

# AGILITY

Pavimentando o caminho DevOps com Agilidade, Qualidade e Testes, práticas DevOps e AWS

## Sumário

O que é Agilidade? .....	4
Waterfall vs Agility.....	4
O que é valor hoje? .....	5
Ciclo de Feedback .....	5
Ice-breaker batalha naval .....	5
Kaizen .....	5
Todos falam a mesma língua.....	5
Criação de um Growth Mindset.....	5
Aumento da Motivação .....	5
Melhor aceitação de novas ideias .....	5
Como aplicar Kaizen no contexto de Feedbacks de Usuário .....	6
Ciclo de Feedback.....	6
PDCA .....	6
NPS® – Net Promoter Score .....	6
Origem.....	6
Como utilizar.....	6
Análise.....	7
Faixa de NPS .....	7
Na Prática.....	8
Outcomes de se utilizar Feedback de Usuários.....	8
Desenvolvimento Individual .....	8
Pilares de uma Equipe de Alta Performance .....	8
Pilar 1: Comunicação do Time .....	8
Pilar 2: Competência individual.....	8
Pilar 3: Comprometimento Geral.....	8
Pilar 4: Compartilhamento de Conhecimento .....	9
Habilidades .....	9
Sugestões de Transmissão de Conhecimento .....	9
Profundidade de Conhecimento .....	9
I Shaped Skill.....	9
T Shaped Skill.....	9
Pi Shaped Skill.....	10
Comb Shaped Skill .....	10
Matriz de Competências.....	10
Quais os benefícios de uma matriz de competência? .....	10
Small Batches.....	10

O que faz parte do lote de trabalho? .....	10
Diferenças entre Big e Small Batches .....	10
Processo com lote grande .....	10
Processo com Lote Pequeno .....	11
Ganhos na otimização de lotes .....	11
Gargalos .....	11
CFD – Cumulative Flow Diagram .....	11
Fluxo e Limite de WIP .....	12
UpStream e DownStream na Agilidade .....	12
O que é WIP? .....	12
Objetivos de Limitar o WIP .....	12
Efeito Zeigarnik .....	12
Estimativa.....	12
Story Points .....	12
Planning Poker .....	13
Indicadores e Métricas.....	13
Velocity .....	13
Capacity .....	13
Throughput.....	13
Lead Time .....	13
Cycle Time .....	14
ACCELERATE: Four Key Metrics.....	14
MTTR .....	14
Deploy Frequency.....	14
Change Fail Rate.....	14
Lead time for Changes.....	14

## O que é Agilidade?

Em resumo, podemos dizer que Agilidade é sinônimo de ser flexível.

Definições modernas de desenvolvimento de software ágil evoluíram a partir dos anos 1990 como parte de uma reação contra métodos “pesados”, usado o modelo em cascata para desenvolvimento.

Em 2001, membros da comunidade se reuniram em Snowbird e adotaram o nome métodos ágeis, tendo publicado o Manifesto ágil, a partir disso o que eram chamados de métodos leves, ganharam o nome de Métodos Ágeis.

No manifesto, foram definidos quatro valores:

INDIVÍDUOS E INTERAÇÕES MAIS QUE PROCESSOS E FERRAMENTAS

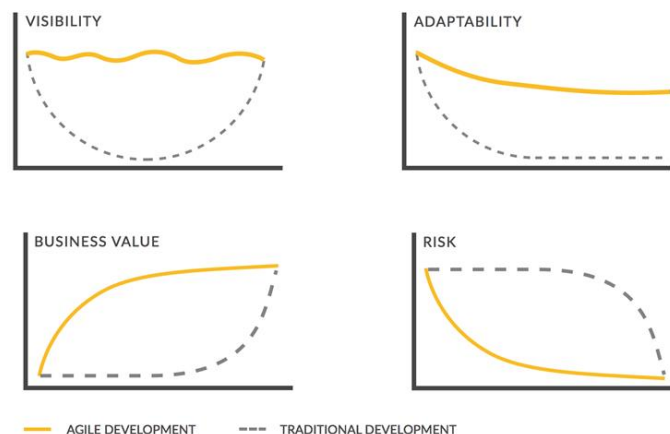
SOFTWARE EM FUNCIONAMENTO MAIS QUE DOCUMENTAÇÃO ABRANGENTE

COLABORAÇÃO COM O CLIENTE MAIS QUE NEGOCIAÇÃO DE CONTRATOS

RESPONDER A MUDANÇAS MAIS QUE SEGUIR UM PLANO

## Waterfall vs Agility

Waterfall é o método mais tradicional de gerenciamento de projeto usado há mais de 50 anos, trata-se de um processo sequencial. Foi criado pelo Dr. Winston W. Royce, em 1970 - O nome apareceu no estudo “Gerenciando o desenvolvimento de grandes sistemas de software”



Visibilidade e transparência em todo o ciclo.

