





## **Modelos e metodologias de desenvolvimento de software**

Conteúdo organizado por **Renato Cividini Matthiesen** em 2022 dos artigos “Análise da abordagem **LOW-CODE** como facilitador da transformação digital em indústrias”, “Desenvolvimento de aplicação móvel para submissão/revisão de despesas recorrendo a metodologias ágeis de desenvolvimento e à plataforma low-code **OutSystems**” e “Desenvolvimento rápido de aplicações – comparação de soluções em **Outsystems** e **Mendix**”.

### **Objetivos de Aprendizagem**

- Compreender e Analisar Modelos e Metodologias de desenvolvimento de *Software*, Ferramentas de desenvolvimento rápido de soluções, Arquitetura de Soluções, Testabilidade, Disponibilidade e Escalabilidade de Estudos de caso *Low Code*.

# Introdução



**Para ajudá-lo a decidir quando controlar o acesso do usuário às informações, faça a si mesmo a seguinte pergunta: isso é uma sugestão ou uma imposição”.**

Tomasi e Tilton (2021, p. 33)

Diversas são as abordagens, metodologias e modelos de desenvolvimento de sistemas de software, e a escolha de uma determinada abordagem deve levar em consideração a *natureza do projeto*, define Sommerville (2018).

Para projetos chamados de preditivos, que possuem definições de requisitos claros, escopo, tempo, qualidade bem definidos já na fase inicial do projeto o uso de modelos clássicos (ou tradicionais) para desenvolvimento de software pode ser uma boa escolha.

Para projetos de natureza adaptativa, onde os requisitos não são bem definidos no início do projeto e possuem desenvolvimento incremental e iterativo, o uso de métodos ágeis podem ser uma boa opção.

No contexto de Engenharia de Software e desenvolvimento de projetos de software existe a oportunidade de escolher uma melhor abordagem para conduzir o desenvolvimento e a gestão do projeto, ou ainda escolher ferramentas, métodos e metodologias que o façam de forma integrada, buscando em cada abordagem o que ela oferece de melhor para cada projeto. Aqui fazemos referência a projetos de maior porte, além de aplicações mais simples que podem ser desenvolvidas com ferramentas da abordagem Low Code.

Ao considerar a necessidade de desenvolvimento de projetos mais complexos de software, Sbrocco (2012, p. 18) defende que **“nos últimos dez anos, as metodologias ágeis têm surgido pela necessidade de buscar alternativas para os modelos tradicionais de desenvolvimento de projetos”**. Diferentes aspectos e um projeto diferenciam as metodologias tradicionais das ágeis no que diz respeito a estrutura da produção do software e no *mindset* da equipe de desenvolvimento.

E Coutinho (2019) nos ensina que a estratégia ágil para desenvolvimento de sistemas de software é vista como uma abordagem adaptativa que traz compatibilidade com mercados dinâmicos da atualidade, o que leva o ágil a maior adaptação a gestão de produtos e serviços em mercados em frequente mutação, com dinâmicas de revisão de produtos e serviços e intensidade de inovações, principalmente baseado em tecnologias.

Estas abordagens são importantes quando há necessidade de desenvolvimento de aplicações de software que transcendam a praticidade e simplicidade de pequenas aplicações, ainda que façam uso de ferramentas baseada no reuso de componentes e módulos para o desenvolvimento de sistemas de software.

As abordagens de desenvolvimento de software atuais são diversas devido a um período evolutivo das linguagens de programação, técnicas de desenvolvimento de sistemas, linguagens de modelagem de sistemas e softwares de apoio, que constituem atualmente plataformas e *frameworks* de apoio a automação de programação de sistemas e aplicativos de software.

## Saiba Mais

As metodologias ágeis têm como ponto de partida o chamado Manifesto ágil, documento desenvolvido no ano de 2001 como saída de uma reunião entre 17 profissionais de desenvolvimento de sistemas de software que já utilizavam algumas metodologias como o XP, DSDM, Scrum dentre outras com uma reflexão sobre uma nova forma de desenvolvimento de sistemas e gestão de projetos de software.

As metodologias ágeis se baseiam em quatro princípios definidos no Manifesto Ágil.

São eles:

- 1. *Indivíduos e iterações*** são mais importantes do que processos e ferramentas.
- 2. *Software funcionando*** é mais importante que documentação completa.
- 3. *Colaboração*** com o cliente é mais importante que negociação de contratos.
- 4. *Adaptação a mudanças*** é mais importante que seguir o plano inicial.

