

Spotify & Resultados Digitais

Alexandre Torres 13/0099767

Eduardo Brasil 11/0115104

Matheus Mello 11/0017692

Spotify

Introdução

Introdução

- O Spotify desenvolveu uma abordagem alternativa para a metodologia ágil que gerou diversos resultados positivos para a empresa. Dessa forma, demonstraremos aqui, como essa empresa construiu uma das melhores culturas de engenharia e como ela funciona.

Origem

- Em 2009, um pequeno grupo de funcionários do Spotify decidiram criar uma cultura para a empresa diferente do que já existe na maior parte das companhias.
- O Spotify estava se expandindo rapidamente, e precisavam de uma nova metodologia.

Origem

- Muitos dos engenheiros da equipe foram para conferências, leram livros, conversaram com outros engenheiros de empresas como Twitter, Netflix e Google
- Para construir essa cultura que apresentaremos, foi necessário realizar inúmeros experimentos.

Características da cultura do Spotify

Características interessantes

Ágil é mais importante que Scrum (ou qualquer metodologia), princípios dos ágeis são mais importantes do que qualquer prática específica.

Agile > Scrum
Principles > Practices

Não utilizam nenhuma metodologia a risca, mas se baseia em várias.

Com o crescimento da empresa, algumas práticas comuns de equipes ágeis como "Sprint Planning Meeting", "Task Breakdown", "Velocity", "Burndown Chart" começaram a atrapalhar, dessa forma passaram a ser opcionais.



Características interessantes

Foco na Autonomia

Um sistema muito centralizado dificulta a escalabilidade. Dessa forma é necessário autonomia das equipes.

Vantagens:

- *Equipes mais motivadas e ágeis (decisões são tomadas localmente e, portanto, são mais rápidas).*
- *Minimiza-se entregas de uma equipe para outra, e a espera de uma pela outra. Diminuindo a dependência entre as equipes é mais fácil de expandir a companhia como um todo.*

Características interessantes

Foco na Autonomia

Exemplo - As SQUADS (Equipes)

- Uma Squad é uma equipe pequena, com diversas funções, que gerencia a si própria, se organiza. Normalmente, é formada por menos de 8 pessoas.
- Cada Squad tem uma missão específica, podendo ser, inclusive, algo técnico. Por exemplo, “fazer com que Spotify seja a melhor plataforma para descobrir músicas”.
- Cada uma das Squads tem responsabilidade de realizar as tarefas relacionadas ao que estão desenvolvendo do início a fim, ou seja, “design”, “commit”, “deploy”, “maintenance”, “operation”.

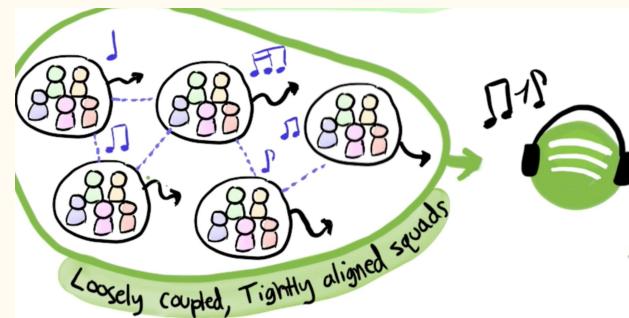


Características interessantes

Alinhamento

Em conjunto com a Autonomia é necessário Alinhamento. Não adianta autonomia se as equipes seguem "caminhos" diferentes. Todas devem caminhar com o principal objetivo (o bem da organização) em mente.

- *As Squads devem estar alinhadas para atingir objetivos que são interesse comuns da empresa.*
- *"Squads pouco acopladas, mas muito alinhadas".*
- *Missão do spotify é mais importante do que a missão de uma específica squad.*

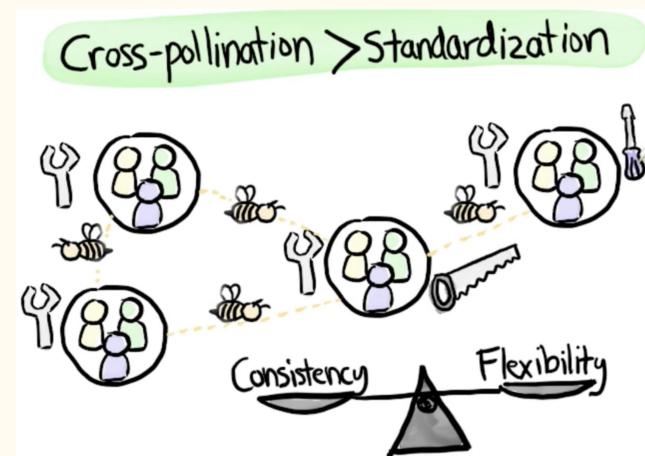


Características interessantes

Cross-Pollination (Polinização Cruzada)