**Adaptabilidade:** em métodos ágeis, o planejamento iterativo facilita a adaptação em caso de mudança de requisitos;

**Business Value:** o planejamento e feedback contínuo trazem entrega de valor desde o início do projeto em gestão ágil. No modelo tradicional, o compromisso maior é com o cumprimento do plano.

**Risco:** os built-in controls, utilizados em gestão ágil, minimizam o risco de produzir desperdício.

### O que é valor hoje?

Deixe o cliente definir o que é valor em seu produto, aquilo que ele não está disposto a pagar é desperdício e deve ser eliminado.

**PRODUTOS** precisam estar em constante evolução, por isso, estudos frequentes são válidos a fim de transformar o produto ao longo do tempo.

## Ciclo de Feedback

### Ice-breaker batalha naval

O jogo AGILE BATTLESHIPS da Boxuk (originalmente criado por James Dcrimshire da SDAPTAGILITY LDT) ajuda a ensinar as equipes a importância de ciclos de feedback rápidos, oferecendo duas rodadas:

- As cegas
- Com feedback instantâneo

Acesse em: <https://boxuk.com/battleships>

### Kaizen

Kaizen, na tradução literal significa “**O ato de melhorar pontos ruins**”, nos dicionários modernos seu significado foi adaptado para “**Uma filosofia de negócios japonesa de melhoria contínua das práticas de trabalho, eficiência pessoal, etc.**”

Ao implementarmos a filosofia de Kaizen, temos quatro principais resultados:

#### Todos falam a mesma língua

Pequenas mudanças contínuas e padronização quase automaticamente colocará todos na mesma página. Os funcionários são parte do processo e de sua criação, melhorando a si mesmo junto com ele.

#### Criação de um Growth Mindset

Os valores da empresa estão entre os componentes mais importantes de um negócio de sucesso. Kaizen é uma maneira de unir todos que compartilham a mesma mentalidade e abordagem para o trabalho e desenvolvimento.

#### Aumento da Motivação

Os membros do time são motivados a se engajar e melhorar quando veem que fazem parte da mudança. Quando eles veem suas pequenas ideias incorporadas em um processo de melhoria contínua, eles estarão mais dispostos a acompanhar e contribuir.

#### Melhor aceitação de novas ideias

Quando a sua organização está acostumada a aceitar o status quo, novas ideias e oportunidades podem, às vezes, serem vistas como algo negativo. Com uma estratégia de melhoria contínua implementada, seus times irão se acostumar e apreciar o trabalho com a noção de que mudança é algo positivo.

## Como aplicar Kaizen no contexto de Feedbacks de Usuário



Podemos utilizar o Ciclo de Feedback associado ao Ciclo PDCA.

### Ciclo de Feedback

1. Resposta
2. Avaliação
3. Mensuração
4. Análise
5. Resolução de Problema

### PDCA

1. Plan: Planeje suas melhorias, incluindo a definição de objetivos
2. Do: Execute as ações exigidas para a melhoria
3. Check: Meça seu sucesso relativo à sua linha base
4. Act: Ajuste suas mudanças

## NPS® – Net Promoter Score

### Origem

Desenvolvido por Fred Reichheld, da Brain&Company, uma grande empresa de consultoria e o indicador ganhou notoriedade quando foi publicado em uma edição da revista de negócios famosa de Harvard, dos Estados Unidos. Por esta publicação, muitas pessoas acreditam que o NPS tenha sido criado pela universidade, mas, seu vínculo é apenas devido a publicação na revista Harvard Business Review.

Toda explicação de como surgiu o indicador pode ser encontrada no livro “A pergunta definitiva” ou “The Ultimate Question”. Esta obra, de fácil e rápida leitura, possui todo racional, origem e estudos que chegaram à pergunta atual. Caso você queira entender profundamente o indicador, é essencial esta leitura.

