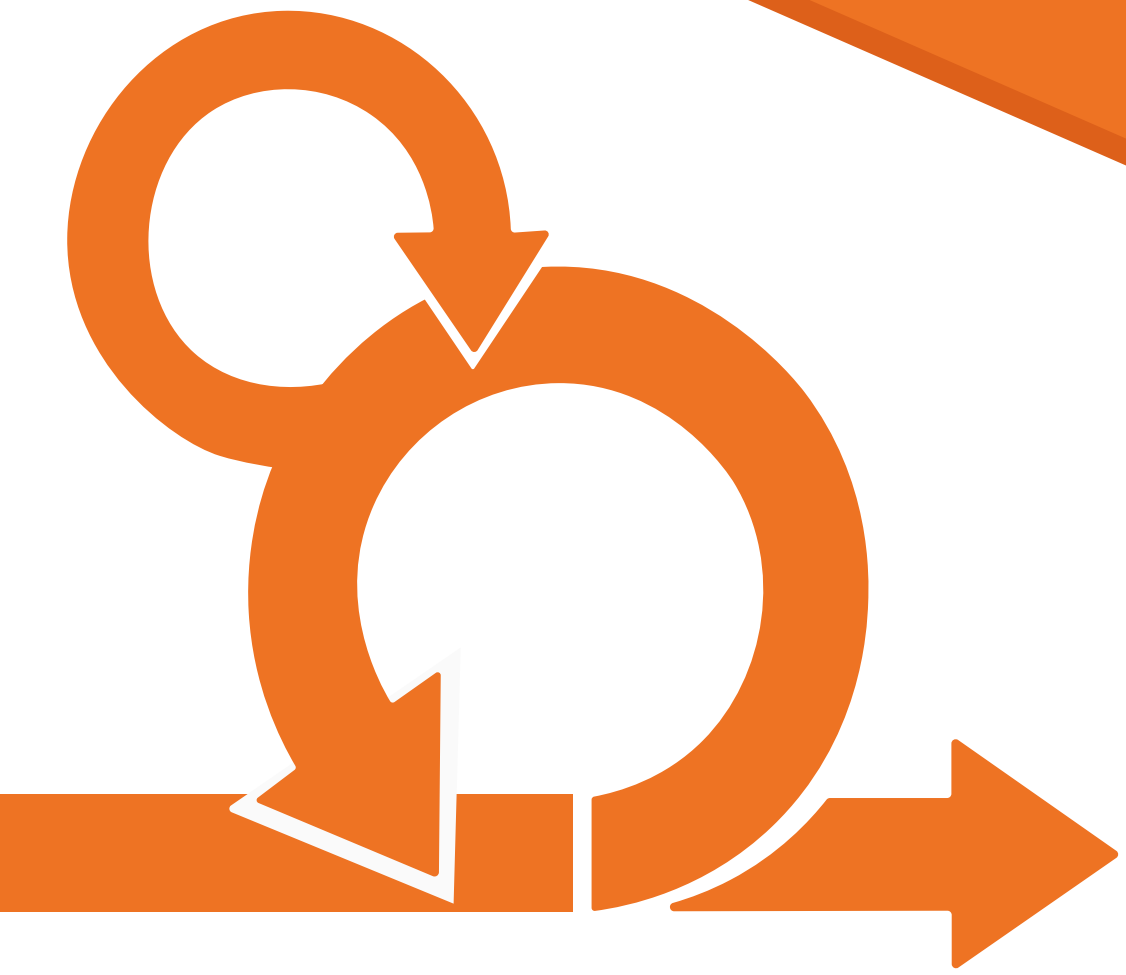
# RESUMO DO SCRUM



# Sumário

# SOBRE O AUTOR

Luiz Fernando Duarte Júnior é Bacharel em Ciência da Computação pela Universidade Luterana do Brasil (ULBRA, 2010) e Especialista em Desenvolvimento de Aplicações para Dispositivos Móveis pela Universidade do Vale do Rio dos Sinos (UNISINOS, 2013).

Carrega ainda um diploma de Reparador de Equipamentos Eletrônicos (SENAI, 2005), nove certificações em Métodos Ágeis de desenvolvimento de software por diferentes certificadoras (PSM-I, PSD-I, PACC-AIB, IPOF, ISMF, IKMF, CLF, DEPC, SFPC) e três certificações de coach profissional pelo IBC (Professional & Self Coach, Life Coach e Leader Coach).

Atuando na área de TI desde 2006, na maior parte do tempo como desenvolvedor, é apaixonado por metodologias ágeis desde que teve o primeiro contato com liderança e gestão de projetos em 2010.

> **Foi amor à primeira vista e a paixão continua a crescer!**

De lá para cá teve oportunidade de liderar times ágeis em diferentes empresas, de startups com 8 funcionários a bancos com mais de 3500 pessoas e mais de 1 milhão de clientes, liderando transformações ágeis e capacitando centenas de profissionais por ano seja com seus cursos ou no exercício da sua profissão. Um grande entusiasta de tais metodologias, espera que com esse livro possa ajudar ainda mais pessoas a terem seu primeiro contato com o assunto e assim despertar o interesse delas em aprender mais, para aumentar a competitividade das empresas brasileiras.

