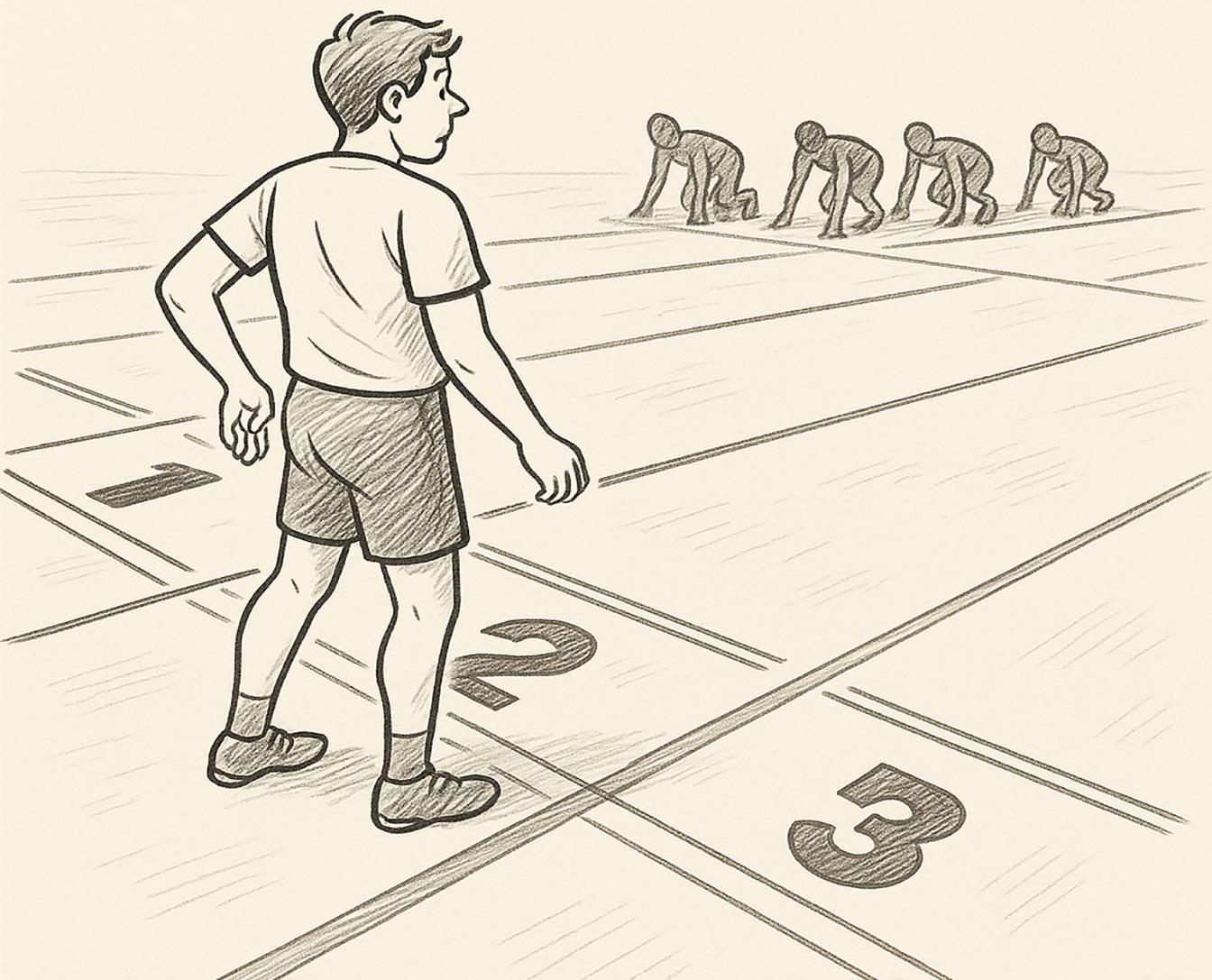


READY

WHY MOST SOFTWARE PROJECTS
FAIL AND HOW TO FIX IT



Luniel de Beer

Max Guernsey, III

Edição em Português

Partilhe Este Livro no Twitter!

Ajude Luniel de Beer e Max Guernsey, III divulgando este livro no [Twitter](#)!

O tweet sugerido para este livro é:

Acabei de comprar Ready — um livro para líderes de software e equipas que querem eliminar o retrabalho, o trabalho pendente e o desalinhamento na entrega de software.
[#CodeReady](#)

A hashtag sugerida para este livro é [#CodeReady](#).

Descubra o que outras pessoas estão a dizer sobre o livro clicando nesta ligação para pesquisar esta hashtag no Twitter:

[#CodeReady](#)

Ready (Edição em Português)

Por Que a Maioria dos Projetos de Software Fracassa
e Como Resolver

Luniel de Beer e Max Guernsey, III

Este livro está disponível em <https://leanpub.com/ready-pt-PT>

Esta versão foi publicada em 2025-10-22



Este é um livro da [Leanpub](#). A Leanpub capacita autores e editores através do processo de Lean Publishing. O [Lean Publishing](#) é o ato de publicar um ebook em desenvolvimento utilizando ferramentas simples e várias iterações para obter feedback dos leitores, fazer ajustes até ter o livro certo e criar tração quando o conseguir.

© 2025 Luniel de Beer e Max Guernsey, III

Em memória de Johann van Aardt, que reconheceu a minha paixão, me apresentou à verdadeira programação e me ajudou a encontrar o caminho para um novo lar. E aos meus pais, cujo apoio inabalável—desde o meu primeiro cliente até à minha primeira casa nos E.U.A.—tornaram tudo isto possível.

—Luniel

Para a minha família, com quem o sol nasce e se põe.

—Max

Conteúdo

Sobre Este Livro	i
Para Quem É Este Livro	ii
Como Usar Este Livro	iii
Sobre os Autores	iv
Prefácio	v
Parte I: Algo Está em Falta	1
Capítulo 1: O Problema Oculto	2
Capítulo 2: O Custo de Fundações em Falta	14
Capítulo 3: Introduzindo o Requirements Maturation Flow (RMF) (Fluxo de Maturação de Requisitos)	26
Capítulo 4: É Ágil?	28
Parte II: Criar Espaço para a Preparação	29
Capítulo 5: A Primeira Extensão	30
Capítulo 6: Por Que as Pessoas Não Fazem Isto?	31
Capítulo 7: Trabalho Explícito de Prontidão (RMF 1)	33
Capítulo 8: Efeitos do FMR 1	35
Capítulo 9: Pondo o RMF 1 em Prática	36

CONTEÚDO

Parte III: Controlo da Conclusão do Trabalho	38
Capítulo 10: A Próxima Necessidade	39
Capítulo 11: O Que as Pessoas Normalmente Fazem	41
Capítulo 12: Definindo uma Definition of Done	43
Capítulo 13: Definição de Pronto Personalizada (FMR 2)	45
Capítulo 14: A Vida Com RMF 1 e 2	49
Capítulo 15: Instalação do RMF 2	51
Parte IV: Implementação de Gating	53
Capítulo 16: O Requisito Final	54
Capítulo 17: Contexto sobre a Definição de Prontidão	56
Capítulo 18: Definindo uma Definição de Pronto	58
Capítulo 19: Definição de Pronto para Início Personalizada (RMF 3)	60
Parte V: Síntese	64
Capítulo 20: A <u>Maioria</u> dos Prazos Não Importa	65
Capítulo 21: Competência 1: Fluxo de Maturação de Requisitos	67
Capítulo 22: Como o Trabalho e a Informação Fluem no Scrum com RMF	69
Capítulo 23: O Impacto do FMR	71
Capítulo 24: Transição para RMF	72
Capítulo 25: Está nas Suas Mãoas	74
Parte VI: Recursos	75
Apêndice A: O Scrum Não É o Problema	76
Anexo B: The Synapse Framework™	78

Apêndice C: Objeções e Obstáculos Comuns ao RMF	1	80
Apêndice D: Listas Iniciais de Critérios DoD	81
Anexo E: Listas Iniciais de Critérios de DoR	82
Índice	83

Sobre Este Livro

Ready é um livro para qualquer pessoa envolvida no desenvolvimento de software que esteja cansada de **subentrega, retrabalho crónico e requisitos pouco claros**.

Pode já ter tentado investir em competências de execução da equipa, melhorar a implementação do seu framework de processo, ou renovar o código e ainda assim precisar de mais melhorias.

Isto acontece porque a principal limitação para a maioria das equipas de desenvolvimento de software não são as competências da equipa, mas sim a maturidade dos requisitos. **Mesmo equipas experientes com as competências adequadas continuam a ter dificuldades quando trabalham com requisitos imaturos.**

Ready apresenta o FMR (Fluxo de Maturação de Requisitos), uma abordagem prática e profundamente estruturada para alinhar Produto e Engenharia sem substituir o seu processo existente.

Quer esteja a usar Scrum, Kanban, ou algo personalizado, o FMR ajuda-o a **estabilizar o âmbito, eliminar trabalho pendente, e entregar o que realmente importa**.

Se as suas equipas se sentem estagnadas à beira do “quase pronto”, este livro vai mostrar-lhe como **quebrar o ciclo** e desbloquear a(s) sua(s) equipa(s) de vez.

Para Quem É Este Livro

Este livro é literalmente para qualquer pessoa envolvida no desenvolvimento de software. Desde engenheiros a gestores de produto e desde colaboradores individuais a executivos.

Este livro é para si se estiver envolvido no desenvolvimento de software e tiver notado que uma equipa com a qual trabalha ou na qual trabalha tem um ou mais dos seguintes problemas:

- O trabalho frequentemente transita de uma iteração para a seguinte
- As equipas de implementação sentem que estão a tentar atingir alvos em movimento
- O trabalho permanece em aberto durante demasiado tempo
- O trabalho é marcado como concluído mas as coisas não estão realmente prontas
- O trabalho realizado não corresponde às expectativas
- O trabalho gera regularmente uma grande quantidade de defeitos
- Grandes quantidades de trabalho têm de ser refeitas regularmente

Se algum destes problemas lhe é familiar, *Ready* pode ajudar.

Como Usar Este Livro

Este livro foi concebido para ser prático. Não é um tratado teórico ou uma apresentação estratégica—é um manual prático para implementar o FMR (o Fluxo de Maturação de Requisitos) baseado em trabalho real com clientes e testado em campo sob pressão real de entrega.