As squads escolhem quais práticas ou ferramentas serão utilizadas, não é exigido padronização. Porém, quando muitas começam a utilizar certas ferramentas ou práticas, outras squads, normalmente, também começam a adotá-las, um exemplo é o GIT.

Dessa forma, apesar da flexibilidade, também existe consistência entre as Squads.



Características interessantes

Pessoas são muito importantes

Foco na colaboração, troca de conhecimento, satisfação dos funcionários.

Exemplos:

- Utiliza-se "peer code review". Com isso a qualidade é aumentada, e o conhecimento é compartilhado.
- Alta preocupação com a satisfação dos funcionários.
- Os ambientes são voltados para colaboração, dessa forma, os integrantes da squad trabalham próximos uns aos outros, podendo ver a tela dos colegas. Grande parte das paredes são quadros brancos, onde podem compartilhar ideias.

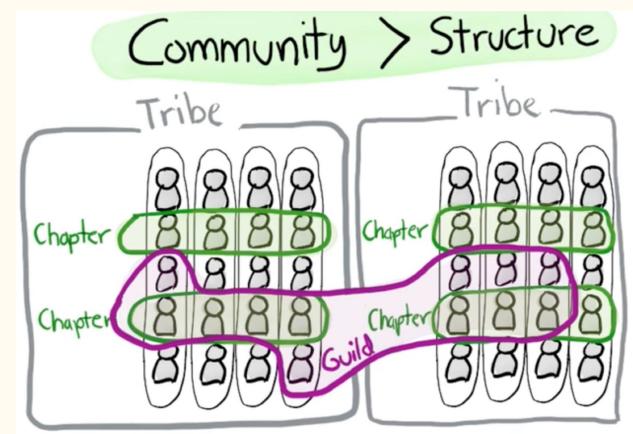


Características interessantes

Comunidade ao invés de estrutura hierárquica

Tenta-se fazer com que todos os funcionários se sintam numa comunidade, onde todos são "igualmente" importantes.

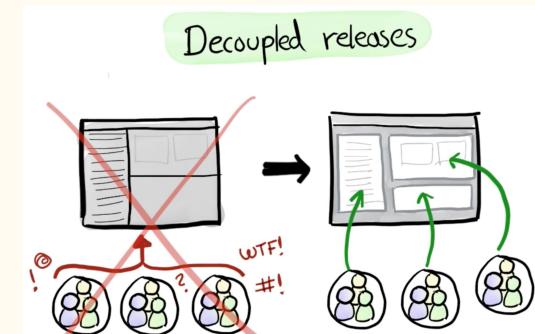
- *As squads são agrupadas em Tribos.*
- *Também utilizam Guilds(grupo de pessoas com os mesmos interesses), que visa compartilhar conhecimentos sobre determinado assunto.*



Características interessantes

Arquitetura desacoplada

- Composta por uma grande quantidade de sistemas (acima de 100) independentes, que focam em necessidades específicas.
- Tenta-se tornar esses subsistemas pequenos e desacoplados, com uma interfaces bem definidas.
- Cada squad é responsável por um ou mais desses módulos e, normalmente, é ela quem mantém o módulo. Porém, o sistema como um todo é compartilhado entre às Squads, e nada as impede de realizar alterações em subsistemas de outras Squads. Apesar disso, posteriormente, devem pedir que essas modificações sejam revisadas.

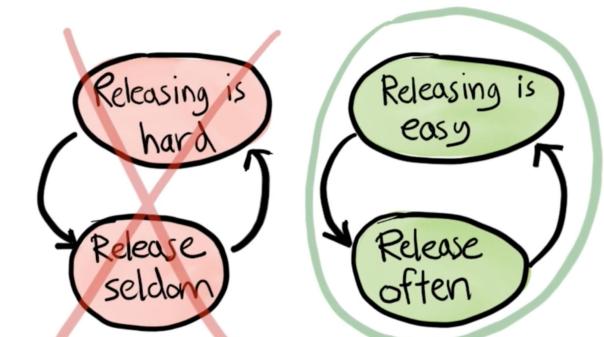


Características interessantes

Releases pequenas e frequentes

- Bastante investimento em testes automatizados e infraestrutura para releases contínuas.
- As releases são desacopladas, ou seja, cada Squad pode realizar a sua própria release (de um subsistema) independentemente das outras.

