

Os processos de desenvolvimento de programas no passado, era um contexto muito diferente do atual e facilmente ocasionava um alto número de projetos de software que não chegam ao fim ou em muitos casos ultrapassava orçamento. Nos estudos de Pressman (1995), a engenharia de software surge em meio a crise de software, para a produção ganhar eficiência. A engenharia de software tornar o desenvolvimento de software um processo organizado, planejado e padronizado. Este processo permite ver “qual” o problema e define “quem” faz “o que”, “quando” e “como” de modo a atingir o seu objetivo (PRESSMAN 1995).

Em seu caráter mais básico o processo de desenvolvimento possui as atividades fundamentais como, Especificação do software, aqui é preciso definir as funcionalidades e restrições do software; Validação, é necessário verificar se o software corresponde ao que o cliente desejava; Projeto e Implementação, nesta etapa é feita a implementação e testes do software de forma que cumpra suas especificações; e sua Evolução, as necessidades do cliente sempre muda então o software precisa evoluir de modo que estas necessidades sejam atendidas. A primeira metodologia de desenvolvimento, também denominada de Modelo Classico ou modelo em cascata é conhecida por dividir as etapas em fases, de forma que cada fase deveria sair um documento de fechamento da fase assinado pelo cliente, ciclo a ciclo e apenas no final do projeto algum produto era apresentado, um problema clássico deste modelo é que se uma mudança em um requisito for proposta, provavelmente será obrigado a retornar a fase anterior e verificar e validar novamente os requisitos. Esses modelos tradicionais são chamados de pesados ou orientados a documentação, e geram certas frustrações por planejar e documentado antes de ser implementado.

Nos anos 90 começaram a surgir processos alternativos de desenvolvimento apelidados de “leves”, em resposta aos considerados excessivamente lentos e burocráticos. Diante dos desafios de um mundo corporativo, alta [tecnologia](#) e competitividade, o desenvolvimento de produtos de software precisou mudar o ritmo. O conceito de Agile foi rapidamente difundido entre especialistas, o que fez frente aos [modelos tradicionais de desenvolvimento](#). Tais metodologias tem como principal objetivo reduzir o ciclo de desenvolvimento em semanas ou meses, já que os modelos “conservadores”, esse ciclo pode durar anos. Em fevereiro de 2001 praticantes de métodos leves de desenvolvimento se reunirão e deliberaram um documento chamado de Manifesto para Desenvolvimento Ágil de Software, ou só Manifesto Ágil.

O Manifesto aborda os valores fundamentais do manifesto ágil que são, Indivíduos e interações (mais que processos e ferramentas), Software em funcionamento (mais que documentação abrangente), Colaboração com o cliente (mais que negociações de contratos), Responder a mudanças (mais que seguir planos). que todos os profissionais ali reunidos acordaram em seguir e disseminar. Esses quatro valores norteiam os 12 princípios base como descrito: Maior prioridade é satisfazer o cliente com entregas contínuas; Mudanças nos

requisitos são bem-vindas em qualquer fase; Mudanças podem trazer vantagens competitivas para o cliente; Entregas frequentes de software funcionando em ciclos curtos (semanas ou poucos meses); Times multidisciplinares onde cliente e equipe trabalhem diariamente em conjunto por todo o projeto. Construa projetos em torno de indivíduos motivados, forneça a eles o ambiente e o suporte necessário e confie neles para fazer o trabalho; O método mais eficiente e eficaz de transmitir informações para e entre uma equipe é através de conversa face a face; Software funcionando é a medida primária de progresso; Os processos ágeis promovem desenvolvimento sustentável. Os patrocinadores, desenvolvedores e usuários devem ser capazes de manter um ritmo constante indefinidamente; Contínua atenção à excelência técnica e bom design aumenta a agilidade; Simplicidade-a arte de maximizar a quantidade de trabalho não realizado-é essencial; As melhores arquiteturas, requisitos e designs emergem de equipes auto-organizáveis; Em intervalos regulares, a equipe reflete sobre como se tornar mais eficaz e então refina e ajusta seu comportamento.

Entre os 17 participantes que assinaram o manifesto vários eram praticantes ou criadores dos métodos leves, que agora seriam conhecidos como Métodos Ágeis. Alguns dos signatários do manifesto já praticavam suas metodologias, isso permitiu uma rápida e fácil propagação dos valores e princípios. Algumas das metodologias ágeis representadas na reunião eram, Scrum, Programação Extrema, Adaptive Software Development, entre outros. Uma coisa comum no mundo ágil é a capacidade de adaptação e até mesmo remixagem entre algumas metodologias, de forma que uma pode complementar a outra. Algumas metodologias são mais eficientes quando aplicadas em outras metodologias, um exemplo prático é a metodologia kambam. Vejamos as metodologias citadas de forma a compreender suas principais características, formas de aplicação e como as remixar.