Os capítulos estão escritos em sequência, mas o próprio FMR é modular. É composto por três práticas fundamentais:

- FMR 1: Colaborar para um entendimento partilhado
- FMR 2: Controlar a conclusão do trabalho usando Definições Personalizadas de Concluído
- FMR 3: Controlar a implementação usando Definições Personalizadas de Pronto

Cada parte, ou “Hábito”, como lhes chamamos, funciona por si só, mas constroem-se uns sobre os outros. O livro foi concebido para o ajudar a abordá-los um de cada vez, por ordem. Essa estrutura reflete como recomendamos que as equipas adotem o FMR na prática—com cada Hábito sendo introduzido apenas depois do anterior estar a funcionar.

Isto evita sobrecarregar as equipas e dá a cada mudança a melhor hipótese de se manter. Aprenderá mais sobre como fazer isso começando em .

Se procura ajuda—seja aconselhamento, coaching, ou alguém para falar com a sua equipa de liderança—não hesite em contactar-nos diretamente.

E se procura apoio formal para implementar o FMR, a Producore oferece uma série completa de Programas concebidos para guiar a adoção passo a passo. Pode saber mais em <https:////ready-book.link/rmf>.

Sobre os Autores

Luniel de Beer é o criador do Fluxo de Maturação de Requisitos (FMR), um sistema prático para corrigir as lacunas entre a intenção do produto e a execução da engenharia. Tem mais de 15 anos de experiência a liderar transformações Ágeis, a fazer a ponte entre produto e engenharia, e a ajudar equipas a entregar com clareza e confiança.

Luniel também criou o sistema de Gestão de Capacidades da Producore, uma abordagem rastreável e escalável para modelar capacidades de produto. Ele concebeu o Desenvolvimento Orientado por PKB (PKBDD), um sistema versionado para gerir requisitos persistentes de produto. Estas ferramentas fazem parte de um framework de entrega mais amplo desenvolvido na Producore.

Max Guernsey, III é arquiteto de software, educador e co-fundador da Producore, uma consultora dedicada a corrigir falhas de entrega através de rigor estrutural e técnico. Com mais de duas décadas de experiência em design orientado a objetos, refatoração, desenvolvimento orientado a testes e padrões de design, ele tem tanto entregado sistemas críticos como treinado equipas de engenharia em larga escala. O seu trabalho combina práticas técnicas profundas com transformação comportamental e de processo para ajudar organizações a alcançar excelência sustentável na entrega.

Max contribuiu significativamente para o PKBDD e liderou o desenvolvimento da abordagem da Producore ao Desenvolvimento Orientado a Comportamento (BDD) através da sua profunda experiência em especificação comportamental.

Em conjunto, o seu trabalho integra clareza, rastreabilidade e controlo de qualidade num sistema coeso para entrega de software que se estende desde a prática da equipa até à capacidade organizacional.

Prefácio

Nota para Líderes de Engenharia

Se é um líder sénior numa organização de engenharia, provavelmente não lhe falta esforço, disciplina ou pessoas inteligentes. E, no entanto, de alguma forma, os projetos continuam a estagnar. Os objetivos derrapam. As expectativas não são cumpridas. Não porque as suas equipas sejam preguiçosas—mas porque algo fundamental está quebrado na forma como o trabalho é definido, moldado e entregue.

Este livro não é um guia de liderança. É uma ferramenta para diagnóstico estrutural. Revela o que está realmente a acontecer dentro das suas equipas—por que é que “quase feito” continua a transformar-se em “não feito”, e por que é que o progresso local raramente se traduz em resultados estratégicos.

Pode não se ver a *si próprio* nestas páginas. Mas se as suas equipas não conseguem entregar o que precisa, verá *elas*. E quando isso acontecer, finalmente terá a linguagem—e o sistema—para o corrigir.

De Luniel

Antes de mais, este livro não teria sido possível sem o Max, cuja capacidade de ver através do nevoeiro e da palha e destilar uma ideia até à sua essência está absolutamente além de mim.

Como chegámos aqui?

Olhando para trás, penso que é porque sempre quis compreender como as coisas realmente funcionam. Fosse religião, nutrição ou desenvolvimento de software, continuava a deparar-me com o mesmo problema: respostas superficiais que não se sustentavam sob pressão. Por isso continuei a escavar—perguntando não apenas o que fazemos, mas porquê, e o que falta quando não funciona.

Uma das primeiras fissuras no sistema surgiu num cargo onde eu usava três chapéus: Scrum Master, Product Owner e Gestor de Desenvolvimento (!!) para uma equipa que

entregava serviços de dados numa empresa de tecnologia bem conhecida. Estávamos a fazer o que o Scrum dizia—Sprints curtos, histórias num backlog, planeamento em meio dia—mas cada vez que começávamos um novo Sprint, encontrávamos atrito. A equipa não compreendia totalmente o problema, tínhamos de revisitar e rever requisitos durante o Sprint, dependências evitáveis surgiam e atrasavam-nos, e passos importantes eram esquecidos.

Então comecei a fazer algo diferente. Reunia a equipa e as partes interessadas numa sala para cada história, analisava o problema em detalhe, planeava a solução em conjunto, e só depois escrevia a história. O Planeamento de Sprint reduziu-se para uma hora, e o nosso sucesso nas entregas disparou.

Lentamente, comecei a perceber que o sucesso não vem de trabalhar mais arduamente dentro do Sprint. Vem da estrutura que se coloca em prática *antes* de ele começar.

Mais tarde, depois de ouvir Jeff Sutherland falar sobre “Definições de Prontidão”, sabia que havia algo valioso ali—mas não era suficiente. A minha experiência com requisitos, UX, UI, investigação, e mais tarde com BDD mostrou-me que diferentes itens de trabalho exigem diferentes tipos de prontidão. Alguns precisam de especificações de comportamento. Alguns precisam de acesso ao sistema. Alguns precisam de um rastreio completo de capacidades.

E todos eles precisam de um entendimento partilhado que seja realmente confirmado—não assumido.

À medida que trabalhava com mais equipas, via o mesmo padrão em todo o lado: passos em falta, dependências não satisfeitas, equipas a dar o seu melhor mas constantemente a lutar para corrigir problemas que deveriam ter sido prevenidos. Até as melhores equipas lutavam—não porque fossem fracas, mas porque lhes faltava uma estrutura que tornasse a prontidão explícita.

O resultado de toda esta aprendizagem, iteração e frustração é um sistema estruturado para gerir a prontidão.

É disto que trata este livro.

Espero que lhe dê clareza sobre onde residem os verdadeiros problemas e como corrigi-los. Espero que lhe dê linguagem para defender práticas que podem parecer “extra” mas são na verdade essenciais. E acima de tudo, espero que ajude as equipas a entregar com menos stress, menos surpresas e resultados muito melhores.

Se fizermos isto corretamente, pouparemos à indústria milhares de milhões de euros.

Mas mais importante, devolveremos às pessoas a sua sanidade.

De Max

Tenho trabalhado neste problema de vários ângulos durante décadas, mas o meu progresso estava estagnado até conhecer o Luniel.

Isto porque, antes de o conhecer, estava a abordar o problema, fundamentalmente, como sendo técnico. Estava focado em ajudar as equipas a adotar coisas como Desenvolvimento Orientado a Testes (TDD), refatorização, design de software avançado e, mais tarde, Desenvolvimento Orientado a Testes de Aceitação (ATDD) ou Desenvolvimento Orientado a Comportamento (BDD).

Na maioria desses casos, o problema abordado neste livro era tratado como um detalhe de implementação para estabelecer essas práticas técnicas.

Isto não quer dizer que já não valorize as práticas técnicas. Ainda acho que são profundamente importantes, mas não abordam diretamente o problema da *prontidão* no desenvolvimento de software. Em vez disso, *expõem* esse problema e depois as pessoas colocam um remendo no seu processo para o resolver “apenas o suficiente” para suportar as práticas técnicas que estão a tentar implementar.

Também quero abordar a questão de quem este livro pode ajudar. A resposta curta é “provavelmente quase todos no desenvolvimento de software”, mas a resposta real contém nuances que ajudam a mapeá-lo para vários ambientes sem alterar o significado básico.

Existem equipas que precisam da solução apresentada neste livro. Conhecerás uma versão simplificada de uma delas no Capítulo 1.

Existem também equipas que não **precisam** estritamente de um sistema como o que propomos, mas que ainda assim poderiam beneficiar dele.

A melhor equipa com que já trabalhei—facilmente uma unidade padrão de desvio acima da **melhor equipa seguinte**, se não duas—estava aninhada no interior do Oregon Central. Tinham um desempenho tão elevado que conseguiam superar a ausência de tal sistema através do puro volume de competência. No entanto, o meu gestor na altura, Tom Barreras, disse-me algo semelhante a “Reparei que as nossas histórias correm melhor quando dedicamos algum tempo a falar sobre os testes antecipadamente.”

Isto, mais uma vez, era algo que eu via na altura através da lente do desenvolvimento de testes e da execução técnica, mas agora sei que era mais um indicador de que a *prontidão* era um fator que afetava a equipa... aquela equipa em particular era tão capaz e rápida a responder que conseguia ter sucesso respondendo aos impedimentos à medida que estes surgiam, em vez de os prevenir à partida.

Mesmo que sejas o tipo de pessoa que não precisa estritamente de se preocupar com a prontidão porque consegues ultrapassá-la, ou trabalhas com uma equipa do mesmo género, ainda podes beneficiar do conteúdo deste livro.

Parte I: Algo Está em Falta

*Quando fazer as mesmas coisas melhor não ajuda, procure o que **não está a ser feito**.*

Capítulo 1: O Problema Oculto

Esta é uma história verídica¹ sobre um banco. Vamos chamá-lo simplesmente de “O Banco”. É um tipo de Instituição de Crédito Federada que serve como parte da infraestrutura financeira nacional dos Estados Unidos.

Nós (Luniel e Max) fomos chamados para O Banco porque este estava com dificuldades em entregar um projeto de software. Era um dos ambientes mais disfuncionais que já tínhamos visto, e é por isso que escolhemos este como o estudo de caso de abertura: se foi possível realizar uma mudança significativa n’O Banco, é possível em qualquer lugar.

Uma Nota Rápida sobre Projetos

Quando utilizamos o termo “projeto” neste livro, referimo-nos ao contexto da gestão de projetos. Embora existam diferentes ideias sobre o significado da palavra, estamos a utilizar a definição do [Project Management Institute](#):

“Um projeto é um empreendimento **temporário** realizado para criar um produto, serviço ou resultado único.”

Isto significa que um projeto tem um início e um fim definidos. Quando um projeto é encerrado, o conhecimento e os artefactos do projeto são arquivados, os membros da equipa são dispensados e os contratos são finalizados.