### Como utilizar

Podemos utilizar em uma pesquisa e ela pode coletar tanto feedbacks quantitativos quanto qualitativos. O resultado auxilia a identificar e priorizar possíveis melhorias.

O livro traz a pergunta definitiva sendo “Você nos recomendaria a um amigo?”, para o nosso contexto, podemos adaptá-la para “Em uma escala de 0 a 10, você indicaria essa empresa (produto) para um familiar ou um amigo (squad)?”.

### Análise

Em uma escala de 0 a 10, classificamos os respondentes em 3 categorias: Os Detratores, os Neutros e os Promotores.

São classificados como Detratores aqueles que responderam entre 0 e 6, podemos considerar que estes respondentes tiveram algum tipo de má experiência ou problema com nosso produto/serviço/empresa, onde possivelmente farão uma propaganda negativa sobre.

São classificados como Neutros Passivos aqueles que responderam entre 7 e 8, podemos considerar que estes respondentes tiveram uma experiência de fato neutra, possivelmente não farão nenhum tipo de propaganda.

São classificados como Promotores aqueles que responderam entre 9 e 10, podemos considerar que estes respondentes tiveram uma ótima experiência com nosso produto/serviço/empresa, onde possivelmente farão uma propaganda positiva sobre.

### Cálculo

Para calcularmos o NPS®, a partir das respostas, devemos encontrar a porcentagem de Promotores e subtrair a porcentagem de Detratores, isto nos dará o Score.

$$\% \text{ Promotores} - \% \text{ Detratores} = \text{Score}$$

Vamos ao seguinte exemplo:

Você recebeu um total de <b>50</b> respostas	<b>100%</b>
5 respostas estavam na faixa de 0-6 ( <b>Detratores</b> )	<b>10%</b>
20 respostas estavam na faixa de 7 a 8 ( <b>Neutros</b> )	<b>40%</b>
25 respostas estavam no intervalo de 9 a 10 ( <b>Promotores</b> )	<b>50%</b>
<b>NPS = 50% - 10%</b>	
<b>NPS = 40</b>	

### Faixa de NPS

A partir do score obtido, podemos classificá-los em:

Faixa	Classificação
Menor que 0	<b>Péssimo</b>
De 0 a 49	<b>Ruim</b>
De 50 a 69	<b>Bom</b>
De 70 a 79	<b>Muito Bom</b>
De 80 a 100	<b>Excelente</b>

## Na Prática

### Como elejo minha base?

Usuários diretos (Squads, Cliente Final, RT, aqueles a quem destinamos valor).

### A partir de quantas respostas posso começar a considerar o score?

Quando pelo menos 1/3 da base escolhida tiver respondido.

### Com que frequência faço o envio?

O livro recomenda que seja a cada 90 dias, porém a frequência pode ser readequada de acordo com as necessidades do serviço/produto/organização.

### O que pergunto para Detratores durante a pesquisa?

O que o motivou a dar esta nota?

### O que pergunto para Neutros durante a pesquisa?

Como posso melhorar sua experiência?

### O que pergunto para Promotores durante a pesquisa?

Existe algo que poderia ser feito de forma diferente?

## Outcomes de se utilizar Feedback de Usuários

- Estreitar relacionamento com cliente/usuário
- Melhorar conhecimento sobre produto
- Melhorar conhecimento sobre o cliente
- Aumentar possibilidade de gerar incrementos e ações no backlog
- Fomentar melhoria contínua

## Desenvolvimento Individual

### Pilares de uma Equipe de Alta Performance

#### Pilar 1: Comunicação do Time

A comunicação efetiva do time é aquela em que o indivíduo sabe filtrar e organizar as informações que deve repassar e escolhe o canal ideal a ser utilizado, para transmitir a sua ideia. Ou seja, se pessoalmente, por e-mail, telefone ou videoconferência, por exemplo.

#### Pilar 2: Competência individual

As melhores equipes têm diversidade, ou seja, forças diferentes formam a equipe: pensamento estratégico, criatividade, organização, habilidades de relacionamento. Como você colabora com o que tem de melhor para a sua equipe?

#### Pilar 3: Comprometimento Geral

O comprometimento de todos os membros da equipe é fundamental – um com os outros, com a organização e com seu próprio crescimento individual.