Há alguns anos que atua como Agile Coach e é autor do blog www.luiztools.com.br, onde escreve regularmente sobre métodos ágeis e desenvolvimento de software, bem como mantenedor da página LuizTools no Facebook, Twitter e Youtube com o mesmo propósito.

Entre em contato, o autor está sempre disposto a ouvir e ajudar seus leitores.

**Aproveita e me segue nas redes sociais:**

Facebook.com/luiztools

luiztools.com.br/youtube

Twitter.com/luiztools

LinkedIn.com/in/luizfduartejr

**Conheça meus outros livros:**

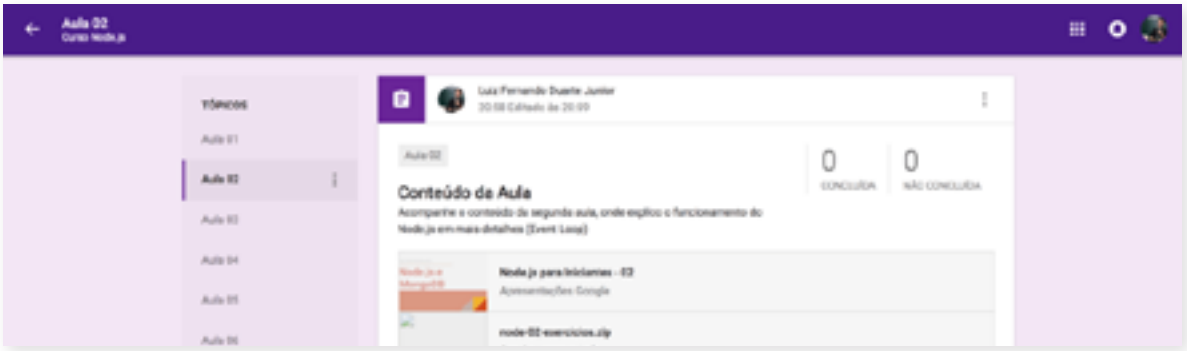

Programação Web com Node.js


MongoDB para Iniciantes


[Node.js e Microservices]


[Scrum e Métodos Ágeis: Um Guia Prático]

**Meus Cursos:**




[Node.js e MongoDB]


[Scrum e Métodos Ágeis]

# UM GUIA RÁPIDO PARA O SCRUM

Conheço e aplico métodos ágeis há pelo menos 9 anos na data em que escrevo este resumo, principalmente o framework empírico Scrum.

Durante o final da minha faculdade a empresa onde trabalhava, RedeHost, necessitava urgente de uma reorganização dos seus processos de desenvolvimento de software e queriam que eu me tornasse um de seus líderes de desenvolvimento. Obviamente agarrei a oportunidade, que foi muito importante para minha carreira, e isso me rendeu uma ida à São Paulo participar da segunda turma de um treinamento latino-americano oficial da Scrum.org para desenvolvedores Scrum. O treinamento de uma semana foi incrível e me capacitou para tirar duas certificações na área de gerenciamento de projetos com métodos ágeis e passei a atuar como facilitador do framework Scrum na empresa, em treinamentos e em consultorias.

Neste ebook eu resumo e comento o famoso Scrum Guide, o livro de apenas 19 páginas que detalha tudo o que você precisa saber sobre o Scrum antes de passar a aplicá-lo em seu time de produto.

## O que é Scrum?

A definição formal, cunhada pelos criadores Ken Schwaber e Jeff Sutherland, diz que Scrum é ...

“Um framework dentro do qual pessoas podem tratar e resolver problemas complexos e adaptativos, enquanto produtiva e criativamente entregam produtos com o mais alto valor possível.”

Entenda um framework como uma caixa de ferramentas, uma estrutura básica na qual você constrói algo em cima. Scrum não é algo pronto que resolve todos seus problemas, mas sim um conjunto de processos que norteiam o desenvolvimento de produtos complexos.

Apesar de leve e simples de entender, Scrum é extremamente difícil de dominar, uma vez que envolve muita disciplina em sua aplicação. Implementações de Scrum que não seguem as regras previstas em seu manual são o que chamamos de Scrum flácido, e tendem a não alcançar o potencial máximo da metodologia. Não trabalhei em muitas empresas que usassem Scrum como metodologia (embora as melhores usassem) então não sei dizer se alguma empresa no mundo segue o manual à risca, sempre há alguma “flacidez” nas implementações por aí.

Mas por que confiar e seguir cegamente este framework?

Scrum é um processo empírico, baseado pura e simplesmente nas experiências passadas de seus criadores que desde o início da década de 90 aplicam Scrum em projetos de software com sucesso, já sendo uma das metodologias ágeis mais usadas no mundo. E uma vez que você se torna adepto do Scrum, o empirismo passa a fazer com que você respeite o processo cada vez mais e adicione novos processos e artefatos complementares baseados nas SUAS experiências com o mesmo.