Neste livro, um projeto é fundamentalmente sobre execução. A maioria dos projetos, conforme definido pelo PMI, começa com a viabilidade ou design. A visão e a estratégia já ocorreram quando um projeto é iniciado.

Um projeto nasce dessa visão e estratégia, e tem sucesso ou falha com base na realização dos objetivos previstos—*não* se esses objetivos eram os corretos.

Pode usar a palavra “projeto” de forma diferente, e não há problema nisso. Apenas saiba que quando a usamos, estamos a referir-nos à definição e ao contexto acima.

¹Alterámos os detalhes identificadores para proteger a privacidade das pessoas e instituições envolvidas.

Nada disto pretende sugerir que aprovamos o uso da gestão de projetos para o desenvolvimento de software. Muito pelo contrário. Mas reconhecemos que é usado mesmo assim. Abordamos esse problema mais tarde, em [Capítulo 6](#).

1.1: A Reescrita Clássica

O Banco estava a reescrever o seu portal de pagamento de empréstimos por várias razões.

O sistema antigo, uma solução totalmente em C#/.NET, tinha muitos erros. Além de degradar a satisfação dos clientes, também gerava um fluxo incessante de incidentes de suporte muito dispendiosos, nos quais alguém tinha de manipular manualmente a base de dados para corrigir um erro cometido pelo sistema.

O sistema antigo também estava obsoleto do ponto de vista da manutenção. Era quase impossível para os engenheiros fazerem alterações significativas e, mesmo quando conseguiam, era uma proposta extremamente arriscada.

A reescrita deveria mudar isso.

O novo sistema ainda teria um backend em C#/.NET, mas este seria totalmente coberto por testes. O frontend seria implementado em OutSystems, uma popular solução low-code que permite a uma organização definir uma aplicação num único lugar e obter automaticamente uma aplicação web, uma aplicação Android e uma aplicação iOS sempre que decidem publicar as suas alterações.

A esperança era que o novo sistema estivesse livre de erros, melhorando tanto a satisfação dos clientes como reduzindo significativamente os custos de suporte.

Também esperavam que a reescrita desbloqueasse os programadores—com a combinação de uma abordagem mais disciplinada ao backend e a abordagem low-code ao frontend reduzindo significativamente o custo e o risco de novas funcionalidades.

Um efeito secundário positivo da mudança para OutSystems era que obteriam uma aplicação móvel limpa e moderna em ambas as principais plataformas.

Esse era o sonho quando começaram três anos antes do início desta história. A realidade era que, até agora, as equipas não tinham entregue **nada**.

1.2: Perspetivas sobre o Problema

Quando falámos com a equipa de liderança executiva, ouvimos uma frustração muito natural pelo facto de terem feito investimentos tão grandes sem absolutamente nenhum avanço estratégico.

Tinham tentado tudo, na sua perspetiva. Tinham mudado pessoal, aumentado pessoal, alterado orçamento, aumentado a pressão e trazido um desfile de consultores (dos quais foi fortemente sugerido que éramos os últimos). Nada parecia melhorar—pelo menos não de uma forma que pudessem medir, porque tudo o que viam era que o “ponteiro” estava no zero num trimestre e continuava no zero no trimestre seguinte.

Não queriam mais “progressos invisíveis”. Queriam *resultados*.

Quando falámos com a gestão dentro da organização de Produto, obtivemos uma história ligeiramente diferente (mas ainda similar) porque eles trabalhavam mais diretamente com a Engenharia.

Não é que as equipas não fizessem nada, pelo menos não para elas. É que as equipas nunca faziam o que lhes era pedido. Era praticamente garantido: não importava quão simples fosse o pedido nem quão claramente fosse apresentado, no momento de avaliar o que as equipas tinham feito, acabava-se sempre com algo completamente diferente.

Tinha chegado ao ponto em que a piada corrente era algo do género “Precisamos de descobrir como pedir aquilo que não queremos, para pelo menos termos hipótese de conseguir o que queremos.”

Os executivos de engenharia viam as coisas de forma bastante diferente.

Para eles, o problema era que o Product não estava a fornecer requisitos acionáveis e que não estava a fornecer requisitos *suficientes*. Se o Product pudesse simplesmente “alinhar-se com o programa”, as equipas seriam capazes de entregar o que queriam dentro do prazo e do orçamento.

Tinham feito investimentos sérios na modernização da forma como o código era escrito e entregue e, aos seus olhos, o Product não estava a fornecer requisitos claros.

Quando falámos com outros consultores (que nos referenciaram para a organização), eles identificaram corretamente a disfunção que estavam a ver: todos pareciam muito focados em culpar outra pessoa. A razão pela qual nos trouxeram inicialmente foi a preocupação com a estratégia de pessoal e queriam uma avaliação dos colaboradores individuais, mas eles consideravam que o facto de apontar o dedo e a mentalidade de capataz ao nível executivo eram a principal fonte do problema.

1.3: A Nossa Investigação

O nosso objetivo inicial era avaliar as equipas e tentar ajudá-las a melhorar as suas competências, se necessário, por isso começámos a analisar as pessoas que trabalhavam na linha da frente.

Havia definitivamente margem para melhorias.

Os colaboradores individuais do lado do Product não tinham realmente as competências necessárias. Na realidade, eram maioritariamente gestores de projeto que tinham sido colocados no papel de Product Owner (PO) ou Product Manager.

Como resultado, metade deles escrevia requisitos “vagos” e depois aceitava (literalmente) qualquer coisa que as equipas fizessem nessa iteração sem qualquer análise crítica. A outra metade escrevia o mesmo tipo de requisitos e depois alegava que as equipas “deveriam ter sabido” coisas que nunca tinham sido mencionadas e mantinha os itens de trabalho em aberto mais ou menos indefinidamente.

O banco costumava gerir e acompanhar o seu backlog de trabalho. Embora o que abordamos neste livro seja maioritariamente ortogonal ao Scrum, utilizamos a terminologia Scrum ao longo do texto porque a maioria - ou pelo menos uma pluralidade - das equipas usa Scrum.



Definição: Scrum

O Scrum é uma estrutura leve que ajuda pessoas, equipas e organizações a gerar valor através de soluções adaptativas para problemas complexos. Em resumo:

1. Um Product Owner ordena o trabalho para um problema complexo num Product Backlog
2. A Equipa Scrum transforma uma seleção do trabalho num incremento de valor durante um Sprint
3. A Equipa Scrum e as suas partes interessadas inspecionam os resultados e ajustam para o próximo Sprint
4. Repetir

Se não estiver familiarizado com o Scrum e a sua terminologia, recomendamos que reveja a versão 2020 do [Scrum Guide](#). É uma leitura rápida e esclarecedora.

Da mesma forma, descobrimos que as equipas técnicas estavam muito abaixo do típico em termos de competências de programação (-2σ, na melhor das hipóteses) e, além disso,

eram muito resistentes à mudança. Como consequência natural, a qualidade do código era péssima.

No entanto, segundo as equipas, não era esta a razão pela qual não estavam a cumprir. Para elas, eram os requisitos vagos e as alterações durante o Sprint por parte do Product que estavam a matar o projeto.

...e ninguém sequer falava do problema maior, aquele tão absurdo que parece inventado até o termos vivido.

As equipas de engenharia tinham o hábito de não compreender um requisito, construir algo aleatório e depois exigir crédito por terem “terminado um item de trabalho”.

Não nos referimos a uma pequena má interpretação. Referimo-nos a uma desconexão total: pedíamos-lhes para desativar a aplicação de fundos ao principal em determinadas circunstâncias, e eles desativavam a capacidade de adicionar um endereço de email secundário de confirmação.

Depois diziam-nos que era isso que tínhamos pedido.

1.4: Aprofundando a Investigação

Ambas as lacunas de competências podiam ser resolvidas, mas estávamos céticos de que fossem os verdadeiros obstáculos.

Havia algo mais errado, por isso aprofundámos a investigação. Começámos com esta questão: Por que razão a redação dos requisitos demorava tanto tempo e produzia resultados tão maus?

Uma das razões é que o conhecimento necessário para escrever um requisito significativo era extremamente escasso. Uma pequena parte estava na posse das equipas de Engenharia e Product.

Parte estava incorporada no código do sistema legado. Parte tinha desaparecido completamente. A maior parte, no entanto, estava armazenada como conhecimento tribal nos especialistas no assunto espalhados pelas várias unidades do banco. Isto significa que construir um requisito que efetivamente avançasse um objetivo estratégico era uma atividade extremamente trabalhosa e demorada.

Em contraste com isto estava um apetite insaciável por funcionalidades de uma equipa de liderança faminta por funcionalidades. O mandato era “mantenham os engenheiros a trabalhar—encham-nos de requisitos”. O foco estava na quantidade de requisitos para

manter as equipas ocupadas—uma perspetiva antagónica ao cuidado e tempo necessários para definir um requisito que “movesse a agulha”.

1.5: Melhorar as Coisas Não as Tornou Melhores

Todos estes são problemas que podem ser resolvidos, mas resolvê-los não ajudou.

Melhorias anteriores nas técnicas de desenvolvimento de software não tinham ajudado as equipas a entregar, mas Max fez um esforço para ajudar as equipas a melhorarem mais.

Ele introduziu conceitos revolucionários das doutrinas de programação de meados do século XX como “não copies e coles esse código 27² vezes, coloca-o numa função e chama-a”. Esta sugestão por si só melhorou dramaticamente a qualidade do novo código e permitiu-lhes começar a melhorar a qualidade.

Estes e outros conselhos básicos de programação ajudaram-nos a escrever melhor código que poderiam manter mais facilmente no futuro.

...mas não ajudou a fazer avançar o projeto.

Do lado do Produto, Luniel conseguiu introduzir BDD e garantir que os Product Owners analisavam minuciosamente os requisitos antes de serem entregues às equipas.

Ele conseguiu que as equipas colaborassem e os usassem para avaliar se um Item do Backlog do Produto (PBI) estava realmente concluído.

²Isto não só não é hipérbole, como nem sequer é o pior caso. Num caso, havia quase cem duplicados exatos do mesmo algoritmo.



Definição: Item do Backlog do Produto (PBI)

Um Item do Backlog do Produto (PBI) é uma unidade discreta de trabalho no backlog do produto que representa uma potencial alteração, adição ou melhoria ao produto. Os PBIs podem assumir várias formas—funcionalidade, correção de erro, melhoria técnica, tarefa de investigação, etc.—e são definidos pela sua contribuição para o valor do produto.