Vale lembrar que comprometimento faz parte de um dos Valores do Scrum (comprometimento, coragem, foco, abertura e respeito)



## Pilar 4: Compartilhamento de Conhecimento

Equipes multidisciplinares aprendem e aprimoram tudo o que fazem. Os integrantes se preocupam com o desenvolvimento uns dos outros e implementam processos compartilhados e métodos de interação que facilitam o aprendizado e, por fim, levam a um melhor desempenho.

### Habilidades

Por mais que estejamos falando de time, cada indivíduo possui suas habilidades, competências, desejos e direcionamentos de carreira e estes devem ser respeitados. Sendo assim, não podemos levar como verdade que o indivíduo precisa ser multidisciplinar, porém dentro do time, disseminar estas habilidades nos leva a ter empatia com outras especialidades.

Exemplo:

Profissionais de UX conhecer o básico do trabalho dos engenheiros para gerar empatia no desenvolvimento de protótipos e telas.

Engenheiros conhecerem o básico do trabalho de um Product Owner para gerar empatia com as histórias escritas.

### Sugestões de Transmissão de Conhecimento

Como os membros de um time podem se complementar e aprimorar uns aos outros por meio da reunião de suas habilidades únicas? Podem ser separadas horas de conhecimento para o time. Abaixo alguns caminhos para essa transmissão:

- Palestras
- Workshops
- DOJOs
- Pair Programming
- Pílulas de Conhecimento
- Sessões práticas

Quando compartilhamos conhecimentos, aumentamos a capacidade do time e a longo prazo, com um objetivo claro e uma boa química isso impulsiona os resultados do negócio. A satisfação no trabalho aumenta, os colaboradores ficam envolvidos em seu trabalho e todos ganham.

### Profundidade de Conhecimento

Cada organização, carreira e indivíduo pode ter seu próprio entendimento de qual perfil de profissional é necessário para exercer seu trabalho. Existem algumas formas de definir a profundidade de conhecimento de um indivíduo, são elas:

#### I Shaped Skill

A letra 'I' representa um indivíduo especialista.

#### T Shaped Skill

A letra 'T' representa um indivíduo com uma especialidade e uma base de conhecimento expandida.

### Pi Shaped Skill

O símbolo 'π' representa um indivíduo com duas especialidades e uma base de conhecimento expandida.

### Comb Shaped Skill

O símbolo de um pente representa um indivíduo especialista em 'N' competências e uma base de conhecimento expandida.

## Matriz de Competências

A Matriz de competências é uma forma simples de visualizar e compartilhar nossas habilidades, conteúdos, ferramentas e processos.

### Quais os benefícios de uma matriz de competência?

#### Para a Equipe

A matriz de competência é basicamente uma ferramenta para ajudar a equipe como um todo. A equipe obtém uma visão geral rápida das habilidades presentes e das que estão faltando.

#### Para o Indivíduo

A matriz de competência ajuda o indivíduo a evoluir. Em primeiro lugar, dá ao insight individual de suas próprias competências e o que elas trazem para o grupo. É um excelente ponto de partida para oportunidades de aprendizado e desenvolvimento.

#### Para a Organização

Em uma escala maior, a organização obtém uma visão geral das competências e áreas de melhoria disponíveis.

#### Para os Clientes

Clientes, recebem um melhor serviço. Não importa realmente se são clientes internos ou externos. As equipes aumentam habilidades que ajudam a evitar armadilhas.

## Small Batches

### O que faz parte do lote de trabalho?

São requisitos, designs, código, testes e outros itens de trabalho que se movem pelo sistema

### Diferenças entre Big e Small Batches

Quando falamos de diminuir o tamanho do lote, significa efetuar a quebra destes itens, de uma forma que faça sentido para que, o item seja pequeno o suficiente para transitar pelo fluxo e grande o suficiente para gerar valor.

### Processo com lote grande

- Mudanças e correções no lote são difíceis ou impossíveis
- Apenas o lote completo pode ser entregue, se o lote não estiver completo, sua entrega será adiada

## Processo com Lote Pequeno

- Mudanças e/ou correções são fáceis
- Flexibilidade para começar um lote diferente do planejado
- Lotes completos podem ser entregues On Demand
- Lotes incompletos podem esperar o próximo deploy

## Ganhos na otimização de lotes

Lotes pequenos fluem mais rápido e com menor variabilidade, possibilitando aprendizado contínuo.