Ou seja, é um processo que evolui não só com o passar do tempo, mas com as suas experiências. E isso não é algo inventado pela comunidade Agile, estava previsto desde o início através dos pilares do Scrum.

## Os Pilares

O Scrum se baseia em três pilares de sustentação, suas ideologias e suas principais qualidades, na minha opinião:

### Transparência

Aspectos significativos do processo devem estar visíveis aos responsáveis pelos resultados. Isso vale para dashboards, métricas, resultados, metas. No cenário ideal, todos sabem o porquê de cada tarefa que realizam, sabem qual a sua contribuição para o objetivo maior, e qual é esse objetivo.

### Inspeção

Os usuários Scrum devem, frequentemente, inspecionar os artefatos Scrum e o progresso em direção a detectar variações. Esta inspeção não deve, no entanto, ser tão frequente que atrapalhe a própria execução das tarefas. Sabe aquele ditado da da administração: "O que não é medido não é gerenciador"? Pois é, este pilar é exatamente isso. Como espera que sua equipe melhores os processos continuamente sem inspeção?

### Adaptação

Se um ou mais aspectos de um processo desviou para fora dos limites aceitáveis ou produto não será inaceitável, o processo ou o material sendo produzido deve ser ajustado. Novamente, de nada adianta ter objetivos claros e inspeção frequente se nenhuma ação é tomada em prol de melhorias?

Note que estes três pilares se retroalimentam: após a adaptação, novamente voltamos para a transparência (em relação ao que foi adaptado) e à inspeção, para verificar se as adaptações estão melhores que antes ou se há novas correções à serem feitas. Este círculo virtuoso lembra, e muito, o ciclo Construir-Medir-Aprender do Lean Startup, metodologia muito utilizada por empresas inovadoras para construir produtos tecnológicos de alto crescimento.

Sobre estes pilares o framework Scrum divide-se em papéis, eventos e artefatos.

## Papéis

Time Scrum é o nome dado ao conjunto de pessoas que possuem todo o conjunto de skills necessárias para desenvolver um produto de software. No Scrum, ao contrário de metodologias de desenvolvimento mais tradicionais como o RUP, temos pouquíssimos papéis. Poucos mesmo. Em um Time Scrum (sim, com T maiúsculo) não temos diferenciação de papéis específicos como testador ou designer e nem mesmo hierarquia como gerentes ou diretores. Temos pessoas comprometidas em fazer um trabalho bem feito, no escopo e prazo combinado.

Todo Time Scrum deve incluir todos os papéis descritos a seguir, e somente eles:

### Product Owner

É o responsável pelo ROI (retorno sobre o investimento) do projeto, por gerenciar e priorizar o Product Backlog (veja mais adiante) e, para que o Scrum funcione realmente e ele consiga trabalhar, toda a organização deve respeitar as suas decisões. No mundo mais tradicional da gestão de projetos, o P.O. seria o Gerente de Produto (Product Manager), aquele que tem a visão, que define o roadmap, que colhe os insights de mercado e os transforma em requisitos para os desenvolvedores trabalharem.

O P.O. é membro do Time Scrum e deve estar presente e disponível para solucionar dúvidas do time em relação ao produto e ao backlog. Se esse papel for assumido por alguém da empresa do cliente, esta pessoa idealmente deve ter um canal de comunicação ininterrupto com o time de desenvolvimento, para que ele possa avançar sem atrasos.

Um P.O. ausente geralmente leva a requisitos pouco definidos, features "meia-boca" e insatisfação do cliente com a entrega gerada. O mesmo vale para Product Owners incompetentes em sua função, mas isso é outra história...

### Desenvolvedores

São os responsáveis por desenvolver o produto. O time de desenvolvedores (que não precisa e não deve ser composto apenas por desenvolvedores) é autogerenciável, multifuncional e compartilham a responsabilidade pelo sucesso ou fracasso do projeto. Aqui não há espaço para duelo de egos ou "eu faço apenas o design", quando o time falha, é culpa de todos e isso deve ser reafirmado em todas reuniões, afinal, eles são um time.

O time de desenvolvimento é quem decide o quanto consegue entregar a cada iteração, para que haja comprometimento com a entrega. O time de desenvolvimento é quem decide como será feita cada feature, pois cabe aos seus membros a capacidade técnica para executar o projeto.

### Scrum Master

É o responsável por aplicar e garantir a adoção do Scrum dentro da equipe e até mesmo dentro da organização onde estão inseridos. Cabe ao Scrum Master, que é um líder-servidor, liderar o time para que os objetivos do Product Owner sejam alcançados e para que o time de desenvolvimento consiga avançar sem impedimentos, removendo-os quando necessário.