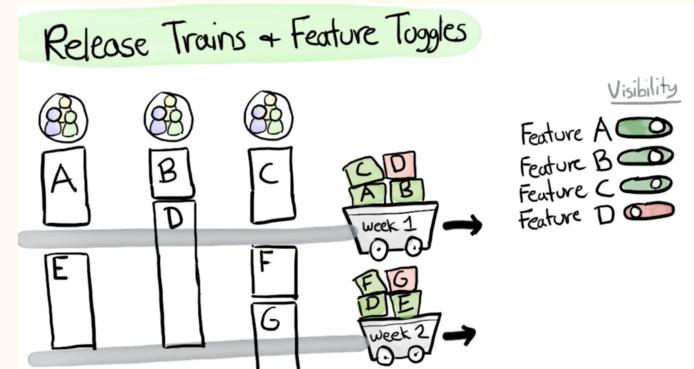
Small + frequent releases



Características interessantes

Release Trains + Feature Toggles

- Cada tipo de aplicação dos clientes tem um "trem de release" que segue um cronograma. Ou seja, a intervalo de tempo definido (como 1 semana), uma série de features será entregue para o cliente.
- Os trens contém as features feitas ou sendo feitas, mesmo se essas não estiverem concluídas.
- Às features não acabadas são escondidas utilizando um "feature toggle". A vantagem disso é que problemas de integração são expostos cedo, minimiza a necessidade de utilizar novas "branches".



Características interessantes

Confiança é mais importante do que controle

Tendo confiança é possível reduzir burocracia e política dentro da empresa. Controle em excesso, atrapalha a escalabilidade da empresa. É interessante a seguinte frase:

"Ágil em escala necessita de confiança em escala"

Exemplo:



- Não há um controle formal, centralizado do que entrará em produção, às próprias squads decidem.

Características interessantes

Fail-friendly environment & Continuous Improvement

O Spotify tem um ambiente amigável a falhas e de melhoria contínua

- Ao falhar rápido, é possível aprender rápido e, consequentemente, melhorar rápido. Essa é uma boa estratégia para sucesso a longo prazo.
- "Nosso objetivo é errar mais rápido do que qualquer um" - Daniel Ek
- Porém é bom aprender algo com os erros, e para isso existe o postmortem, onde indica-se o que foi aprendido e quais mudanças seriam feitas.

Fail Fast → Learn Fast → Improve Fast



Daniel Ek

Failure Recovery > Failure Avoidance



Características interessantes

Limited Blast Radius & Gradual Rollout

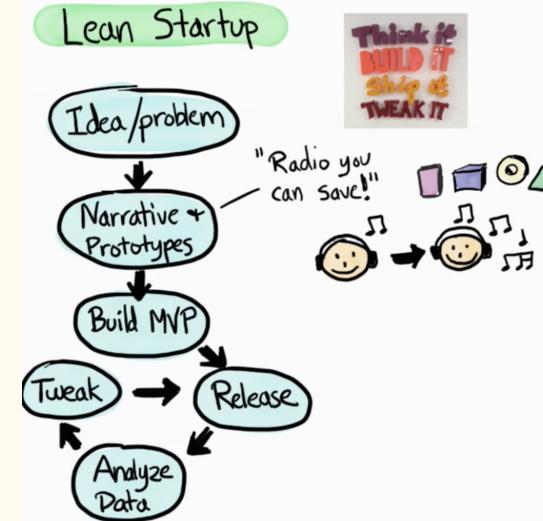
- Um erro tem um "raio" de impacto limitado ("Limited Blast Radius").
- Se uma Squad comete um erro, isso não pode afetar todo o sistema, deve afetar somente uma pequena parte do sistema, para isso é ideal ter um arquitetura desacoplada.
- "Gradual Rollout" está relacionado ao fato de que as alterações são entregues para um número pequeno de usuários de cada vez. Dessa forma, se houver alguma falha, o número de pessoas impactadas será pequeno.
- Isso dá mais coragem para as Squads realizarem mais experimentos e aprender rapidamente.