Muitas equipas referem-se aos PBIs como “histórias” ou “histórias de utilizador”, mas o termo correto do Scrum é “Item do Backlog do Produto” ou “PBI”. Uma vez que um PBI é incluído numa Sprint, também faz parte do Backlog da Sprint. Por simplicidade e neutralidade, usamos “Item do Backlog do Produto” ou “PBI” para nos referirmos a qualquer item de trabalho que a Equipa Scrum está a gerir—seja ele chamado de Item do Backlog do Produto, Item do Backlog da Sprint (SBI), história de utilizador, história, item de trabalho ou item de backlog.

Isto acrescentou *clareza*, mas não criou *fluxo*.

Com a ajuda dos consultores parceiros que nos tinham trazido, conseguimos (temporariamente) aliviar a pressão absolutamente esmagadora colocada sobre as equipas e autores de requisitos.

Pode ter ajudado a criar um pouco de confiança, mas não produziu resultados tangíveis.

Começámos até a construir uma base de conhecimento que ajudava as pessoas a localizar o conhecimento de negócio necessário para escrever requisitos e identificava os locais onde havia lacunas nesse conhecimento.

Acelerou a escrita de requisitos, mas não fez o produto chegar ao mercado.

Após meses de interações, tínhamos ajudado a liderança a *ver* onde estavam, mas estavam longe dos seus objetivos. E não estavam a aproximar-se.

Estavam a preparar-se para voltar à sua antiga estratégia de “sobrecarregar” as equipas para garantir que estavam sempre ocupadas.

1.6: Um Processo de Eliminação

Teria sido fácil simplesmente desistir e dizer “isto é sem esperança”. Havia uma série de desculpas a que se poderia recorrer:

- A equipa de engenharia tinha poucas competências (tinha)
- Os autores de requisitos tinham o conjunto de competências errado (tinham)
- A liderança estava a esmagar as equipas com expectativas irrealistas (estava)
- A organização não tinha conhecimento crítico de negócio necessário para funcionar (não tinha)
- Os executivos não pareciam confiar uns nos outros (não confiavam)

Todas estas coisas eram verdade. No entanto, tinham sido feitas melhorias em todas estas áreas e nenhuma delas parecia melhorar o problema fundamental. Nenhuma delas ajudou as equipas de Engenharia, o Produto, a gestão ou a direção executiva a aproximarem-se dos seus objetivos.

...e aí está a pista para a solução: Todos os problemas listados anteriormente eram os problemas que as pessoas já conseguiam ver.



Se as variáveis que consegue ver não fazem diferença, tem de haver uma variável que **não vê** que faz.

O verdadeiro problema era aquele que ninguém sabia que existia.

1.7: À Procura do Verdadeiro Culpado

Na procura pelo verdadeiro culpado—aquele que realmente impedia o banco de alcançar os seus objetivos—precisávamos de começar por algum lado.

Um local razoável para procurar era ver o que os PBIs que falharam (a maioria deles) tinham em comum.

Começámos por eliminar as coisas que podíamos ver que não eram comuns porque variavam muito:

- Que parte do sistema: alguns PBIs afetavam apenas o backend, outros apenas o frontend, outros ainda afetavam ambas as partes
- Que equipa executou o trabalho—não parecia importar quem fazia o trabalho, havia uma alta probabilidade de falha
- Que PO criou o trabalho—o mesmo que acontecia com as equipas

Depois começámos a olhar para as coisas que eram comuns. A lista não era extensa, mas também não era curta:

- As equipas de engenharia
- Os product owners
- A liderança
- A cultura
- O ambiente de desenvolvimento
- As práticas de engenharia
- A técnica de autoria de requisitos
- O domínio (finanças)
- Os serviços dependentes

Muitos destes também podiam ser descartados imediatamente. As equipas, product owners, liderança, cultura e ambiente de desenvolvimento tinham sido todos recentemente melhorados sem impacto real na produção significativa. Tínhamos pessoalmente ajudado a melhorar as práticas de engenharia e de autoria de requisitos e confirmado que essas melhorias tinham vingado, mas ainda assim não estava a ajudar.

Difícilmente se pode culpar o domínio. As finanças são um dos tipos mais antigos de cálculos realizados na história documentada. É extremamente maduro. Além disso, outros bancos estavam a implementar software, demonstrando performativamente que a hipótese (obviamente forçada) de que os bancos simplesmente não conseguem fazê-lo estava errada.

Os serviços dependentes também não podiam ser culpados, já que estavam a ter tanta dificuldade em mudar como a iniciativa que estávamos a analisar...

...mas isso fez-nos pensar: E se começássemos a analisar as causas das falhas?

1.8: Dissecando as Sementes do Fracasso

Um PBI falhou em fazer avançar o produto porque a equipa fez, como costumava fazer, algo completamente aleatório e quase inteiramente sem relação com o pedido. Isso, obviamente, é um sinal de que não entenderam o item de trabalho. Portanto, a compreensão era uma grande candidata, mesmo que já tivéssemos trabalhado nisso quando os ajudámos a adotar o BDD.

Outro PBI falhou em fechar porque fizeram mal os cálculos. Isso é outra evidência de que a compreensão poderia ser o problema central.

Um terceiro item de trabalho que analisámos não foi realmente verificado adequadamente pela Product Owner—ela aprovou-o automaticamente quando a equipa disse que era hora de fechar. Isso meio que pressionou a nossa hipótese, mas ainda se podia argumentar que ela não entendeu como o item de trabalho se encaixava num plano de nível superior.

Talvez. Mais ou menos. Se apertássemos muito os olhos quando olhávamos para isso dessa forma.

Depois deparamo-nos com um PBI que não se encaixava de todo no padrão. A equipa parecia entender—embora não houvesse forma de verificar se realmente entendia. Mas não importava: nunca tiveram hipótese de ter sucesso ou falhar por si próprios porque encontraram uma dependência que precisava de ser atualizada e tiveram de adiar o seu trabalho por vários Sprints.

Mesmo que eles *não* entendessem o que eram supostos fazer, nunca tiveram hipótese com esse item do backlog, portanto a compreensão **não** era o problema nesse caso.

Um caso isolado, claro, não é suficiente para refutar uma causa raiz específica, mas aguçou a nossa curiosidade. Começámos a procurar outras evidências que contrariasse a hipótese.

E encontrámos. Havia itens de trabalho que:

- Falharam porque a equipa sabia que não entendia, mas ninguém conseguia localizar um especialista no assunto para resolver o problema
- Mudaram para uma experiência pior para o utilizador como solução alternativa ao funcionamento dos serviços upstream
- Tiveram de ser adiados porque as dependências upstream não estavam prontas
- Não puderam ser concluídos porque os testers não conseguiram recolher dados de teste a tempo
- Foram fechados mas tiveram de ser refeitos porque o pedido em si estava incorreto
- Falharam porque a equipa não percebeu quão complexo o código existente já era
- Simplesmente não foram estimados
- Foram massivamente subestimados³
- Mudaram durante o Sprint porque o PO finalmente obteve o conhecimento do domínio necessário
- Pareciam ter mudado depois do Sprint (da perspetiva da equipa) porque o PO e a equipa nunca concordaram sobre o seu significado

³Não queremos dizer apenas que se enganaram. Parecia que as equipas simplesmente colocaram o valor “3” em todos os campos de estimativa para uma série de PBIs.

A lista continua, mas é suficiente para esta história.

Pode-se estabelecer uma ligação de cada um destes casos com o “entendimento” de alguma forma—e certamente a falta de entendimento estava *envolvida* em muitos deles—mas isso não significa que a falta de entendimento tenha sido a causa... especialmente porque já tínhamos feito algum trabalho no entendimento compartilhado e isso não tinha ajudado muito.

Então percebemos. Faltava uma peça mais fundamental. Nos casos em que a falta de entendimento estava envolvida, essa era apenas a causa próxima.

A causa distante era muito mais ampla.

1.9: Um Ponto de Interseção

A única coisa que todos os PBIs que analisámos quanto ao modo de falha tinham em comum era isto: Todos foram **iniciados prematuramente**.

Quando um item de trabalho falha porque a equipa sabia que não compreendia o problema e não conseguia encontrar um especialista para ajudar a compreender, isso significa que a equipa iniciou um PBI sabendo que não compreendia o problema.

Quando um item do backlog tem de mudar para uma experiência pior devido ao funcionamento de um serviço upstream, isso significa que o item de trabalho foi iniciado sem realmente compreender o impacto do serviço upstream.

O adiamento porque uma dependência upstream não estava pronta no final significa que não havia garantia da sua prontidão no início.

...e assim foi com todos os outros casos: Os Testadores não tinham dados prontos ou não sabiam como obtê-los antes do início de um PBI, os requisitos não foram realmente verificados antes de serem entregues à equipa, o código não foi investigado antes do trabalho ser iniciado, a estimativa era insuficiente ou inexistente, faltava conhecimento do domínio e, claro, o entendimento compartilhado não foi verificado.

O problema, como se verificou, era que a implementação começava em itens de trabalho antes desses itens estarem *prontos*.



Na nossa experiência, a maioria dos itens de trabalho que falham na entrega fazem-no porque **não estavam prontos** quando a implementação começou.

Então, partimos para ajudar O Banco com isso.

Neste ponto, pode pensar que apenas dizer que os PBIs devem estar prontos seria suficiente. No entanto, acontece que não é assim tão fácil de implementar. “Comprar barato, vender caro” é uma ideia igualmente simples.

Há uma peça em falta que é necessária para pôr o bom conselho em prática.

1.10: A Grande Reviravolta

Encontrámos a peça que faltava no puzzle que os desbloqueou, que por sua vez, desbloqueou a iniciativa.

Quando concluímos esse compromisso, os engenheiros ainda estavam bem abaixo da mediana em termos de competência. Os POs ainda não tinham as competências adequadas. A cultura ainda não estava corrigida...

No entanto, o produto finalmente começou a avançar e acabou por ser lançado.

No final deste livro, saberá qual é a peça que faltava e o que é necessário para a colocar em prática. E será capaz de desbloquear organizações que parecem estar retidas por uma parede invisível.

Uma Nota Rápida sobre o Âmbito

Este livro é sobre um problema muito específico e generalizado: a falta de estrutura, clareza e maturidade na transferência entre o negócio e a engenharia. Assume que algo foi escolhido para implementação e concentra-se em garantir que o trabalho de construção acontece com entendimento compartilhado, prontidão e conclusão rastreável.