As metodologias ágeis na gestão de projetos de software vêm ganhando força nos últimos vinte anos considerando a maior complexidade de projetos de sistemas de software. Novas técnicas e ferramentas de gestão se unem às metodologias tradicionais e também às metodologias ágeis para maximizar a produtividade em projetos. Enquanto parte dos sistemas necessitam ser desenvolvidos por especialistas em tecnologia da informação, outras podem ser desenvolvidas por profissionais de outras áreas de forma colaborativa.

Os métodos ágeis permitem entregas incrementais, adaptação no projeto e forte interação com a integração do cliente e equipe de projeto e cooperação. Ao abordar os métodos ágeis disponíveis no mercado e já com razoável maturidade são enaltecidas filosofias de gestão como o *Lean* e metodologias de desenvolvimento de projetos como o *Scrum*, o *eXtreme Programming*.

Vejamos com um pouco mais de detalhes três metodologias ágeis para o desenvolvimento de sistemas de software mais complexos:

- 1. Scrum:** metodologia ágil mais utilizada na atualidade, com origem no ano de 1993, o *Scrum* foi desenvolvido por Jeff Sutherland e tem como foco o andamento do projeto de *software* através de controles realizados por um subconjunto de atividades realizadas em formato de rituais, chamados de *Sprints*. A metodologia *Scrum* faz uso de *Backlogs* para manter o controle do escopo e objetivo do projeto (nos preceitos da Engenharia de Requisitos), com as características e requisitos do produto de *software* assim como o objetivo de cada *Sprint* (subconjunto do projeto de desenvolvimento do software).
- 2. eXtreme Programming (XP):** metodologia ágil desenvolvida em 1997 desenvolvida para oferecer excelência no desenvolvimento de *softwares* baseada em agilidade, economia e qualidade. A Extreme Programming tem como foco entregas parciais, com qualidade e eficiência através de histórias, que representam os requisitos funcionais do projeto, estabelecidos pelo cliente e com práticas de programação em pares (mais de um programador desenvolvendo cada parte do sistema).
- 3. Microsoft Solutions Framework (MSF):** metodologia ágil criada pela Microsoft em 1994 com um conjunto de boas práticas verificadas em seus serviços de consultoria, desenvolvidas para criar um *framework* flexível, capaz de guiar o desenvolvimento ágil de sistemas de *software*.

## FERRAMENTAS DE APOIO AO DESENVOLVIMENTO DE SOFTWARE ÁGIL



Aliadas do desenvolvimento de sistemas de software em diferentes abordagens (tradicional considerando o modelo em cascata, também chamada de **waterfall**, onde uma fase do projeto é iniciada quando a antecessora finalizar) ou ágil, desenvolvida de forma interativa e adaptativa por menores ciclos de desenvolvimento e entregas de versões ao cliente, ferramentas de gestão são utilizadas para auxiliar a gestão dos projetos. Estas ferramentas e abordagens também podem ser utilizadas para o desenvolvimento de aplicações com aplicações Low Code. Vejamos algumas delas:

1. **Lean**: trata-se de uma cultura originária da indústria japonesa de montadoras de automóveis e possui alinhamento com o Manifesto Ágil. O *Lean* é considerado um modelo cultural para busca excelência em operações de acordo com Ries (2019).
2. **Design Thinking**: é uma abordagem de desenvolvimento de produtos e inovação que busca mudar o nosso estado mental (*mindset*) para resolver problemas com novas perspectivas. A abordagem se baseia na capacidade de ser intuitivo, reconhecer padrões, desenvolver ideias que tenham um significado emocional

além do funcional, nos expressar em mídias além de palavras ou símbolos. As fases do projeto no do *design thinking* são: Descoberta (Empathize), Interpretação (*Define*), ideação (*Ideate*), experimentação (*Prototype*) e evolução (*Test*).