Podemos sumarizar os ganhos em:

- Aumento na produtividade da Squad
- Maior eficiência
- Redução do chaveamento de contexto
- Infraestrutura e automação:
  - Integração contínua
  - Automação do ambiente de construção
  - Testes de regressão automatizados

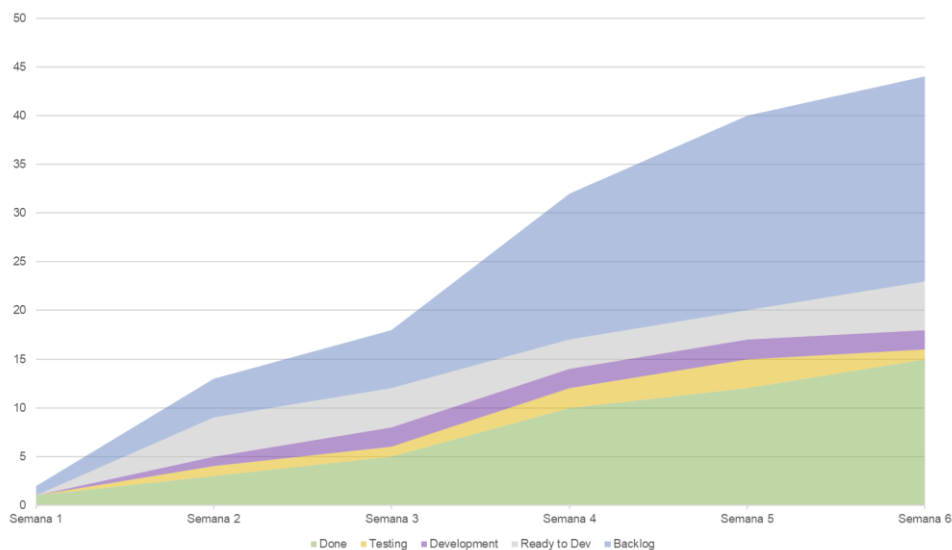
## Gargalos

Temos que um gargalo se forma quando a demanda de trabalho é maior que a capacidade de execução.

Quando encontrarmos um possível gargalo em nosso fluxo de trabalho devemos avaliar a etapa anterior e a etapa posterior e tentar identificar problemas e possíveis causas, a partir disso podemos aplicar pequenos ajustes para eliminar ou diminuir o gargalo encontrado.

## CFD – Cumulative Flow Diagram

Para analisar a saúde do fluxo, podemos utilizar o CFD, que em sua tradução significa Diagrama de Fluxo Cumulativo, que nos mostra a quantidade de itens acumulados ao longo do tempo, diferenciando as etapas do nosso fluxo.



# Fluxo e Limite de WIP

## UpStream e DownStream na Agilidade

Upstream são as etapas do fluxo de trabalho que tem o objetivo de amadurecer e validar ideias antes de aplicá-las no mundo real.

Já o Downstream se refere à todas as etapas seguintes do fluxo de trabalho a partir do backlog de itens gerados no Upstream, ou seja, podemos dizer que o Downstream é onde encontramos o Delivery.

## O que é WIP?

<b>Work In Progress = Trabalho em Progresso</b>
-------------------------------------------------

- WIP tudo aquilo que está dentro do fluxo de trabalho do time
- Já começou a ser criado/desenvolvido e não está de fato pronto.
- WIP é qualquer trabalho iniciado na esteira de desenvolvimento da squad até ser concluído.

## Objetivos de Limitar o WIP

- Diminuir o tempo de espera (waiting time) e consecutivamente o leadtime
- Mapear limite do WIP de acordo com as habilidades da equipe
- Trabalhar de forma colaborativa, reduzindo a ociosidade
- Desenvolver outras habilidades (promover a multidisciplinaridade)
- Proteção de cultura de engenharia sustentável
  - Quantidades menores
  - Produtos melhores
  - Código mais saudável
  - Chaveamento de contexto - Reduz o tempo de mudança entre uma atividade e outra

## Efeito Zeigarnik

Bluma Zeigarnik percebeu que garçons lembravam melhor dos pedidos cujas contas ainda não haviam sido pagas. Assim que eles recebiam o pagamento, a informação desaparecia das suas mentes. A partir desta percepção, Bluma chegou à seguinte conclusão, nós lembramos das tarefas não concluídas com mais facilidade do que aquelas completadas. Um **sistema de tensão** será criado quando o indivíduo receber uma tarefa para realizar. Quando a tarefa for concluída a **tensão desaparecerá**.