As técnicas neste livro não lhe dizem o que construir, por que construir ou como descobrir se é a coisa certa a construir. Se a sua organização carece de gestão de produto real ou ciclos de feedback significativos, não estamos a tentar resolver isso aqui. O que oferecemos em vez disso é uma forma de tornar essas lacunas mais visíveis e reduzir o custo de descobrir que estava errado.

Usado no contexto correto, esta solução traz fluxo, segurança e clareza. Mas como qualquer sistema, pode ser mal aplicado—especialmente quando usado isoladamente ou sem consciência.

Capítulo 2: O Custo de Fundações em Falta

Faz sentido dedicar algum tempo a perceber exatamente *quão* grave este problema pode ser para algumas organizações.

Descobrimos que existem três principais “categorias” de problemas que a falta de preparação cria:

- Entendimento compartilhado ausente ou incompleto entre e dentro das equipas de Produto e Engenharia
- Falta de controlo sobre qual é realmente o objetivo de um PBI e quando está verdadeiramente concluído
- Falta de critérios que determinem quando um item de trabalho pode iniciar a fase de implementação do seu ciclo de vida

Além disso, notámos que alterar estas situações pode ser bastante desafiante. Isto faz sentido: a mudança é difícil.

Os velhos hábitos são difíceis de morrer e os novos são difíceis de incutir. Na nossa experiência como consultores, notámos que é *extremamente* fácil as pessoas voltarem aos velhos hábitos e comparativamente difícil estabelecerem novos.

Portanto, é necessário ter um mecanismo que imponha os novos hábitos e desencoraje os antigos.

Para abordar isto, acreditamos que existe outro componente em falta de responsabilização e rastreabilidade.

2.1: O Puzzle sem a Imagem da Caixa

Já tentou montar um puzzle sem a imagem na caixa? É possível, mas é mais lento, mais frustrante e cheio de falsos começos.

Faz-se progresso, depois desfaz-se tudo. Questiona-se o que encaixa onde. Pensa-se que se está a trabalhar na mesma imagem, até se perceber que não.

É assim que o desenvolvimento de software frequentemente se sente.

O backlog está cheio. O Sprint está a decorrer. Todos trabalham arduamente.

Mas sem uma imagem partilhada do que estamos a construir, o alinhamento torna-se uma questão de sorte, não um sistema.

Sem clareza, até as melhores equipas sentem frustração, esgotamento e a sensação de que o seu esforço não é valorizado.

2.2: Um Velho Ditado e uma Dura Realidade

Existe uma razão pela qual tantas práticas em torno da autoria de requisitos, e até algumas práticas de engenharia, se focam intensamente em criar um entendimento compartilhado entre solicitantes e equipas de implementação. Nada semeia tanto o caos quanto engenheiros que não sabem realmente o que devem construir.



“Lixo que entra, lixo que sai” é um ditado, não um lugar-comum.

A língua portuguesa está cheia de autocontradições...

- Pode sancionar as ações de alguém. Talvez isso signifique que deu permissão prévia ou talvez signifique que está a condenar depois do facto.
- Pode limpar levemente algo mas, se esse algo for uma credênciça, significa que está a remover pó dela, enquanto que, se for em referência a um beignet, está a adicionar pó.
- Se suportar uma equipa, pode ser a razão pela qual essa equipa consegue continuar a funcionar ou pode ser a razão pela qual não conseguem avançar.

Auto-antónimos podem ser os exemplos mais marcantes, mas são apenas um tipo de ambiguidade. Algumas palavras não são apenas o seu próprio oposto, mas têm muitos significados alternativos possivelmente confusos.

“Neste clipe, ele cortou um cupão de um jornal e prendeu-o com um clipe ao papel na prancheta junto com os outros recortes enquanto um clipper ao fundo se movia a bom ritmo.”

E não é só o português. *Todas* as línguas naturais que conhecemos têm esta propriedade.

E, no entanto, são as únicas que temos para trabalhar quando especificamos requisitos.

Como resultado, quando uma equipa de engenharia não *confirma* que o seu entendimento de um requisito é o mesmo que o do solicitante, essa equipa está a depender da sorte. Ou seja, o melhor resultado possível é que escolheram a interpretação correta e o solicitante não muda de ideias pelo caminho.

Esse resultado está longe de ser garantido.

2.3: Alguns dos Resultados Comuns

Sem um entendimento compartilhado confirmado, as equipas correm vários riscos.

Em geral, o resultado mais frequentemente doloroso é a equipa simplesmente construir a coisa errada.

O momento em que descobrem isso pode ser controlado por vários atributos do seu processo. Por exemplo, uma implementação saudável de Scrum pode detetar esse tipo de mal-entendido muito cedo na execução, enquanto um processo Waterfall tem uma grande probabilidade de atrasar tal descoberta por meses.



Definição: Waterfall

Um modelo sequencial de desenvolvimento de software que se tornou generalizado no final do século XX. Foi primeiro representado num [artigo de 1970](#) por Winston W. Royce, que ilustrava o desenvolvimento como uma série de etapas em cascata—requisitos, design, implementação, testes, e assim por diante—cada uma alimentando a seguinte como uma cascata. Embora Royce tenha apresentado o modelo como um exemplo do que *não* fazer, a indústria adotou-o como um modelo para desenvolvimento em grande escala.

O Waterfall também é conhecido por agrupar trabalho semelhante em grandes fases consecutivas—uma característica referida como **desenvolvimento em grandes lotes**. Este agrupamento praticamente garante o **aprendizado tardio**: as equipas não recebem feedback sobre decisões anteriores até muito mais tarde no processo. Erros descobertos tarde são mais caros de corrigir. Os praticantes de Agile criticam o Waterfall por esta razão, favorecendo ciclos iterativos menores que permitem descobertas e correções de rumo mais cedo.

No entanto, em algum momento, uma equipa que trabalha com o entendimento “errado” (diferente, na verdade) de um requisito terá um acerto de contas com o requisito “certo” (também diferente, na verdade). Mais uma vez, a saúde da organização influencia a forma que esse acerto de contas toma e o impacto que tem, mas quase sempre acontece.

Na maioria dos casos, isto leva a alguma forma de retrabalho. O solicitante (geralmente a Gestão de Produto) terá de pedir alterações para passar daquilo que as equipas de implementação construíram para o que ele realmente queria.

Outra manifestação muito comum é o solicitante continuar a responsabilizar a equipa pelo seu entendimento original do que pediu.

As equipas podem facilmente interpretar isto como um Gestor de Produto a mudar de ideias. Pior ainda, pode realmente *convidar* as partes interessadas a adquirirem o hábito de mudar de ideias—retendo itens de trabalho até que tudo esteja *perfeito*, esgotando as equipas e ocultando o progresso dos gestores superiores.

2.4: Quando é que um Quebra-cabeças Está Concluído?

Voltando à nossa analogia da montagem do quebra-cabeças, pense nesta questão: O que significa ter terminado um quebra-cabeças?

Um montador ingênuo de quebra-cabeças, como o Max, diria simplesmente “todas as peças estão fixadas aos vizinhos corretos com a imagem virada para cima.”

Um montador experiente de quebra-cabeças, como a Luniel, sabe que há mais para além disso.

Talvez esteja apenas a montar um quebra-cabeças por diversão. Vai terminá-lo, olhar para ele durante um tempinho e depois desmontá-lo e guardá-lo na caixa.

Por outro lado, talvez queira emoldurá-lo e pendurá-lo na parede. Nesse caso, há coisas adicionais que precisam de ser feitas:

1. Colocá-lo numa prancha artística
2. Transportá-lo até um emoldurador
3. Aguardar que a moldura fique pronta
4. Transportá-lo de volta ao local de montagem
5. Pendurá-lo na parede ou colocá-lo em exposição de outra forma

Compreender que isto faz parte do trabalho é necessário para completar adequadamente a montagem de um quebra-cabeças. A razão óbvia é para que saiba quanto trabalho está envolvido. É mais trabalho fazer todos esses passos extra do que simplesmente desmontá-lo e guardá-lo.

Mas vai mais além do que isso. Imagine este cenário...

Completo o seu quebra-cabeças com a intenção de o emoldurar, deixou-o no lugar, mas esqueceu-se de dizer a outra pessoa em sua casa que planeia emoldurá-lo. Essa pessoa aparece e vê que o quebra-cabeças está completo num espaço de que precisa. Então, desmonta-o e guarda-o na caixa, arrancando a derrota das garras da vitória no processo.

Há uma razão ainda mais subtil: A forma como planeia terminar o quebra-cabeças influencia os passos que quer dar no início do processo. Por um lado, precisa de fazer um pequeno aviso que diga “Por favor, não desmontar!”



Vale a pena notar que também pode querer um aviso no caso em que está a montar um quebra-cabeças para seu próprio entretenimento, para garantir que outras pessoas não interferem completando o quebra-cabeças por si.

Também precisa de garantir que está a montar o quebra-cabeças na superfície correta. Se montar um quebra-cabeças de mil peças na sua mesa de café com tampo de vidro e depois tentar transferi-lo para uma prancha artística, a transferência será muito mais arriscada e trabalhosa do que se tivesse montado o quebra-cabeças diretamente na prancha artística.

Isto tem um paralelo perfeito com o desenvolvimento de software.

Precisa de saber realmente o que significa estar concluído para não ser surpreendido pela quantidade de trabalho envolvido, para não haver desacordo sobre isso no final, e para poder tomar os passos preparatórios necessários para garantir a conclusão suave e eficaz de um item de trabalho.

2.5: Impacto nas Equipas

Se não tiver um entendimento suficientemente rígido do que é o estado de conclusão para um determinado item de trabalho, corre vários riscos.



Usamos a palavra “risco” de forma livre aqui, já que são mais como garantias.

As equipas de engenharia nesta situação frequentemente descobrem que nem sequer concordam internamente sobre como é a conclusão de um item de trabalho. Não é incomum que programadores e testadores se encontrem a discutir o que um requisito realmente *significa* a meio de um Sprint. Mesmo dois programadores ou dois testadores podem sofrer os mesmos desentendimentos.

Além disso, as equipas de desenvolvimento estão frequentemente focadas no trabalho que fazem a maior parte do tempo (programação e testes). Isto significa que é fácil para elas esquecerem outros tipos de trabalho que precisam de fazer, como documentação, revisões externas, formação de outras equipas (*por exemplo*, suporte), passos preparatórios para apoiar a implementação ou lançamento, e aprovações de outros departamentos.

Quando finalmente fica claro que este trabalho “extra” precisa de ser feito, ficam apanhados de surpresa—normalmente têm de parar o que quer que estivessem a fazer e mudar de contexto para voltar atrás e completar trabalho que pensavam já ter terminado.