Todos os eventos Scrum são facilitados pelo Scrum Master da equipe, que idealmente não deve ter outra função paralela no time Scrum. Se gerenciar os processos e remover impedimentos não tomar tempo suficiente do seu Scrum Master, é porque ou o projeto é muito pequeno ou o Scrum está sendo adotado de maneira errônea.

## Eventos

O Scrum chama seus eventos de time-boxes, uma vez que são eventos de duração fechada. Um evento pode ser encerrado em tempo menor do que o previsto, mas nunca maior, e o Scrum Master deve garantir isso enquanto facilitador dos eventos. Permitir a extrapolação cria um sentimento inconsciente de impunidade e faz com a equipe não respeite os objetivos e o planejamento da Sprint. Por sua vez, falhas

constantes em entregar os objetivos de uma Sprint forçarão o time de desenvolvimento em serem mais eficientes no planejamento, focarem mais nos objetivos e se comprometerem de maneira mais coerente com as entregas.

Lembre-se: time-box!

### Sprint

As sprints são time-boxes de 1 mês ou menos e são o coração do Scrum. Durante o período da Sprint um incremento utilizável do produto é criado. Mas nem só de desenvolvimento vive a Sprint, fazendo parte da mesma também o planejamento, as reuniões diárias, a revisão e a retrospectiva.

Geralmente recomenda-se que times iniciantes no Scrum comecem com Sprints pequenas, como uma semana ou 15 dias, para só então irem aumentando até chegarem no valor ideal: um mês. No entanto, evite que, após esse período inicial de adaptação, as sprints tenham tamanho variável ou isso irá criar confusão no time. Sempre faça com que o time tenha que estimar as tarefas com o tempo que possuem, e não com o tempo que gostariam. Isso pode soar um pouco tirano, mas dê mais tempo à uma equipe e ela se tornará cada vez menos eficiente pois terá margem para postergação.

### Sprint Planning

Time-box de 8h para uma sprint de um mês, ou menos tempo de acordo com o tamanho da Sprint. Nesta reunião é onde o Product Owner é ouvido em relação às prioridades e os objetivos desta Sprint. É nela também onde o time irá deliberar sobre o que conseguem fazer nesta sprint em relação às necessidades do P.O., formalizando o Sprint Backlog, ou lista de coisas que serão feitas no próximo mês.

É comum o uso de artefatos externos ao Scrum nesta fase, como o Scrum Poker, usado para estimar o custo de tarefas a serem realizadas, como veremos em um capítulo posterior. O ideal é que independente da técnica usada para estimar (pontos de função, por exemplo), que cada tarefa não ocupe um membro do time por mais de um dia. Se isso acontecer, veja como pode quebrar a tarefa grande em tarefas menores.

Outro artefato muito popular também é o Kanban (especialmente usando Trello, como veremos em um capítulo mais adiante), um board

onde temos colunas com cards, onde em cada card temos uma tarefa a ser realizada, previamente estimada. Cada coluna representa um estágio do ciclo de desenvolvimento de uma tarefa, tais como: TODO (para fazer), DOING (fazendo), Testing (testando) e DONE (pronto).

O Scrum Master deve cuidar para que os participantes do Sprint Planning não desviem do objetivo principal da próxima Sprint, traçado pelo PO, e que não percam tempo estimando software que não será desenvolvido na próxima sprint. Caso algum requisito necessite de pesquisa e desenvolvimento de uma prova de conceito, não estime ele como sendo uma tarefa conhecida, ao invés disso inclua a tarefa de P&D nesta sprint e idealmente deixe o requisito para a próxima.

### Daily Scrum

Timebox de 15 min que deve acontecer diariamente, sempre no mesmo local e horário para gerar consistência e evitar perda de tempo, facilitada pelo Scrum Master. Nesta reunião, que deve ser muito dinâmica e que popularmente é feita em pé (para evitar prolongamentos e distrações), cada membro do time deve responder apenas três perguntas: o que eu fiz ontem, o que eu vou fazer hoje e se tem algo me impedindo.

A reunião diária é o método mais eficaz de alinhamento dos membros do time que tive a oportunidade de experimentar. Ela permite a descoberta rápida de problemas e desvios de curso e suas respectivas correções, por isso deve ser respeitada.

Artefatos passíveis de serem usados nessas reuniões incluem um Burndown Chart, que nada mais é do que um gráfico de features a serem desenvolvidas (eixo y, vertical) versus tempo da Sprint (eixo x, horizontal). O gráfico começa no alto do eixo y (ou seja, faltam todas features) e no ponto 0 do eixo x (ou seja, ainda tem todo o tempo da sprint sobrando) e deve ser atualizado diariamente com o progresso do time. Um Burndown Chart sempre visível garante o dia inteiro, todos os dias, que todos sabem o quanto falta para terminar a sprint, e caso estejam longe do seu objetivo, de resolverem os problemas. A ideia aqui não é analisar o Burndown Chart na reunião diária, que é curta e deve se manter assim, mas apenas acompanhar rapidamente, também de maneira diária, o progresso do time. Falaremos mais dele no capítulo apropriado

Esse evento é um dos eventos que representam o pilar de inspeção do Scrum.