3. **PM Canvas:** originário do *Project Model Canvas* que permite a criação de plano ou modelo de negócios. A ferramenta visual de ideação de projetos possui foco central em desenvolver toda uma visão completa de um negócio de forma rápida e clara. Utiliza-se de um painel com nove campos para descrição das características de um negócio, ou um projeto, e não exige nada além de caneta e alguns impressos em uma folha de papel.
4. **Scrumboard:** é um quadro de acompanhamento de processos que utiliza a concepção do Kanban (controle com cartões) para implementação e acompanhamento dos processos e tarefas na metodologia *Scrum*. Através dele são controladas o *Product Backlog* e seu andamento durante os *Sprints* no modelo *Scrum*.
5. **OKR (*Objectives and Keys Results*):** sistema de definição e gestão de indicadores de desempenho que estabelece um conjunto de objetivos inter relacionados para contribuir com a estratégia da organização. São traçados objetivos mensuráveis que se relacionam à estratégia da empresa, com grande significado aos negócios, em uma linguagem acessível a todos os níveis organizacionais. Eles devem ser gerenciados em ciclos de três meses, em geral, e levam a empresa a uma reflexão e realinhamento de táticas e operações constantes. O OKR deve responder duas perguntas de acordo: Aonde quero ir? (Objetivos) e Como sei que estou chegando lá? Esta metodologia é utilizada por empresas como Oracle, LinkedIn, Twitter e Facebook.

## Saiba Mais

Existem diversos sistemas de softwares e aplicativos úteis para a gestão de um projeto de software e podem ser utilizados de forma livre, outros com custo de licenciamento. Segue exemplos:

- **Trello:** a ferramenta traz opções de utilização de quadros, listas, cartões, membros, *checklists* e anexos, um *power-up* por quadro (integrações com outras ferramentas).
- **Wrike:** software que permite o gerenciamento de tarefas, atualizações em tempo real, integração (Google Drive, Dropbox, Box, Microsoft Office 365, OneDrive e iCal) e aplicativos para Android e iPhone.
- **Taiga:** a ferramenta permite fazer projetos públicos ilimitados.
- **PangoScrum:** ferramenta permite o gerenciamento do *Product Backlog*, planejamento de *Sprints*, Monitoramento de progresso e Agendamento de eventos.
- **Asana:** aplicação de *software* que permite trabalhar com tarefas, conversas e projetos ilimitados e painéis básicos.

## Desenvolvimento rápido de software

O desenvolvimento rápido de software deverá então fazer uso de uma nova abordagem de desenvolvimento de aplicações que transcendem os modelos ágeis de desenvolvimento, exemplificados pelo *Scrum*, *eXtreme Programming* dentre outras. A autora considera que as organizações que ainda não fazem uso de metodologias ágeis e adaptativas deverão conhecer e buscar adaptação ao chamado RAD (*Rapid Application Development*) para entrega mais acelerada e menos custosa de produtos de software, principalmente para clientes internos de uma organização empresarial.

Pinto (2021) nos lembra que as transformações tecnológicas e as mudanças rápidas das necessidades dos clientes têm pressionado as empresas a manterem modelos adaptativos de desenvolvimento de sistemas e mesmo de gestão, para conseguirem dar respostas aos seus clientes.

Como objeto de estudo para desenvolvimento rápido de softwares, vamos apresentar aqui uma metodologia ágil de desenvolvimento de sistemas de software criado pela empresa OutSystems.

De acordo com Karmali (2019), a OutSystems sugere uma abordagem diferente para organização e controle de projetos, baseando-se nas metodologias ágeis. A necessidade de entrega rápida e constante mudança é atendida, entregando assim aplicações que realmente correspondem às necessidades solicitadas pelo negócio.

A metodologia apresentada pela autora é um modelo de gestão ágil, com semelhanças relatadas junto a metodologia *Scrum*, mas adaptada de forma a serem adequados à natureza particular de projetos realizados dentro de plataformas de geração de aplicações através da abordagem Low Code.

A metodologia desenvolvida pela OutSystems utiliza-se dos *Sprints* do Scrum onde há uma demonstração da versão atual da aplicação desenvolvida para coleta e análise de *feedback* e gestão de requisitos para a organização dos próximos *Sprints*, com duração de uma ou duas semanas com fases de análise de requisitos, desenvolvimento do sistema e testes.

Os requisitos que não estão claros ou tiveram algum tipo de modificação são revisitados e reestruturados para adaptação do sistema e programação nos *Sprints* futuros. Karmali (2019) nos ensina que o OutSystems dá ênfase à relação de tempo e esforço e requisitos entregues, limitando o período do projeto e colocando foco especial na organização dos recursos para garantia de entrega nos prazos determinados e conforme orçamento estabelecido.