Características interessantes

Lean Startup

- Antes de criar algo, deve-se pensar sobre o assunto.
Será que as pessoas realmente querem aquilo?
- Depois pensa-se em como aquilo impactará positivamente o produto. E são criados protótipos.
- Posteriormente, é criado um MVP ("Minimum viable product"), para validar os conceitos.
- Depois é criado uma release para um pequeno número de usuários.
- Após isso, dados dos usuários são colhidos, para identificar se o impacto desejado foi atingido, caso negativo, são feitas alterações.

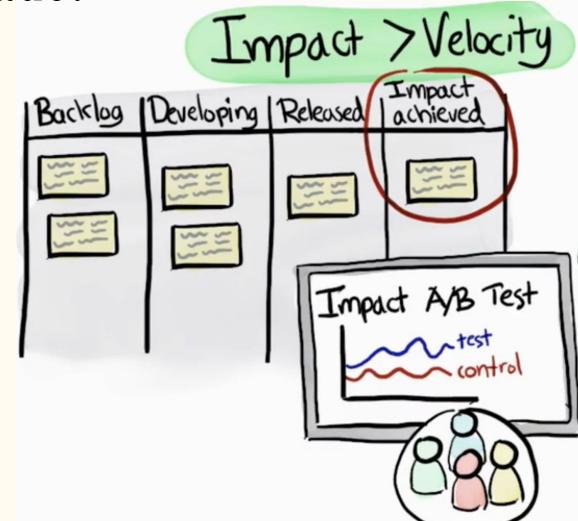


Características interessantes

Impacto mais importante que velocidade

O impacto de uma feature importa mais do que o quanto rápido ela será finalizada. Não tem problema atrasar a entrega da feature caso o intuito disso seja melhorá-la a fim de atingir o impacto desejado.

- *Uma feature não é considerada pronta até atingir o impacto desejado.*
- *Se o impacto desejado não tiver sido alcançado, são feitas novas modificações.*

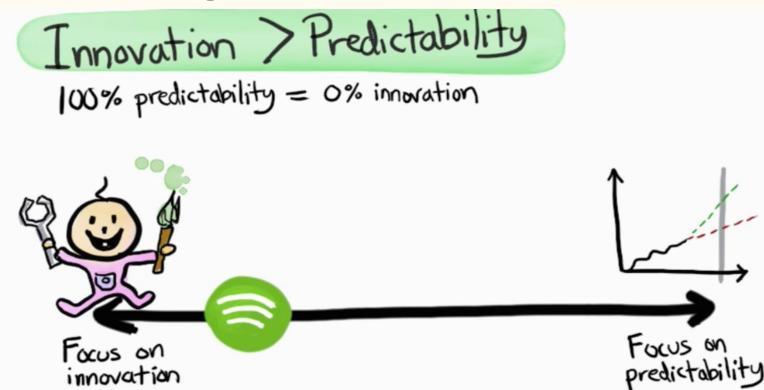


Características interessantes

Inovação mais importante do que previsibilidade

- O foco da organização não é em planejar o que deve ser feito, mas sim em inovar.
- Muito planejamento e previsibilidade impactam diretamente na inovação. Dessa forma, é interessante deixar os funcionários mais livres para inovar, criar, ao invés de seguirem um calendário rigoroso.

Exemplo de uma fala no vídeo: "Como sabemos o que será entregue daqui a um tempo determinado? A resposta curta é que em geral não sabemos."



