido no plano de comunicação do projeto. |
| Escopo | Variável. | Fixo. |
| Prazo | Fixo (2-4 semanas). | Fixo. |
| Custo | Fixo (tamanho fixo da equipe). | Fixo. |
| Tipo de contrato mais adequado | Tempo e Material [b]. | Preço Fixo [c]. |

**Nota**. Fonte: Elaborado pelos autores.
[a] A estrutura organizacional é um fator ambiental da empresa que pode afetar a disponibilidade dos recursos. O nível de autoridade do gerente de projeto varia conforme o tipo de estrutura organizacional: funcional – autoridade inexistente; matricial fraca – pouca autoridade; matricial balanceada – autoridade moderada; matricial forte – autoridade alta; projetizada – autoridade total (Project Management Institute, 2013). [b] O contrato por Tempo e Material costuma ser usado para aumento de pessoal quando não é possível detalhar os requisitos do trabalho (Project Management Institute, 2013). [c] No contrato de Preço Fixo os compradores devem especificar, com precisão, os requisitos do trabalho adquirido (Project Management Institute, 2013).

## Conclusão

Os principais aspectos a serem analisados antes de se decidir qual metodologia utilizar são: tipo da estrutura organizacional e nível de detalhamento do escopo. Com relação à estrutura organizacional, o *Scrum* necessita de dedicação integral e trabalho conjunto da equipe, inclusive fisicamente e com participação do Dono do Produto. Desse modo, o modelo matricial forte ou projetizado é mais adequado, pois, conquanto não exista a figura hierárquica do chefe, o *Scrum Master* é considerado o líder da equipe e é responsável pelo andamento da execução do projeto. E, ao contrário, o Modelo V não apresenta qualquer restrição à estrutura organizacional, podendo ser executado em uma estrutura funcional, matricial ou projetizada. Como as tarefas são encadeadas sequencialmente e os responsáveis definidos no planejamento do projeto, por conseguinte, o monitoramento das atividades pode ser realizado pelo gerente do projeto sem a necessidade de intervenção diária.

Com relação ao nível de detalhamento do escopo do projeto, o Modelo V é perfeitamente adequado quando os requisitos do projeto (escopo) estão muito bem-definidos e são facilmente entendidos por toda a equipe de desenvolvimento. O escopo detalhado permite a elaboração de um orçamento com prazo e custo assertivos, o que melhora a previsibilidade do processo, dessa forma, sugere-se um contrato do tipo Preço Fechado com o fornecedor. Entretanto, se os requisitos do projeto não estão claramente definidos, o Modelo V tem poucas chances de sucesso, portanto, neste caso, o *Scrum* é mais adequado. O *Scrum* parte da premissa de que o prazo e custo são fixos, enquanto o escopo é variável e será definido durante o desenvolvimento do projeto, por isso, sugere-se um contrato do tipo Tempo e Material com o fornecedor.

As implicações gerenciais deste estudo sugerem que os gestores de projeto, ao decidir entre a utilização do *Scrum* ou Modelo V,

avaliem cinco questões: (a) a equipe de desenvolvimento varia de 3 a 9 pessoas? (b) é possível implantar funcionalidades relevantes em ambiente produtivo a cada 2-4 semanas? (c) o escopo do projeto não está claramente definido? (d) existe disponibilidade de espaço físico para alocação da equipe no mesmo ambiente? (e) os gerentes funcionais liberam seus colaboradores até a conclusão do projeto? Ao responder sim para todas estas perguntas, é possível considerar a utilização do *Scrum*, caso contrário, sugere-se utilizar o Modelo V.

Dessa forma, ao propor um *check-list* com cinco questões, responde-se à questão de pesquisa apresentada para este estudo, o qual se propôs a analisar como utilizar critérios assertivos para decidir entre a utilização das duas metodologias estudadas.

O principal fator limitante para este estudo foi a atuação direta do autor como gerente do projeto que tinha a prerrogativa de decidir qual MDS utilizar. Em pesquisas futuras é importante analisar projetos que utilizem o *Scrum* e quais as vantagens obtidas em relação à utilização do Modelo V.

## Referências


Beck, K., Beedle, M., van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., & Grenning, J. (2001). *Manifesto for agile software development*. Recuperado de http://agilemanifesto.org/

Boehm, B. W. (1988). A spiral model of software development and enhancement. *Computer, 21*(5), 61-72. doi: 10.1109/2.59

Bucanac, C. (1999). *The V-Model*. Recuperado de http:// www.bucanac.com/documents/The_V-Model.pdf

Federação Brasileira de Bancos. (2012a). *CIAB FEBRABAN 2012*. *A sociedade conectada.* Recuperado de http://www.febraban.org.br/7Rof7SWg6qmyvwJcFwF7I0aSDf9jyV/sitefebraban/Ciab12-Anuario%20Febraban%2006.07.pdf

Federação Brasileira de Bancos. (2012b). *Relatório Anual 2012.* Recuperado de http://www.febraban.org.br/Relat%C3%B3rio_Anual_FBB_2012.pdf

Gil, A. C. G. (2010). *Como elaborar projetos de pesquisa* (5a ed.). São Paulo: Atlas.

Guntamukkala, V., Wen, H. J., & Tarn, J. M. (2006). An empirical study of selecting *software* development life cycle models. *Human Systems Management, 25*(4), 265-278.

Lalsing, V., Kishnah, S., & Pudaruth, S. (2012). People factors in agile software development and project management. *International Journal of Software Engineering & Applications, 3*(1), 117-137. doi: 10.5121/ijsea.2012.3109

Martins, G. A. de, & Theóphilo, C. R. (2009). *Metodologia da investigação científica para ciências sociais aplicadas* (2a ed.). São Paulo: Atlas.

Pressman, R. S. (2011). *Engenharia de software: uma abordagem profissional* (7a ed.). Porto Alegre: AMGH.

Project Management Institute. (2013). *A guide to the project management body of knowledge: (PMBOK® guide)*. Newtown Square: PMI.

Royce, W. W. (1970, Agosto). Managing the development of large software systems. *Anais do IEEE WESCON*, Los Alamitos, CA, USA, 26. Recuperado de http://leadinganswers.typepad.com/leading_answers/files/original_waterfall_paper_winston_royce.pdf

Schwaber, K., & Sutherland, J. (2011). *The scrum guide*. Recuperado de https://www.scrum.org/Portals/0/Documents/Scrum%20Guides/2013/Scrum-Guide.pdf#zoom=100

Sommerville, I. (2011). *Software engineering* (9a ed.). Boston: Pearson Education.