O Scrum é uma metodologia de desenvolvimento ágil que se concentra principalmente sobre como gerenciar tarefas dentro de um ambiente de desenvolvimento baseado em equipes. A popularidade do Scrum decorre de sua relativa simplicidade de implementação e facilitação na gestão. Os projetos são divididos em ciclos, geralmente mensais, chamados de Sprints. Um sprint representa uma tarefa que está sendo executada, seu ciclo pode durar entre uma semana ou um mês. E ao final de cada sprint a parte implementada é incrementalmente um produto entregável ao cliente. As atividades a sempre implementadas em cada sprint está presente em uma lista denominada de Product Backlog, que é um conjunto de prioridades de requisitos de alto nível definidos pelo Product Owner. O product Owner é em muitos casos o dono do projeto ou cliente, ele é a pessoa que mais entende sobre o projeto, por que é dele que sai toda e qualquer demanda ou mudanças. Algumas atividades são elencadas como, Sprint Planning Meeting, reunião mensal utilizada para planejar o Product Backlog, estão presentes tanto o Scrum Master, Time de desenvolvimento quanto o Product Owner, outra reunião, porém de caráter diário são as Daily Scrum, esta reunião visa trocar conhecimento e

informações geradas no dia anterior, como identificar as dificuldades e elencar as prioridades recentes participam desta reunião o Scrum Master e o Time de desenvolvimento. O scrum marter é um mediador entre o Time e o Product Owner, não necessariamente precisa ser um gerente do projeto. Por fim, as duas ultimas atividades são o Sprint Review, uma reunião para rever o que foi ou não foi construído e a Scrum Reperspective que verifica o que deu certo e o que pode ser melhorado para a próxima sprint ou novo.

Também criado na década de 1990, o eXtreme Programming, chamado de XP, é um método ágil com foco no desenvolvimento de softwares com base em três pilares: agilidade no desenvolvimento da solução, economia de recursos e qualidade do produto final. Para chegar à excelência, uma equipe XP deve se basear em valores, isto é, um contrato de atitudes e comportamentos que levam ao sucesso. Os valores do XP são: Comunicação, Simplicidade, Feedback, Coragem e Respeito. A Comunicação, entre clientes e equipe ela deve ser constante. Simplicidade, assegurando que a equipe se concentre em fazer, primeiro, apenas aquilo que é claramente necessário. Feedback, permite encurtar ao máximo a defasagem de tempo entre execução e o resultado observado, desenvolvedores procuram entregar novas funcionalidades no menor prazo possível, para que o cliente compreenda rapidamente as conseqüências daquilo que pediu. Coragem, [XP](#) não tem uma solução mágica para eliminar determinados risco, equipes XP acreditam que errar é natural e quebrar o que vinha funcionando pode acontecer eventualmente, sendo assim procuram se adaptar a elas com segurança e confiar em seus mecanismos como desenvolvimento orientado a testes, [programação em par](#) e integração contínua. O ultimo valor é o Respeito, uma equipe só se preocupa em comunicar-se melhor se, se importarem uns com os outros, sendo um valor básico acima de todos os valores. XP é uma das metodologias mais focadas em testes de software, ela estimula além da programação em par o uso de técnicas de Desenvolvimento Guiado a Testes (TDD) e o uso de teste de integração no XP é quase que obrigatório, permitindo integrar as partes do software que está em produção.

Um primeiro parênteses levantado sobre a adaptabilidade das metodologias ágeis é a dupla, Scrum e XP podem ser usados em conjunto, adaptando o melhor de cada um às necessidades do projeto e da equipe. Algumas práticas importantes como programação em par, reuniões diárias rápidas (stand up meeting), sprints, product owner, integração contínua, entregas constantes, ou seja, habilidades onde uma pode complementar a outra.

A metodologia *Desenvolvimento de Software Adaptativo* (Adaptive Software Development - ASD), é uma técnica para o desenvolvimento de softwares complexos, e concentra-se na *colaboração humana* e na *auto-organização*. Esta dobradinha aparece quando agentes independentes cooperam para criar resultados. Esta metodologia é tem por característica sem incremental e interativa, para suporte a sistemas grandes e complexos, e mantém o cliente sempre presente. Cada iteração do ciclo de desenvolvimento é chamada de missão, que pode mudar ao longo do projeto, ser orientado a componentes o desenvolvimento ideal são sempre em pequenas partes. A ASD tem prazos fixados e tangíveis forçando que a equipe defina severamente decisões do projeto logo cedo, principalmente por ser tolerante a mudanças e todos os itens que são considerados de alto risco tem seu desenvolvimento priorizado.

Com foco em administração da produção, a metodologia Kamban foi originalmente criada na Toyota, com a intenção de encontrar melhorias que mantivesse um alto nível da produção. O seu objetivo é aumentar a eficiência da produção e otimizar seus sistemas de movimentação, produção, realização de tarefas e conclusão de demandas. A implementação do kamban é simples, sendo necessário apenas um quadro dividido em colunas (a fazer, feitas, testadas, integradas, concluído) e cartões identificadores (post-its) para cada tarefa. Cada cartão (post-its) identifica uma tarefa que está pronta para entrar no quadro na primeira coluna e a medida que o fluxo segue até cartão, ou melhor a tarefa chegar na coluna de “Concluído”. **Kanban** implementa conceitos da Teoria das Restrições procurando identificar e administrar restrições que limitam a performance do sistema. O sistema relaciona a capacidade ao número de cartões que podem ser colocados em circulação, evitando criar gargalos comuns, como problemas nos teste ou na integração. Este método a cada ano ganha mais destaque na indústria de TI, pois promove a melhoria contínua no processo de trabalho de muitas equipes de empresas das mais variadas áreas de negócio e tamanho. Da mesma forma que XP e Scrum podem ser complementares, equipes de SCRUM, também têm enxergado bastante valor na forma como o Kamban pode ser associada à gestão do trabalho.

Podemos concluir que o Manifesto Ágil é um marco na Engenharia de Software, bem como a aplicação de métodos ágeis vem crescendo de forma muito rápida. Como já visto, os métodos ágeis podem ser aplicados em diversos ramos diferentes. Seu caráter adaptativo permitindo remixar várias metodologias para melhorar um processo, produto, tarefa, afim de acelerar processos produtivos mantendo padrão de qualidade e produtividade por meio de métodos ágeis. Essa característica determina que as metodologias ágeis sejam evolutivas, e sempre se renovem.