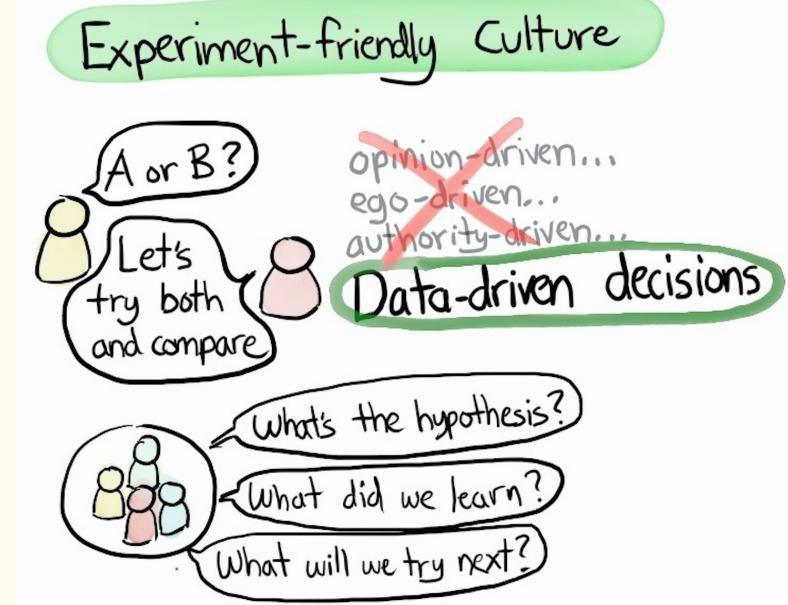
Características interessantes

Experiment-friendly culture

O Spotify possui uma cultura em que todos são incentivados a experimentar coisas novas.

Exemplos:

- *Não sabe qual ferramenta utilizar, testa as duas!*
- *Realmente precisamos do planejamento da sprint? Não sabemos, vamos experimentar algumas sprints sem planejamento!*



Características interessantes

Data-Driven Decisions (Decisões baseadas em Dados)

Deve-se tomar decisões baseando-se em dados.

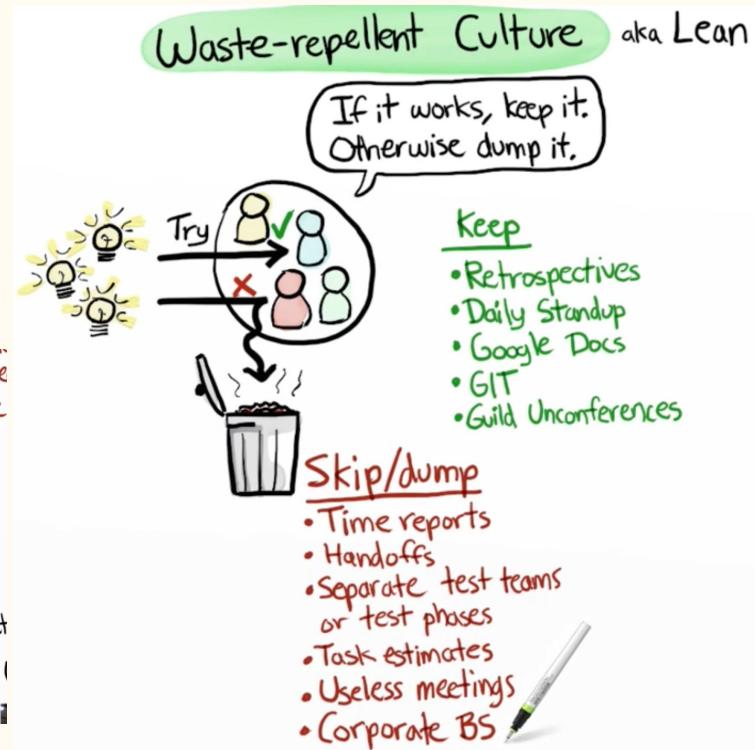
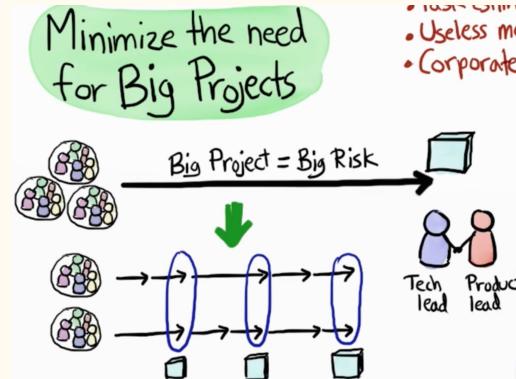
- Após realizar uma série de experimentos, é recomendado coletar dados para sabermos se houveram ou não melhorias.
- Dessa forma é possível evitar decisões tomadas por opiniões, achismo, ou até mesmo por autoridades.

Características interessantes

Waste-repellent Culture (Cultura que "repele" desperdício)

Ao testar novas coisas, o que funciona é mantido, o que é inútil ou atrapalha é descartado.

Um exemplo comum de desperdício são Grandes Projetos. Dessa forma, eles devem ser evitados. Mas caso necessário, devem ser utilizadas práticas que reduzam o desperdício. Como, quadros de progresso, reuniões diárias, e "demos" semanais. Essas práticas melhoram a colaboração e feedback.