### Sprint Review

Time-box de 4h para sprints de um mês onde o incremento do produto, que está pronto para uso, é apresentado ao Product Owner (e demais stakeholders) para apreciação. Também é na review (que deve ser facilitada pelo Scrum Master) que o Product Owner apresentará os números, gráficos e tudo o mais que for importante à equipe saber sobre o produto. Novas prioridades, movimentos do mercado, etc, tudo focado em manter os objetivos coerentes ao longo das sprints.

Essa é o evento que melhor representa o pilar de inspeção do Scrum e caso o Time tenha sido bem sucedido, é importante que todos sejam parabenizados.

### Sprint Retrospective

Time-box de 3h para sprints de um mês onde o time de desenvolvedores e o Scrum Master (que atua apenas como facilitador) falam sobre os resultados obtidos na Sprint que passou e as lições tiradas a partir daí, para melhorar o processo, fortemente arraigado ao pilar de adaptação.

Como adaptação é um dos pilares-chave do Scrum, dediquei um capítulo inteiro deste livro a lhe ensinar como conduzir retrospectivas de Sprint que realmente funcionam.

## Artefatos

O Scrum determina alguns poucos artefatos, que são ferramentas para lhe auxiliar na aplicação da metodologia. No entanto, noto que cada vez mais equipes adotam artefatos adicionais para suprir necessidades individuais ou coletivas de quem adota esta metodologia ou qualquer outra metodologia ágil em geral. Boa parte dos capítulos seguintes deste livro são sobre artefatos não contemplados no Scrum Guide mas que adicionam um enorme valor à adoção de métodos ágeis em equipes de software.

Mas antes, vamos conhecer os artefatos originais:

### Product Backlog

Lista ordenada por prioridade de tudo que deve ser necessário no produto, e origem única dos requisitos para qualquer mudança a ser feita no mesmo. Gerenciado única e exclusivamente pelo Product Owner, que o faz a todo momento, o product backlog deve ser mantido longe do time de desenvolvimento, para evitar dispersão pensando no futuro.

Muitas empresas têm usado a ferramenta Trello para seus products backlogs e é bom ter em mente que ele nunca está pronto de verdade. Variáveis como mercado, concorrência, novas tecnologias e modelos de negócio inovadores tendem a fazer com que o Product Backlog tenha de ser alterado para que o próprio produto sobreviva. Essas e outras decisões cruciais acerca do produto cabe ao Product Owner tomar, baseado em inputs do cliente.

### Sprint Backlog

Uma versão reduzida de backlog apenas com os itens que devem ser desenvolvidos nesta Sprint, retirados do backlog original. Também pode ser organizada em formato de kanban (como veremos no capítulo apropriado) no Trello, mas em um board separado do Product Backlog.

Cabe ao time de desenvolvimento montar o Sprint Backlog com base nas prioridades do Product Owner e é sua responsabilidade dar cabo de todos itens até o fim da sprint, servindo o Scrum Master como um facilitador para que nada interrompa o ciclo de desenvolvimento.

Ao contrário do Product Backlog, o Sprint Backlog é imutável após sua construção e deve ser defendido pelo Time de Desenvolvimento e pelo Scrum Master, para garantir que o planejamento inicial, com suas metas, prazos e escopo, sejam realizados com sucesso. Caso isso não seja possível, o ideal é encerrar a Sprint neste exato momento, realizando review e retrospectiva, para então começar outra inserindo as mudanças emergentes.

## Considerações

É isso. O Scrum Guide tem apenas 19 páginas e esse resumo umas 2000 palavras. O Scrum é pequeno, mas difícil de ser posto em prática porque exige muita disciplina. Nos treinamentos que já ministrei aplico sempre um exercício prático aos alunos para enriquecer a experiência e vivenciar na prática, mesmo que por poucas horas, como funciona o ciclo de desenvolvimento de um produto usando Scrum.

# GLOSSÁRIO

Um pequeno glossário de termos oriundos do nosso universo ágil, em ordem alfabética.

## Agile

Termo usado comumente para designar as metodologias (ou métodos) ágeis de se desenvolver software, em oposição aos métodos tradicionais, formais e burocráticos.

## Agile Coach

Profissional responsável pela disseminação e melhoria contínua de processos ágeis dentro de uma organização. Dependendo da empresa ele pode ser um Scrum Master mais focado na estratégia do processo em alto nível, enquanto que sua contraparte é mais operacional. Em outras empresas pode ser apenas um Scrum Master "desvinculado" de um processo ágil específico (o Scrum), também referido algumas vezes como Agile Master.

## Agile Master

Ver Agile Coach.

## Aliança

Ver Alliance.

## Alliance

Nome dado a junção lógica, temporária ou permanente, de várias Tribes pela necessidade de uma grande entrega ou pela sinergia entre as mesmas. Algo como uma versão ágil de superintendências tradicionais. Popularizado pela empresa sueca Spotify.