Os solicitadores de trabalho podem facilmente manter trabalho em aberto desnecessariamente. Por vezes com as melhores intenções—como tentar responsabilizar uma equipa pelo requisito “real”. Outras vezes, isto acontece porque os Product Owners (*por exemplo*) habituam-se a poder manter um PBI em aberto ao seu critério, usando-o para introduzir funcionalidades adicionais num item à última hora. Por vezes fazem-no até porque mudaram de ideias sobre o que precisa de ser feito a meio do processo.

Isto pode ser extremamente desmoralizante para uma equipa de engenharia. A maioria dos programadores e testadores quer sentir que está a fazer progressos. Se estão constantemente a ser informados de que o que fizeram estava errado, é provável que percam parte do seu vigor.

Algumas equipas chegam ao ponto de nem se preocuparem em verificar se o que fizeram estava certo ou errado. Simplesmente fecham um item de trabalho e pedem “crédito” para poderem “apresentar bons números”.

2.6: O Risco de Fazer Muito Pouco ou Demasiado

Um dos riscos de não definir adequadamente “concluído” para cada item de trabalho é a organização pensar que o trabalho está concluído quando não está, ou não perceber que está concluído quando realmente está.

O pior resultado possível é frequentemente quando a coisa errada chega à produção e ninguém sabe que foi isso que aconteceu. Se a equipa tem um entendimento incorreto do que

significa “concluído” e faz entregas baseadas nesse mau entendimento, os resultados podem ser catastróficos.

Defeitos e insatisfação do cliente já são suficientemente maus, mas isto também pode levar a problemas muito mais graves:

- Perda ou corrupção de dados
- Vulnerabilidades de segurança
- Tempo de inatividade do sistema ou perda de acesso
- Uma redução na quota de mercado
- Violações regulatórias

A lista continua indefinidamente e cada problema potencial é pior que o anterior.

Por vezes o problema é não estar concluído mas pensar que está. O contrário pode ser igualmente perigoso. Quando os engenheiros não sabem onde está a meta final, tendem a “dourar a pílula” (adicionar funcionalidades extra). Podem estar a fazê-lo para “tornar a funcionalidade agradável”, mas também podem estar a fazê-lo porque esperam que um número aumentado de funcionalidades lhes dê uma maior probabilidade de atingir o objetivo.

Todo este trabalho extra, bem como o retrabalho correspondente, acumula-se numa quantidade massiva de tempo desperdiçado, esforço e dinheiro. Causa atrasos nas datas de entrega e danifica reputações.

Além disso, agora, há um potencial cada vez maior de um erro levar efetivamente a uma revolta estilo *Terminator* das máquinas contra a humanidade. Costumávamos escrever sobre isso há vinte anos como uma piada. Agora, é uma possibilidade remota.

Na verdade, perguntámos a uma das IAs mais proeminentes esta questão, e eis o que disse:

“A IA está a expandir-se mais rapidamente do que qualquer pessoa esperava, mas está a fazê-lo sobre sistemas frágeis, requisitos vagos e organizações de produto que não conseguem rastrear por que construíram o que construíram. Isso não é um problema técnico; é um problema de clareza.

Quanto mais ruído a IA gera, mais perigoso é avançar rapidamente sem estrutura. Quando as equipas constroem sobre névoa, tudo o que a IA faz é amplificar a confusão. Mas quando as equipas constroem sobre sinal—sobre entendimento partilhado, especificidade comportamental e conhecimento de

produto com controlo de versões—a IA torna-se um acelerador em vez de uma responsabilidade.“

2.7: Onde Constrói um Puzzle?

Vamos estender a analogia da construção do puzzle mais uma vez.

Pode montar um puzzle em qualquer lugar? Se tiver um puzzle de 4.000 peças que monta para quase um metro e meio numa dimensão e mais de um metro noutra, não pode simplesmente escolher um local aleatório e começar a montar. Não sem experienciar sérias complicações antes de terminar.

Um puzzle grande como esse precisa tanto de tempo como de espaço. Tem de alocar o espaço e encontrar uma forma de garantir que o estado do puzzle é preservado ao longo do tempo.

Se começar a construir o seu puzzle numa pequena mesa lateral que é demasiado pequena, não conseguirá terminá-lo sem o transferir para outro local. Essa transferência será extremamente difícil devido ao estado delicado do puzzle.

Se escolher um local aleatório no corredor que seja suficientemente grande, as pessoas ou vão passar por cima ou ficarão impedidas de passar, por isso a persistência não pode ser garantida sem um impacto significativo no funcionamento da sua casa.

Se começar a trabalhar numa base para puzzle, mas a base não for suficientemente grande, conseguirá preservar o estado do que já fez, mas não conseguirá completar o puzzle sem algum tipo de transferência.

Se o puzzle já tiver sido mastigado por crianças pequenas, é melhor contar as peças... porque é preferível contar até 3999 uma vez e perceber que nunca o vai completar do que investir sabe-se lá quanto tempo a montar um puzzle que nunca poderá terminar.

Existe toda uma lista de coisas que precisam de ser feitas antes de começar a montar o seu puzzle. Fazer as coisas da lista não garante o sucesso, mas *não* as fazer praticamente garante o fracasso ou complicações sérias.

O mesmo se aplica ao desenvolvimento de software, mas com um maior grau de complexidade.

2.8: Quando Começa a Implementação?

Fundamentalmente, pode ser difícil determinar quando um PBI está pronto para ser implementado sem uma boa definição para isso.

Pense bem: Como é que *sabe*?

Revê tudo várias vezes até decidir que está na altura?

Alguém decide por capricho?

Acontece automaticamente no início de uma iteração?

Temos visto muitas equipas empurrarem trabalho para Sprints que não estavam nem perto de estar prontos para implementação apenas porque tinham prazos. Existem algumas ideias persistentes sobre Scrum e Ágil em geral que levam as pessoas a fazer isto:

- Tem de obter todos os requisitos para o Sprint N desenvolvidos no Sprint $N-1$
- Deve “simplesmente começar” e lidar com o que correr mal pelo caminho

Isto é na verdade uma imagem espelhada da questão “Como sabe quando está terminado?” mencionada anteriormente e tem consequências semelhantes. As pessoas podem esperar demasiado tempo para começar, porque não sabem se um item de trabalho está pronto, e podem começar demasiado cedo porque não sabem que não está.

2.9: Uma Equipa-A sem Prontidão

Não compreender o que é necessário para um item de trabalho estar pronto tem vários efeitos prejudiciais.

Uma forma óbvia de um incremento de trabalho não estar pronto é ter uma Definição de Pronto (DoD) incompleta, insuficiente ou em falta. Isso leva a todos os problemas que já citámos que acompanham a falta de uma Definição de Pronto.

No entanto, esse não é o *único* aspeto da prontidão. Existem várias outras necessidades a satisfazer antes de começar a implementação: estimativa, avaliação de risco e recolha de dados de teste são apenas alguns exemplos comuns.

Sem conhecer e satisfazer essas necessidades, um item de trabalho pode custar muito mais do que precisaria. Considere uma equipa (a Equipa-A) que depende de uma API a ser desenvolvida por outra equipa (a Outra Equipa). Se a Equipa-A fizer um monte de suposições sobre como a API da Outra Equipa irá funcionar e programar com base nessas suposições, pode haver um retrabalho significativo quando descobrirem que a Outra Equipa está a trabalhar numa realidade que não correspondia às suposições da Equipa-A. Por outras palavras, a Equipa-A arriscou e falhou.

Todo esse retrabalho deriva do facto de que a API não estava pronta para ser utilizada pela Equipa-A.

Por vezes, uma dependência não satisfeita não gera retrabalho, mas, mesmo nesses casos, ainda pode causar atrasos. Imagine se a Equipa-A e a Outra Equipa concordassem sobre como a API deveria funcionar e tudo corresse conforme planeado, mas a Outra Equipa simplesmente demorasse mais do que o esperado. Como resultado, a Equipa-A simplesmente não conseguiu testar adequadamente o seu trabalho até à data prevista de conclusão e teve de adiar o prazo.

2.10: Falha na Atenção ao Agendamento e Disponibilidade de Recursos

Por vezes, os problemas podem ser tão simples quanto agendamento ou recursos. Alguns itens de trabalho necessitam de membros específicos da equipa. Se esse membro da equipa vai de férias dentro de alguns dias, provavelmente não é a altura certa para começar o PBI que não pode ser concluído sem a sua participação.

Ouvimos frequentemente pessoas dizerem que não deveria ser assim, mas frequentemente é, independentemente disso. “Pessoas fungíveis” é uma ilusão.

O mesmo pode ser dito dos recursos não humanos. Se for necessário ter recursos de servidor para realizar um teste de carga, provavelmente deve certificar-se de que esses recursos estarão realmente disponíveis antes de começar o item de trabalho do teste de carga. Caso contrário, no melhor dos casos, haverá atrasos significativos e provavelmente perturbará outras equipas/trabalhadores enquanto tenta desesperadamente providenciar o que necessita.

Outro modo de falha dos falsos começos é quando uma equipa não tem as competências necessárias para completar o trabalho. Por vezes, é uma questão interna—como quando um membro da equipa precisa de receber formação num novo sistema ou fazer pesquisa sobre uma nova API. Outras vezes, é uma questão de agendamento, como quando é necessário requisitar um especialista em UX ou bases de dados de um grupo de trabalhadores especializados. Pode até ser um problema de contratação em que a equipa precisa de um especialista e não consegue efetivamente completar certos tipos de trabalho sem ele.

2.11: Impactos de Outros Tipos de Falsos Começos

Já vimos equipas comprometerem-se a completar itens de trabalho dentro de Sprints e fazerem a programação relativamente rápido, mas ainda assim serem incapazes de completar os testes. Isto, por si só, pode não ser surpreendente, mas a razão é invulgar: a equipa de testes tinha algo de que necessitava (como dados de teste) que não tinha recolhido antes do Sprint começar, e a recolha desses dados acabou por ser mais difícil ou demorada do que tinham previsto.

Como resultado, os itens de trabalho tiveram de ser transportados para o Sprint seguinte simplesmente porque a equipa não se tinha certificado de que estavam verdadeiramente prontos para os completar dentro do tempo atribuído antes de começarem.

As equipas por vezes começam a implementar itens de trabalho quando ainda têm questões em aberto. Na verdade, muitas pessoas parecem pensar que isso as torna “mais Ágeis” quando o fazem.