## Estimativa

### Story Points

**Story point** é uma medida abstrata que utiliza a História como insumo para representar o esforço do time para realizar a implementação da solução.

É um número que representa a opinião consensual do time sobre Esforço, Complexidade, Tamanho e Risco.

## Planning Poker

Quando vamos rodar um Planning Poker pela primeira vez, vamos precisar definir uma **âncora**, que será um ponto de partida para as estimativas. Esta âncora deverá ser um denominador comum para o time, que todos saibam pelo menos como funciona (exemplo, um CRUD ou uma API simples), esta âncora deverá ficar no meio da escala escolhida e a estimativa deverá se basear no quão mais ou menos Esforço, Complexidade, Tamanho e Risco aquele item tem em relação à âncora.

A sequência da dinâmica segue da seguinte forma:

1. O PO apresenta um item do backlog, explica o objetivo e o valor dele
2. Após a explicação, todas as dúvidas do time são tiradas e a estimativa se inicia
3. Cada membro produtivo do time seleciona, de forma oculta, a carta que melhor classifica aquele item
4. Quando todos tiverem escolhido, as cartas serão mostradas
5. Os participantes que apresentaram as cartas de maior e menor pontuação, exibem seu ponto de vista e o porquê escolheram a pontuação
6. Após a explicação, o time todo joga novamente, até haver um consenso ou equilíbrio evidente sobre o item
7. Quando a pontuação estiver definida, podemos seguir para um próximo item e recomeçar

## Indicadores e Métricas

### Velocity

**Que pergunta essa métrica responde?**

Qual é a média de pontos entregues em um determinado período?

### Capacity

**Que pergunta essa métrica responde?**

Qual o total de tempo de engenharia disponível para determinada iteração?

### Throughput

**Que pergunta essa métrica responde?**

Qual a quantidade de itens prontos (que habilita ou entrega valor para o cliente) entregues em um determinado período?

### Lead Time

**Que pergunta essa métrica responde?**

Qual a média de tempo (a partir do ponto de compromisso) que disponibilizamos os itens prontos para nosso cliente?

O cronômetro do lead time começa a ser contado a partir do momento que o item passa o nosso ponto de compromisso. No mercado, geralmente o ponto de compromisso é colocado a partir do momento que se inicia o Desenvolvimento

## Cycle Time

### Que pergunta essa métrica responde?

Quanto tempo tem durado cada etapa do fluxo para produzir e entregar soluções?

## ACCELERATE: Four Key Metrics

### MTTR

Esta métrica está relacionada à qualidade e estabilidade da solução entregue. Normalmente produtos e serviços de softwares são complexos para atender as necessidades existentes. Entretanto, sabemos que em muitos casos podem ocasionar problemas e prejudicar a usabilidade. Então, qual é a capacidade de tempo da organização em conseguir resolver isso de maneira mais rápida? Essa resposta está na métrica MTTR, que é a média (em um período analisado) de tempo para realizar a correção do software e estabilizá-lo.

### Deploy Frequency

Como a métrica anterior, esta também está relacionada a velocidade do time. Porém, não é apenas isto. Em software quando falamos em diminuir o tamanho da solução que será entregue, estamos falando diretamente no conceito apresentado pela Toyota, com pequenas entregas e em maior frequência conseguimos reduzir o tempo do ciclo desta atividade, aceleramos o feedback, reduzimos os riscos e sobrecarga, custos e aumentamos a produtividade do time

### Change Fail Rate

Como a anterior, esta métrica também está relacionada à qualidade e estabilidade. No caso é a porcentagem da taxa de falha em produção causada por uma nova entrega. Ou seja, quando tem uma implantação de produção e devido a isso ocorre alguma falha e necessidade de correção. O cálculo é feito com relação às implantações realizadas naquele período analisado.

### Lead time for Changes

Essa métrica está relacionada ao indicador de velocidade do time, identificando o tempo de criação para a conclusão de uma atividade. Ou seja, o tempo entre criar uma atividade, desenvolvê-la e concluí-la em produção. Não necessariamente que, quando a atividade é criada, imediatamente ela seja desenvolvida. Em metodologias ágeis, quando criamos uma atividade ela entra no backlog do time, esperando para ser desenvolvida.

# THANK YOU

**Time de Agilidade do CoE e NTT DATA + DevOps**

Markus Fantone

Renan Pires Calcanhoto

Wagner Alencar