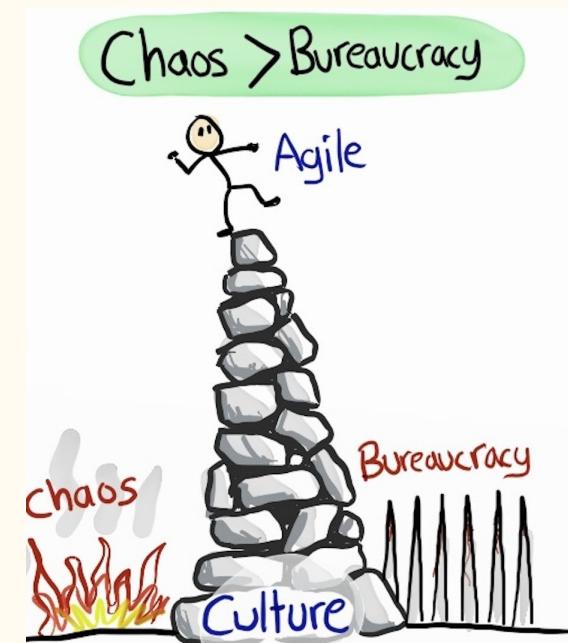


Características interessantes

Caos vs Burocracia

Quando uma organização cresce, ela corre o risco de se tornar caótica. Para evitar isso, é necessário algum nível de estrutura e utilização de processos (que geram burocracia). Como ambos causam desperdício, procura-se alcançar a quantidade ideal de cada um.

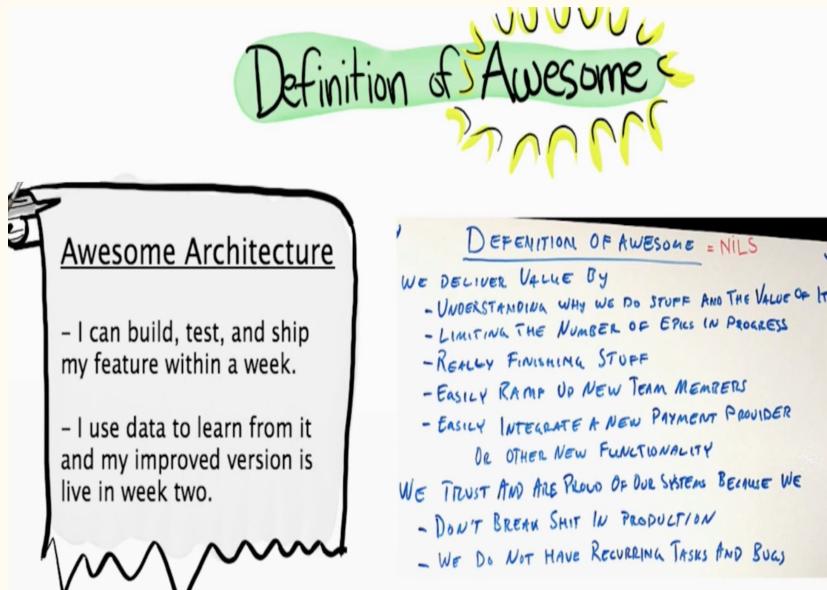
- Qual o mínimo de burocracia (estrutura e processos) possível que garanta que a empresa não fique em uma situação caótica? Isso seria o ideal.



Características interessantes

Definition of Awesome (Definição de Espetacular)

- As squads definem o que seria perfeito para elas, com relação ao produto, processo, e tentam mover-se nessa direção.
- Serve como um guia, direcionando a equipe.
- Num “mundo” ideal, como tudo seria?



Características interessantes

Cultura Saudável cura um processo Ruim

Uma empresa com uma boa cultura irá continuamente identificar os problemas em seus processos e melhorá-los.

É necessário, portanto, sempre manter uma cultura saudável. O Spotify se esforça bastante em melhorar sua cultura continuamente.



Conclusão

Conclusão

- 1) O foco desta apresentação é na **cultura** do spotify, justamente pelo fato dela ser considerada mais importante do que qualquer processo, ferramenta, ou técnica.

"Healthy culture heals broken process"

Conclusão

2) Outro ponto igualmente importante está relacionado a adequação de metodologias Ágeis para uma grande empresa ("Ágil em escala").