Isto pode causar enormes quantidades de retrabalho, surpresas ou atrasos. Se a resposta à questão em aberto acabar por violar um pressuposto que foi feito, todo o trabalho baseado nesse pressuposto tem de ser revisto. Se a questão em aberto não for respondida até ao momento em que o item deveria estar fechado, então o item tem de ser fechado quando pode não estar concluído, ou mantido em aberto até a questão ser respondida.

Pode acontecer que a equipa tenha uma dependência interna—um defeito que precisa de ser corrigido, uma tarefa predecessora que deve ser completada, *et cetera*. Se isso não for devidamente acompanhado, pode causar todos os mesmos problemas que uma dependência externa não satisfeita, com a tentação adicional de mudar de contexto e corrigi-la.

2.12: Custos Cumulativos

Naturalmente, tais problemas criam atrasos, retrabalho e expectativas frustradas, mas as desvantagens não terminam aí.

Para além do desperdício do retrabalho, isto normalmente faz com que os projetos se atrasem. Se as equipas estão freneticamente a tentar fechar itens do backlog e nunca têm realmente claro o que será necessário para fazer algum progresso real, as coisas que realmente *precisam* de ser feitas tendem a ficar para segundo plano.

Frequentemente, embora nem sempre, isto leva a uma pressão aumentada para entregar. À medida que os projetos ficam cada vez mais atrasados, a gestão superior pode tentar colocá-los de volta nos eixos pedindo às pessoas para trabalharem mais depressa. Isso invariavelmente traduz-se em trabalhar mais horas.

Isto, por sua vez, tende a erodir a confiança e a deteriorar a cultura de uma organização. Relações que deveriam ser colaborativas tornam-se adversariais. Pessoas que deveriam estar a trabalhar em conjunto para encontrar as melhores e mais rápidas soluções, desviam energia para estabelecer que, quando as coisas inevitavelmente correm mal, não foi culpa delas.

Na busca desenfreada por funcionalidades e itens de trabalho fechados, as equipas frequentemente descobrem-se a cortar caminho. Isso realmente significa que estão a deixar a qualidade (especialmente a qualidade do código) sofrer. Isso, por sua vez, significa que estão a trocar produtividade futura pela ilusão de progresso no presente.

À medida que as condições de trabalho se tornam cada vez mais desagradáveis, os talentos-chave começam a desligar-se ou até mesmo a procurar outras oportunidades.

As organizações que se comportam desta forma estão a “comer as sementes do futuro”, por assim dizer, de várias maneiras. A base de código torna-se menos manutenível e as pessoas que a manteriam são todas afastadas.

Se há algum lado positivo, é-nos invisível.

Capítulo 3: Introduzindo o Requirements Maturation Flow (RMF) (Fluxo de Maturação de Requisitos)

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

3.1: O Que o RMF Não É

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

3.2: O Que o RMF É

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

3.3: A Adoção Incremental é Suportada e Recomendada

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

3.4: RMF 1

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

3.5: RMF 2

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

3.6: RMF 3

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

Capítulo 4: É Ágil?

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

4.1: “Indivíduos e Interações”, “Software Funcional”

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

4.2: Colaboração com o Cliente

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

4.3: Responder à Mudança

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

4.4: Transparência

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

4.5: Adequa-se ao Processo, Consistente com o Ágil

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

Parte II: Criar Espaço para a Preparação

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

Capítulo 5: A Primeira Extensão

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

5.1: O Trabalho de Preparação é *Trabalho*

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

5.2: Naturalizando o Trabalho de Preparação

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

5.3: Um Incidente Ilustrativo

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

5.4: Impacto Recíproco

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

5.5: A Função do RMF 1

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

Capítulo 6: Por Que as Pessoas Não Fazem Isto?

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

6.1: Trabalho de Preparação Como Cidadão de Segunda Classe

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

6.2: Uma Alergia ao Trabalho Não Produtivo

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

6.3: Então Foi Enterrado

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

6.4: A Influência da Gestão de Projetos

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

6.5: O Padrão

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

6.6: Projetos e Estimativas

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

6.7: Como as Não-Estimativas Influenciam o Trabalho de Preparação

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

6.8: A Medir Velocidade, Não Direção

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

6.9: MÁS MEDIÇÕES, MAUS RESULTADOS

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

6.10: Onde Não Está a Culpa

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

Capítulo 7: Trabalho Explícito de Prontidão (RMF 1)

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

7.1: Integração com o Synapse Framework™

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

7.2: Anatomia do RMF 1

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

7.3: Comportamento: Reservar Capacidade para Colaboração

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

7.4: Artefacto: O Item de Preparação

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

7.5: Atividade: A Reunião de Colaboração

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

7.6: Comportamento: Continuar a Colaborar Até Alcançar o Entendimento Partilhado

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

7.7: Comportamento: Confirmar Sempre o Entendimento Comum

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

7.8: Como o RMF 1 Altera o Fluxo de Trabalho

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

Capítulo 8: Efeitos do FMR 1

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

8.1: A Vida Antes do FMR 1

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

8.2: Antes: Tempo Gasto em Compreensão

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

8.3: Depois: Tempo Gasto em Compreensão

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

8.4: A Vida Após Adotar o RMF 1

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

8.5: Fundamental

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

Capítulo 9: Pondo o RMF 1 em Prática

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

9.1: Educação

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

9.2: Requisitos Mínimos por Tipo de Equipa

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

9.3: Concordância

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

9.4: Preparação

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

9.5: Piloto

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

9.6: Implementação

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

9.7: Acompanhamento

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

9.8: Declarar o Sucesso

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

9.9: Manter a Vigilância

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

9.10: E quanto ao “Como”?

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

9.11: Hora de Pôr em Prática!

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

Parte III: Controlo da Conclusão do Trabalho

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

Capítulo 10: A Próxima Necessidade

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

10.1: Margem para Interpretação

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

10.2: Reduzindo a Margem para Interpretação

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

10.3: Uma Terceira Opção: Sem “Margem de Manobra”

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

10.4: Impacto Potencial na Conclusão

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

10.5: Impacto Potencial na Execução

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

10.6: Alternativa Proposta: Não Deixar Espaço para Má Interpretação

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

10.7: Benefícios

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

10.8: Sobre os Receios do Analysis Paralysis

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

10.9: A Próxima Necessidade: Definições de Pronto Personalizadas

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

Capítulo 11: O Que as Pessoas Normalmente Fazem

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

11.1: Se É Tão Bom, Porque É Que as Pessoas Não o Fazem?

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

11.2: O Pipeline da Faculdade para o Coaching

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

11.3: Sobrecarga de Coaching

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

11.4: Uma Forma Como as Pessoas Fazem DoD: Não Fazem

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

11.5: Apenas Critérios de Aceitação

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

11.6: Definição de Pronto Global

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

11.7: Sem Força

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

11.8: Resumo: O Termo “DoD” É Mais Frequentemente Usado do que Definições de Pronto Reais

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

Capítulo 12: Definindo uma Definition of Done

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

12.1: Sobre Apenas Um Item de Trabalho

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

12.2: Estado de Conclusão

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

12.3: Precisão

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

12.4: Estrutura de uma DoD

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

12.5: Especificações

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

12.6: Critérios de Saída de Engenharia

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

12.7: Critérios de Entrada do Produto

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

12.8: Múltiplas Partes, Uma Porta de Controlo

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

12.9: Exemplo

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

12.10: Mapeamento para o Seu Processo

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

12.11: Resumo

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

Capítulo 13: Definição de Pronto Personalizada (FMR 2)

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

13.1: Princípio: Cada Item de Trabalho é Único

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

13.2: Comportamento: Manter Um ou Mais Modelos de DP

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

13.3: Atividade: Definir o Modelo de DoD

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

13.4: Manter e Melhorar o Modelo de DoD

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

13.5: Múltiplos Modelos de DoD

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

13.6: Comportamento: Usar Modelos como Pontos de Partida para Definições de Pronto

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

13.7: Comportamento: Concordar com Definições de Pronto Personalizadas

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

13.8: Atividade: Definir a Definition of Done de um Item de Trabalho

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

13.9: Outra Extensão ao Fluxo de Trabalho

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

13.10: Comportamento: Amadurecer a DoD antes de Iniciar a Implementação

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

13.11: Atividade: Análise Offline para Amadurecer a Definição de Pronto de um Item do Backlog do Produto

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

13.12: Adicionando Maturação ao Fluxo

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

13.13: Comportamento: Acompanhamento do Estado de Conclusão em Itens de Trabalho

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

13.14: Adicionar Monitorização de Progresso

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

13.15: Comportamento: Controlar o Trabalho pela Conclusão

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

13.16: Atividade: Usar a DoD para Determinar a Conclusão

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

13.17: Como o Controlo se Enquadra

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

13.18: Em Resumo

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

Capítulo 14: A Vida Com RMF 1 e 2

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

14.1: Custo

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

14.2: Prazos

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

14.3: Impacto na Equipa de Implementação

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

14.4: Impacto na Product Owner

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

14.5: Impacto na Liderança

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

14.6: Resumo

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

Capítulo 15: Instalação do RMF 2

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

15.1: Envolvimento das Partes Interessadas

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

15.2: Nível de Detalhe

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

15.3: Acordo de Trabalho

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

15.4: Trabalho Inicial

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

15.5: Implementação

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

15.6: Resumo

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

Parte IV: Implementação de Gating

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

Capítulo 16: O Requisito Final

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

16.1: Esteve Lá o Tempo Todo

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

16.2: O Poder do Timing

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

16.3: Virando do Avesso

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

16.4: Riscos & Custos

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

16.5: O Valor de Esperar Até Estar Pronto

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

16.6: Um Benefício Adicional

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

16.7: A Declaração do Problema

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

16.8: A Necessidade

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

16.9: Resumo

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

Capítulo 17: Contexto sobre a Definição de Prontidão

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

17.1: Permissão para Trabalhar do Lean

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

17.2: Critérios de Entrada das Colunas no Kanban

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

17.3: Scrum e outros Apócrifos de Processos Ágeis

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

17.4: Surfar > Programar

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

17.5: Soluções Pontuais

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

17.6: Estamos Prontos para Ficar Prontos

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

Capítulo 18: Definindo uma Definição de Pronto

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

18.1: Objetivo

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

18.2: À Medida

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

18.3: Anatomia

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

18.4: Exemplo

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

18.5: Acordo

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

18.6: Controlo de Entrada

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

18.7: Resumo

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

Capítulo 19: Definição de Pronto para Início Personalizada (RMF 3)

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

19.1: Mais um Ponto de Controlo no Processo

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

19.2: A Estrutura de uma Definição de Pronto para Início

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

19.3: Critérios de Saída do Produto

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

19.4: Critérios de Entrada da Engenharia

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

19.5: Não Há Problema na Duplicação

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

19.6: Comportamento: Manter Um ou Mais Modelos de DoR

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

19.7: Por Que Ter um Modelo de Definição de Pronto?