## Artefato

Nome dado às técnicas e recursos físicos usados como apoio ao processos ágeis.

## Backlog de Produto

Lista de tarefas e recursos a serem desenvolvidos em um software.

## Backlog de Release

O mesmo que que Backlog de Produto, mas sendo um subconjunto de tarefas e recursos que devem ser realizados para cumprir o projeto de uma nova release do produto.

## Backlog de Sprint

O mesmo que o Backlog de Produto, mas sendo um subconjunto de tarefas e recursos que devem ser realizados na sprint atual.

## Burndown Chart

Gráfico que mostra a relação entre tarefas a serem realizadas e tempo disponível para realização das mesmas dentro de uma iteração, dando uma previsão se o prazo será cumprido ou não. Ver capítulo 6 para mais detalhes.

## Buy a Feature Game

Técnica colaborativa de priorização de roadmap onde os participantes usam dinheiro falso limitado para comprar as features. As features que receberem mais dinheiro são as mais prioritárias.

## Chapter

Dentro de uma Tribe (ver Tribe), transversalmente às Squads (ver Squad), os profissionais de mesma skill reúnem-se em chapters. Por exemplo, todos os programadores Java das Squads de uma Tribe formam o Chapter Java. Popularizado pela empresa sueca Spotify.

## Chapter Leader

O líder técnico de um chapter, a referência na tecnologia do chapter. Ver Chapter.

## Comunidade

Ver Guild.

## Condição-Alvo

Dentro do Improvement Kata da Toyota, uma condição-alvo é o próximo passo de curto prazo a ser alcançado rumo a uma visão de longo prazo.

## Daily Meeting

Reunião diária de 15 minutos para alinhamento do time, onde todos ficam sabendo o que cada um fez no dia anterior, o que fará hoje e se há algum impedimento a ser resolvido.

## Daily Scrum

Outro nome para o Daily Meeting.

## David J. Anderson

Autor da conversão do Kanban de chão de fábrica para o mundo do software.

## Definição de Preparado

Artefato Ágil para criar um conceito comum do que READY significa entre todos os membros do Time Scrum. Um item de backlog somente pode ser posto para TODO por um desenvolvedor se ele passar na Definição de Preparado.

## Definição de Pronto

Artefato ágil para criar um conceito comum do que DONE significa entre todos os membros do Time Scrum. Um item de backlog somente pode ser posto em DONE por um desenvolvedor se ele passar na Definição de Preparado.

## Definition of Done

Ver Definição de Pronto.

## Definition of Ready

Ver Definição de Preparado.

## Desenvolvedor

Dentro do framework Scrum é qualquer profissional que pertença ao time e que não seja o Scrum Master ou o Product Owner. Isso inclui designers, testers, programadores, etc.

## Desenvolvimento Orientado à Testes

Ver Test Driven Development.

## Extreme Programming

Metodologia ágil muito utilizada no mundo, talvez apenas não mais do que o Scrum.

## Futurespective

Técnica usada para alinhar o roadmap de um produto entre o time e até mesmo os stakeholders, onde colaborativamente constrói-se o roadmap baseado no entendimento pessoal e coletivo dos participantes do que é o mais importante.

## Grooming

Nome popular da cerimônia de Refinamento do Product Backlog, onde P.O. e Time de Desenvolvimento discutem e esclarecem os itens do Product Backlog antes da próxima Sprint Planning.

## Guia do Scrum

O guia oficial da metodologia Scrum, disponível gratuitamente na Internet pelos seus criadores. Ver capítulo 2 para mais detalhes.

## Guild

Nome dado a comunidades ágeis que possuam um interesse comum dentro de uma empresa, transversalmente às Tribes (ver Tribe).
Por exemplo, todos os profissionais de programação da empresa, independente da tribe, podem formar uma guild de Programação Segura. Popularizado pela empresa sueca Spotify.

## Head

O líder de uma Alliance.

## Improvement Kata

Técnica de melhoria contínua da Toyota para a busca de uma visão através da decomposição da mesma em condições-alvo menores e mais facilmente atingíveis.

## Iteração

Um ciclo fechado de desenvolvimento de software de curta duração, onde um conjunto limitado de tarefas serão realizadas e que resultarão em um incremento no produto final.
Ver também: Sprint.

Jeff Sutherland

Co-criador do Scrum.

## Kanban

Um artefato “roubado” da indústria tradicional, sendo um quadro com cartões que sinalizam o andamento das tarefas ao longo do pipeline de desenvolvimento de uma iteração. Ver capítulo 5 para mais detalhes.

## Kata de Melhoria

Ver Improvement Kata.

## Ken Schwaber

Co-criador do Scrum.

## Kent Beck

Criador do Extreme Programming (XP) e do Test Driven Development (TDD).