Por fim, na metodologia ágil criada pela OutSystems para desenvolvimento de aplicações em suas plataformas de Low Code, utiliza-se o conceito de *training* para recolher uma grande quantidade de *feedback* e realizar as devidas alterações que podem ser cruciais para o desenvolvimento iterativo do projeto antes da entrega de uma versão finalizada.

## Saiba Mais

A metodologia Scrum busca gerenciar projetos adaptáveis com resultados rápidos, eficientes, flexíveis e com processos iterativos. A estrutura da metodologia utilizada para esta iteração é o Sprint. Esta estrutura de gestão é um elemento Time boxed, pois possui restrição de tempo para garantir que os membros do Scrum desenvolvam as atividades de desenvolvimento dentro de períodos planejados.

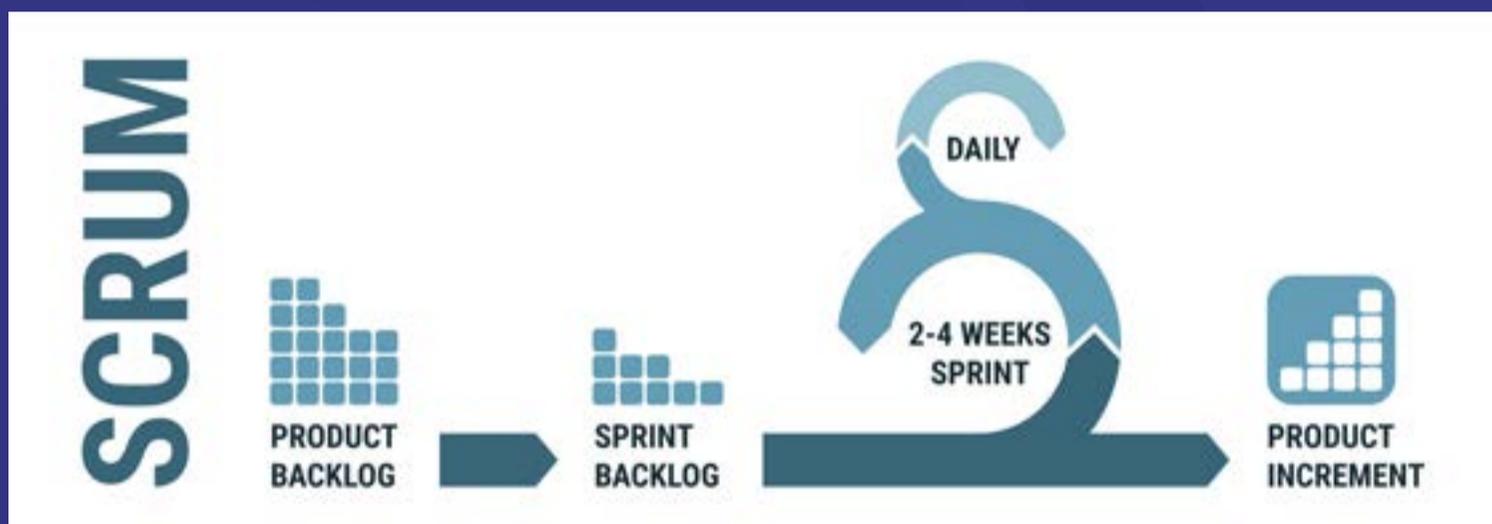
Cada Sprint pode ser considerado com um subprojeto com período de tempo limitado a um mês de duração. De acordo com Sutherland e Sutherland (2019, p. 79), “o tempo é o maior limitador dos projetos humanos. Ele afeta tudo”.

As cerimônias são os rituais ou reuniões na metodologia Scrum, utilizadas para a gestão e acompanhamento do projeto. São atividades realizadas em tempos definidos (timeboxing). São cerimônias no Scrum:

- ***Sprint Planning***: reunião realizada para início de um *Sprint* para criar o *Sprint Backlog*.
- ***Daily Scrum***: ritual com uma reunião diária para planejamento das atividades no desenvolvimento dos sistemas a serem realizadas no dia seguinte.
- ***Sprint Review***: ritual com análise dos entregáveis e apresentação ao Dono do Produto.

- ***Sprint Retrospective Meeting***: ritual realizado para analisar e ajustar problemas encontrados pelo time após os *Sprint Reviews* e antes do próximo *Sprint*. Veja a figura a seguir com a representação do *Sprint* dentro do *Scrum*.