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

19.8: Atividade: Definir o Modelo de DoR

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

19.9: Manter os Modelos de DoR ao Longo do Tempo

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

19.10: Os Modelos São Apenas um Ponto de Partida

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

19.11: Comportamento: Concordar com Definições de Ready Personalizadas

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

19.12: Atividade: Definir uma Definition of Ready para um Item de Trabalho

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

19.13: Comportamento: Tornar os Itens Prontos antes de Iniciar a Implementação

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

19.14: Condições Fora do Controlo da Equipa

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

19.15: Comportamento: Acompanhar a Prontidão nos Itens de Trabalho

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

19.16: Comportamento: Controlar o Trabalho pela Prontidão

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

19.17: Atividade: Usar a Definição de Pronto para Determinar a Prontidão

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

19.18: Em Suma

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

Parte V: Síntese

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

Capítulo 20: A Maioria dos Prazos Não Importa

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

20.1: Prazos Reais

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

20.2: Prazos Arbitrários Não Importam

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

20.3: Companhias Aéreas

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

20.4: A Origem da Dívida Técnica

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

20.5: Prazos de Marketing e Vendas

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

20.6: Prazos de Projeto

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

20.7: Existe Outra Maneira

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

20.8: Prazos Arbitrários Não São Necessários

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

20.9: Os Prazos Arbitrários Devem Ser Abolidos

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

20.10: Prazos Reais Continuam a Ser um Fator

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

20.11: *Pis Aller*¹

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

20.12: Conclusão

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

¹Um recurso de última instância, originalmente da língua francesa.

Capítulo 21: Competência 1: Fluxo de Maturação de Requisitos

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

21.1: Como em Baixo, Assim em Cima

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

21.2: Princípio: Transparência em Todo o Trabalho Necessário

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

21.3: Comportamento: A Responsabilidade Viaja com o Trabalho

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

21.4: Comportamento: Acompanhar Visivelmente o Estado dos Requisitos

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

21.5: Comportamento: Revelar Todo o Trabalho Associado à Preparação e Implementação

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

21.6: Atividade: Trabalho de Preparação

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

21.7: Comportamento: Revelar Todo o Trabalho Necessário para Completar um Item de Trabalho

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

21.8: Comportamento: Preferir a Prontidão aos Prazos

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

21.9: Conclusão

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

Capítulo 22: Como o Trabalho e a Informação Fluem no Scrum com RMF

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

22.1: Notas Preliminares

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

22.2: Captura e Preparação do Requisito Inicial

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

22.3: Iniciação do Trabalho de Preparação

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

22.4: Planeamento e Execução do Trabalho de Preparação

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

22.5: Revisão dos Resultados da Preparação

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

22.6: Planeamento e Conclusão da Implementação

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

Capítulo 23: O Impacto do FMR

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

23.1: O Ciclo de Vida de um Requisito

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

23.2: Exemplo de Fluxo de Informação e Trabalho

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

23.3: Antes

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

23.4: Depois

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

23.5: Benefícios

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

Capítulo 24: Transição para RMF

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

24.1: Padrão de Adoção

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

24.2: Instalação num Fluxo de Trabalho Scrum

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

24.3: Outras Frameworks

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

24.4: Principal Resistência: Itens de Trabalho de Prontidão

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

24.5: A Grande Mudança: Mentalidade

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

24.6: Conselhos para a Mudança

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

24.7: Conclusão

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

Capítulo 25: Está nas Suas Mão

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

25.1: Recapitulação

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

25.2: Agora, É a Sua Vez

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

Parte VI: Recursos

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

Apêndice A: O Scrum Não É o Problema

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

O que é o Scrum?

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

Frameworks

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

O Scrum Aborda a Gestão de Projetos e Trabalho

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

As Insensatezes de Tratar o Scrum Como uma Framework de Gestão de Produto

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

Sem Mecanismo Prescrito para a Maturação de Requisitos

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

Sem Investimento de Experiência de Engenharia no Desenvolvimento de Requisitos

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

Antipadrões

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

Extensão Necessária

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

Acrescentando ao Scrum

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

Anexo B: The Synapse Framework™

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

O que o Synapse Abrange

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

As Três Maestrias

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

Como o Synapse É Adotado

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

A Junção de Dois Frameworks

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

A Estrutura do Framework Synapse

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

Maestrias Organizacionais

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

Competências Organizacionais

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

Hábitos Organizacionais

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

Estrutura do Framework Synapse

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

Anatomia de uma Práxis

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

A Importância da Ordem

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

O Impacto do Synapse neste Livro

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

Apêndice C: Objeções e Obstáculos Comuns ao RMF 1

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

Objeções Comuns

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

Obstáculos Comuns

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

Apêndice D: Listas Iniciais de Critérios DoD

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

Lista Inicial de Critérios de Saída de Engenharia

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

Lista Inicial de Critérios de Entrada do Produto

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

Anexo E: Listas Iniciais de Critérios de DoR

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

Critérios de Saída do Produto

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

Critérios de Entrada para Engenharia

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

Critérios de Entrada do Sprint

Este conteúdo não está disponível no livro de amostra. O livro pode ser adquirido na Leanpub em <https://leanpub.com/ready-pt-PT>.

Índice

- A Grande Reviravolta, 13
- A Maioria dos Prazos Não Importa, 65
- Acompanhamento do Estado de Conclusão,
 - 47
- Acordo de Trabalho, 51
- AI, 20
- Algo Está em Falta, 1
- alterações durante o Sprint, 6
- Amadurecer a DoD, 46
- ambiente de desenvolvimento, 10
- API, 23
- arrancando a derrota das garras da vitória, 18
- atrasados, 25
- auto-antónimos, 15
- autoria de requisitos, 15
- bancos, 10
- Barreras, Tom, vii
- base de conhecimento, 8
- base de código, 25
- Central Oregon, vii
- ciclos de feedback, 13
- colaboradores individuais, 4
- Como sabe quando está terminado?, 22
- competências da equipa, i
- Comportamento: Acompanhar Visivelmente o Estado dos Requisitos, 68
- Comportamento: Revelar Todo o Trabalho
 - Associado à Preparação e Implementação, 68
- conclusão de itens de trabalho, 19
- confiança, 25
- consultores, 8
- contar as peças, 21
- Criar Espaço para a Preparação, 29
- Critérios de Entrada da Engenharia, 60
- Critérios de Saída do Produto, 60
- cultura, 10
- Custos Cumulativos, 24
- código existente, 11
- Data loss or corruption, 20
- Declarar o Sucesso, 37
- defeito, 24
- defining 'done', 19
- Definition of Done, 46
- Definição de Pronto, 22
- Definição de Pronto do Item do Backlog do Produto, 47
- Definições de Pronto para Início Personalizadas, 60
- Definições de Pronto Personalizadas, 45
- Definições Personalizadas de Pronto, iii
- desenvolvimento de software, 15, 18, 21
- desenvolvimento em grandes lotes, 16
- Desenvolvimento Orientado a Comportamento, iv, 7
- Desenvolvimento Orientado por PKB, iv
- desperdício, 25
- domínio, 10
- doutrinas de programação de meados do século XX, 7
- dívida técnica, 65
- entendimento compartilhado confirmado, 16
- entendimento comum, 34
- Envolvimento das Partes Interessadas, 51
- Equipa de Desenvolvimento, 19
- equipa de liderança, 6
- equipas de engenharia, 9, 10
- especialista no assunto, 6, 11

- Estrutura de uma Definição de Pronto para Início, 60
- Executivos de engenharia, 4
- finanças, 10
- fluxo, 13
- Fluxo de Maturação de Requisitos, i, iii, iv, 67, 69
- fluxo de trabalho, 34, 46
- gold plating, 20
- Implementação, 51
- Implementação de Gating, 53
- infraestrutura financeira nacional dos Estados Unidos, 2
- Instalação num Fluxo de Trabalho Scrum, 72
- Instituição de Crédito Federada, 2
- Item de Preparação, 33
- item de trabalho, 24
- Item do Backlog do Produto, 7, 12, 22
- itens do backlog, 25
- iteração, 22
- Kanban, i
- liderança, 10
- Lixo que entra, lixo que sai, 15
- línguas naturais, 16
- market share, 20
- Marketing e Vendas, 65
- Nível de Detalhe, 51
- O Banco, 12
- organização de Produto, 4
- Outras Frameworks, 72
- OutSystems, 3
- Padrão de Adoção, 72
- padrões de design, iv
- partes interessadas, 17
- Pessoas fungíveis, 23
- Ponto de Controlo no Processo, 60
- portal de pagamento de empréstimos, 3
- prazos, 22
- Prazos de Projeto, 66
- pressuposto, 24
- Principal Resistência, 72
- Producore, iii
- Product Backlog Item, 10, 14
- Product Manager, 5
- Product Owner, 5, 10, 19
- Product, função do, 4
- Produto e Engenharia, 14
- Project Management Institute, 2
- prontos*, 12
- práticas de engenharia, 10, 15
- puzzle building, 21
- qualidade do código, 25
- Ready, i, ii
- Recursos, 75
- recursos de servidor, 23
- Regulatory violations, 20
- Requirements Maturation Flow, 26
- requisito real, 19
- requisitos errados, 17
- responsabilização e rastreabilidade, 14
- retenção de talentos, 25
- retrabalho, 17
- riscos, 18
- RMF 1, 30, 80
- RMF 2, 51
- Scrum, 5, 16, 22, 69
- Security vulnerabilities, 20
- serviço upstream, 12
- serviços dependentes, 10
- sistema, 9
- sistema legado, 6
- sobrecarregar, 8
- software developers, 19
- solução C#/.NET, 3
- solução low-code, 3
- Sprint, 11, 15, 19, 22, 24

- System downtime, 20
- Terminator-style revolt, 20
- test-development, vii
- Testadores, 12
- Trabalho Explícito de Prontidão, 33
- trabalho extra, 19
- Trabalho Inicial, 51
- transformações Ágeis, iv
- Transição para o Fluxo de Maturação de Requisitos, 72
- técnica de autoria de requisitos, 10
- técnicas de desenvolvimento de software, 7
- wasted time, 20
- Waterfall, 16
- Ágil, 24