## Manifesto Ágil

Documento assinado por grandes nomes da engenharia de software mundial se comprometendo a melhorar os métodos pelos quais os softwares estavam sendo desenvolvidos até então, tornando os processos mais ágeis e menos burocráticos. Ver capítulo 1 para mais detalhes.

## Matriz Risco x Valor

Técnica ágil usada para priorizar roadmap e backlog usando um plano cartesiado de risco e valor dividido em quatro quadrantes, onde pode-se tomar decisões baseadas em diferentes estratégias.

## Minimum Viable Product

Ver Produto Mínimo Viável.

## MVP

Ver Produto Mínimo Viável.

## Next Target Condition

Ver Condição-Alvo.

## Pair Programming

Técnica ágil onde um programador mais experiente senta junto de um menos experiente para juntos realizarem uma tarefa do projeto de maneira mais eficiente. Ver capítulo 7 para mais detalhes.

## Peer Review

Técnica ágil onde um programador testar o software desenvolvido por outro programador, para garantir mais qualidade na entrega. Ver capítulo 7 para mais detalhes.

## Planejamento da Sprint

Reunião onde define-se o escopo de tarefas da próxima Sprint e estima-se o tempo de desenvolvimento das mesmas.

## Planning Poker

Artefato ágil utilizado para estimar tempo de desenvolvimento de software usando cartas de baralho em uma sequência de Fibonacci. Ver capítulo 3 para mais detalhes.

## P.O.

Ver Product Owner.

## Pontos de Caso de Uso

Artefato mais tradicional para estimar tempo de desenvolvimento de software usando pontos atribuídos à casos de uso conforme sua complexidade, mão de obra disponível, experiência prévia, etc.

## Pontos de Função

O mesmo que Pontos de Caso de Uso mas sendo uma estimativa mais granular, a nível de métodos e funções de código.

## Product Backlog

O mesmo que Backlog de Produto.

## Product Owner

Papel dentro de um Time Scrum semelhante ao tradicional Gerente de Produto, responsável pela visão do produto, gerenciamento do Product Backlog e retorno sobre o investimento (ROI).

## Produto Mínimo Viável

Protótipo do produto final com o mínimo de funcionalidades para validar interesse do mercado, precificação e funcionalidades importantes.

## Programação em Pares

Ver Pair Programming.

## Release Backlog

Ver Backlog da Release.

## Retrospectiva da Sprint

Reunião final de uma Sprint onde inspeciona-se os processos utilizados na última sprint e adapta-os visando melhoria contínua. Ver capítulo 7 para mais detalhes.

## Reunião Diária

Ver Daily Meeting.

## Revisão da Sprint

Penúltima reunião de uma sprint onde o time apresenta os resultados obtidos durante a última sprint para os envolvidos no projeto.

## Scrum

Método ágil mais usado no mundo para construção de produtos de software complexos. É um framework iterativo incremental. Ver capítulo 2 para mais detalhes.

## Scrum Guide

Ver Guia do Scrum.

## Scrum Master

Papel dentro de um Time Scrum semelhante ao de um Gerente de Projeto tradicional, mas mais alinhado com o perfil de líder-servidor, sendo suas principais funções garantir que os processos sejam seguidos e que o time avance rumo às metas sem impedimentos.

## Scrum Poker

Ver Planning Poker

## Sprint

É uma iteração, um ciclo dentro do processo do Scrum com geralmente 30 dias onde constrói-se um incremento de software pronto e entregável. Ver também: iteração.

## Sprint Backlog

Ver Backlog da Sprint.

## Sprint Planning

Ver planejamento da Sprint.

## Sprint Retrospective

Ver Retrospectiva da Sprint.

## Sprint Review

Ver Revisão da Sprint.

## Squad

Nome popular dado a times ágeis multidisciplinares, não necessariamente adotando Scrum. Popularizado pela empresa sueca Spotify.

## TDD

Ver Test Driven Development.

Team Leader

O mesmo que Tribe Leader.

## Test Driven Development

Metodologia de desenvolvimento de software criada por Kent Beck cujo foco é construir soluções partindo dos testes necessários, para depois codificar o funcionamento dos mesmos.

## Time de Desenvolvimento

Todos membros de um Time Scrum que não são Product Owner ou Scrum Master fazem parte do Time de Desenvolvimento. Ver também: Desenvolvedor.

## Time Scrum

O Time Scrum é formado por todas as pessoas necessárias para criar o incremento de software desejado pelo cliente, tendo um Product Owner, um Scrum Master e tantos Desenvolvedores quanto necessário.

## Time-box

Um evento de duração fechada e que faz parte dos processos do Scrum.

## Tribe

Nome dado a um conjunto de squads (ver Squad) que possuam uma proposta de valor em comum, que atuem em produtos/sistemas separados, mas para uma mesma frente da empresa. Popularizado pela empresa sueca Spotify.

## Tribe Leader

O gestor ágil de uma tribo. Responsável pela gestão de recursos, alinhamento estratégico com a empresa, formação das squads, etc. Ver Tribe.