Figura 1 – Sprint no Scrum



Fonte: Shutterstock.

## Em Resumo

Nesta aula, você pode perceber que diversas são as abordagens, metodologias e modelos de desenvolvimento de sistemas de software, e a escolha de uma determinada abordagem deve levar em consideração a *natureza do projeto*. No contexto de Engenharia de Software e desenvolvimento de projetos de software existe a oportunidade de escolher uma melhor abordagem para conduzir o desenvolvimento e a gestão do projeto, ou ainda escolher ferramentas, métodos e metodologias que o façam de forma integrada, buscando em cada abordagem o que ela oferece de melhor para cada projeto. Aqui fazemos referência a projetos de maior porte, além de aplicações mais simples que podem ser desenvolvidas com ferramentas da abordagem Low Code. Os métodos ágeis permitem entregas incrementais, adaptação no projeto

e forte interação com a integração do cliente e equipe de projeto e cooperação. Ao abordar os métodos ágeis disponíveis no mercado e já com razoável maturidade são enaltecidas filosofias de gestão como o *Lean* e metodologias de desenvolvimento de projetos como o *Scrum*, o *eXtreme Programming*. A OutSystems sugere uma abordagem diferente para organização e controle de projetos, baseando-se nas metodologias ágeis. A necessidade de entrega rápida e constante mudança é atendida, entregando assim aplicações que realmente correspondem às necessidades solicitadas pelo negócio. A metodologia desenvolvida pela OutSystems utiliza-se dos *Sprints* do Scrum onde há uma demonstração da versão atual da aplicação desenvolvida para coleta e análise de *feedback* e gestão de requisitos para a organização dos próximos *Sprints*, com duração de uma ou duas semanas com fases de análise de requisitos, desenvolvimento do sistema e testes.

## Na ponta da língua



## Referências Bibliográficas

COUTINHO, Heitor. (2019). **Da estratégia ágil aos resultados**. São Paulo: Saraiva.

KARMALI, S. (2019). Desenvolvimento de aplicação móvel para submissão/revisão de despesas recorrendo a metodologias ágeis de desenvolvimento e à plataforma *low-code* OutSystems. [Departamento de Ciências e Tecnologias da Informação da ISCTE]. **Instituto Universitário de Lisboa**. Lisboa, PT.

PINTO, I. M. M. M. (2021). Desenvolvimento rápido de aplicações – comparação de soluções em Outsystems e Mendiz. [Instituto Superior de Engenharia do Produto]. **Politécnica do Porto**. Porto, PT.

RIES, Eric. (2019). **A startup enxuta**: como usar a inovação contínua para criar negócios radicalmente bem-sucedidos. Rio de Janeiro: Sextante.

SBROCCO, Jose. H. T. C.; MACEDO, Paulo C. de. (2012). **Metodologias ágeis: Engenharia de Software sob medida**. São Paulo: Érica.

SOMMERVILLE, Ian. (2018). Engenharia de Software, 10 ed. Pearson. Ebook Pearson.

SUTHERLAND, Jeff; SUTHERLAND, J. J. (2019). **Scrum**: a arte de fazer o dobro do trabalho na metade do tempo. Rio de Janeiro: Sextante.

TOMASI, C.; TILTON, B. (2021). **Low-code apps**. Hoboken, NJ: John Wiley & Sons, Inc.





## **ARTIGOS DE REFERÊNCIA:**

### **Análise da abordagem LOW-CODE como facilitador da transformação digital em indústrias.**

[Revista e-TECH: Tecnologias para competitividade industrial, V. 15, n. 2]. E-TECH Tecnologias para competitividade Industrial. Florianópolis, BR - ALVES, F.; ALCALÁ, S. G. S. (2022). DOI: <https://bit.ly/3S1UNpC>

### **Desenvolvimento de aplicação móvel para submissão/revisão de despesas recorrendo a metodologias ágeis de desenvolvimento e à plataforma low-code OutSystems.**

[Departamento de Ciências e Tecnologias da Informação da ISCTE]. Instituto Universitário de Lisboa. Lisboa, PT - KARMALI, S. (2019).

### **Desenvolvimento rápido de aplicações – comparação de soluções m Outsystems e Mendiz.**

[Instituto Superior de Engenharia do Produto]. Politécnica do Porto. Porto, PT. PINTO, I. M. M. M. (2021).



**MUST**  
**UNIVERSITY**  
FLORIDA - USA