## Tribo

Ver Tribe.

## XP

Ver Extreme Programming.

# SEGUINDO EM FRENTE

Este ebook termina aqui.
Pois é, certamente você está agora com uma vontade louca de aprender mais e criar apps incríveis que resolvam problemas das empresas e de quebra que o deixem cheio de dinheiro na conta bancária, não é mesmo?

Este livro é propositalmente pequeno, com menos de 50 páginas. Como professor, costumo dividir o aprendizado de alguma tecnologia (como Android) em duas grandes etapas: aprender o básico e executar o que foi aprendido no mercado, para alcançar os níveis intermediários e avançados. Acho que este livro atende bem ao primeiro requisito, mas o segundo só depende de você.

De nada adianta saber muita teoria se você não aplicar ela. Então agora que terminou de ler este livro, inicie hoje mesmo (não importa se for tarde) um projeto de app para ser vendido para uma empresa ou que atenda a um nicho de profissionais liberais. Caso não tenha nenhuma ideia, cadastre-se agora mesmo nas plataformas de freelancers que mencionei na primeira seção deste capítulo. Mesmo que não ganhe muito dinheiro sem seus primeiros projetos, somente chegarão os projetos grandes, bem pagos e realmente interessantes depois que você tiver experiência.

Me despeço de você leitor com uma sensação de dever cumprido. Caso acredite que está pronto para conceitos mais avançados, sugiro dar uma olhada em meu blog **http://www.luiztools.com.br**, em minha página do **Facebook.com/luiztools** e em meu outro livro **Criando apps para empresas com Android**.

Outros livros muito bons que recomendo são: **Google Android** do Ricardo Lecheta e **O Império dos Apps**, de Chad Mureta. O primeiro é a “bíblia do Android”, imenso e completo. Já o segundo é um livro de negócios com apps, para ganhar dinheiro pra valer!

Caso tenha gostado do material, indique esse livro a um amigo que também deseja aprender a desenvolver apps. Não tenha medo da concorrência e abrace a ideia de ter um sócio que possa lhe ajudar nos projetos.
Um abraço e até a próxima!

## Curtiu o Livro?

**Aproveita e me segue nas redes sociais:**

Facebook.com/luiztools

luiztools.com.br/youtube

Twitter.com/luiztools

LinkedIn.com/in/luizfduartejr

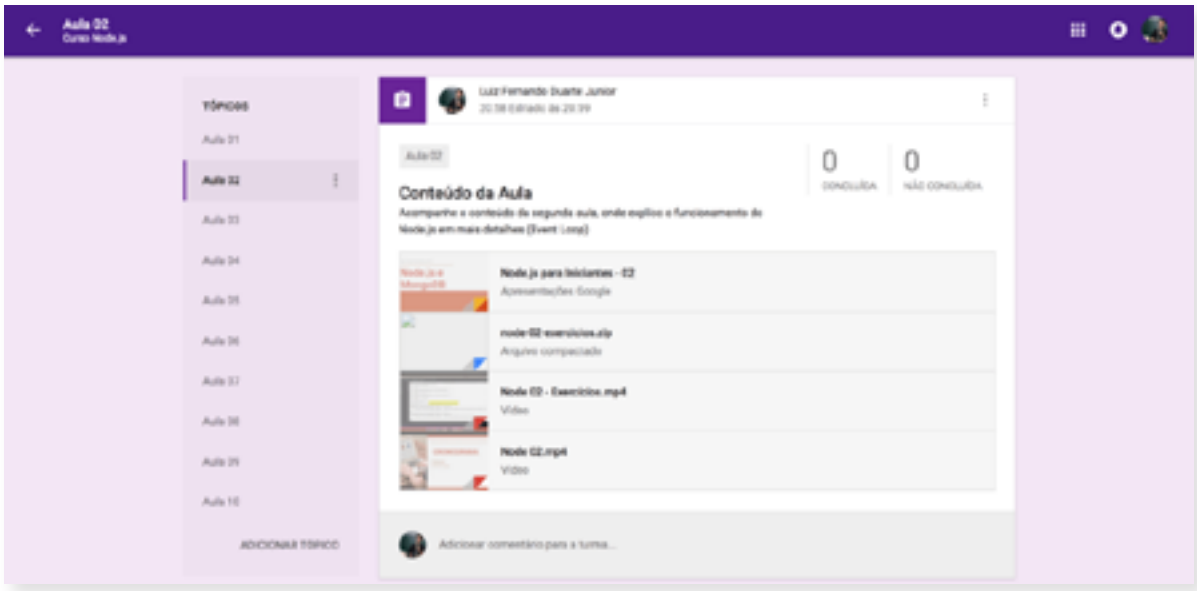
**Conheça meus outros livros:**


Node.js e Microservices


Criando apps para empresas com Android


[Scrum e Métodos Ágeis: Um Guia Prático]


[Java para Iniciantes]




[Node.js e MongoDB]


**Scrum e Métodos Ágeis**