- Metodologias ágeis foram planejadas para serem utilizadas em pequenas equipes. Porém organizações como o Spotify conseguiram adequá-las para o uso em larga escala.
- Para que isso seja possível é necessário, principalmente:
 - Autonomia
 - Alinhamento *"Loosely coupled, Tightly aligned squads"*
 - Confiança *"Agile at scale requires trust at scale"*

Conclusão

3) **Experimentar**, colher dados e utilizá-los para tomar decisões é Essencial.

- Para chegar na cultura atual do Spotify, foram necessários muitos experimentos. E para que hajam novas melhorias também serão necessários novos experimentos.
- Só foi possível desenvolver este "ágil em escala" realizando inúmeros experimentos.
- A fim de evitar gargalos relacionados à metodologia, não é seguida nenhuma metodologia a risca. São testadas práticas de diversas metodologias, mas são utilizadas somente as que mais agregam valor e melhor se encaixam na organização.

Resultados Digitais

Introdução

- Resultados Digitais é uma empresa no ramo de Marketing Digital que teve um crescimento impressionante e rápido. Tal crescimento levou a empresa a adaptar-se e evoluir a sua estrutura baseando-se na metodologia Spotify que vimos anteriormente.

2015

Em 2015 a Resultados Digitais era composta por:

- 100 Pessoas
- 1200 Clientes
- 1 Product Manager

2017

Hoje a Resultados Digitais é composta por:

- 100 Pessoas só na área de produto
- 12 Product Managers
- Mais de 7000 Clientes

Um crescimento e tanto para apenas 2 anos, vamos ver que mudanças a Resultados Digitais adotou para comportar todo esse crescimento.

Estrutura e Problema

Estrutura em 2015

Em 2015, o time de produto era composto por 30 pessoas divididas em 4 times para o desenvolvimento do software e 1 time para sistemas internos. Os Product Managers ficavam separados do times e existia o Kanban do Produto que continha os Épicos que estavam rodando nos times. Esse kanban ajudava a organizar e priorizar as funcionalidades e montar o backlog de cada time.

Problema

Essa estrutura enxuta funcionava muito bem mas começou a apresentar problemas conforme o time crescia:

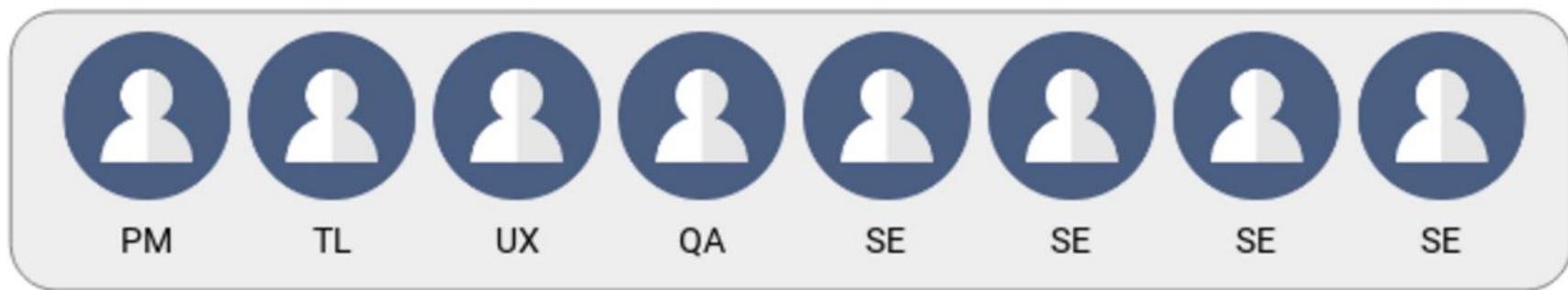
- Cada vez mais demanda e menos visão do que iria ser feito nos próximos meses.
- Aumento no número de *Quick Wins* ou Histórias sem Épicos
- Falta de tempo para estudar um ou dois problemas complexos ao mesmo tempo

Nova Estrutura e Solução

Times Especialistas

- Foi adotada a estrutura de times especialistas (Squads) que funcionavam como uma mini Startup. Cada Squad era formada por:
 - 1 Product Manager
 - 1 Tech Leader
 - 1 Designer
 - 1 Quality Assurance
 - 2 a 4 Engenheiros de Software

Squad



Benefícios

- A estrutura de Squads trouxe vários benefícios
 - Autonomia sobre determinada temática
 - Todas as decisões, de Roadmap a como gerenciar o Suporte, eram tomadas pelo próprio time
 - Possibilitou que os Product Managers pudessem mergulhar em um assunto específico

Resultados

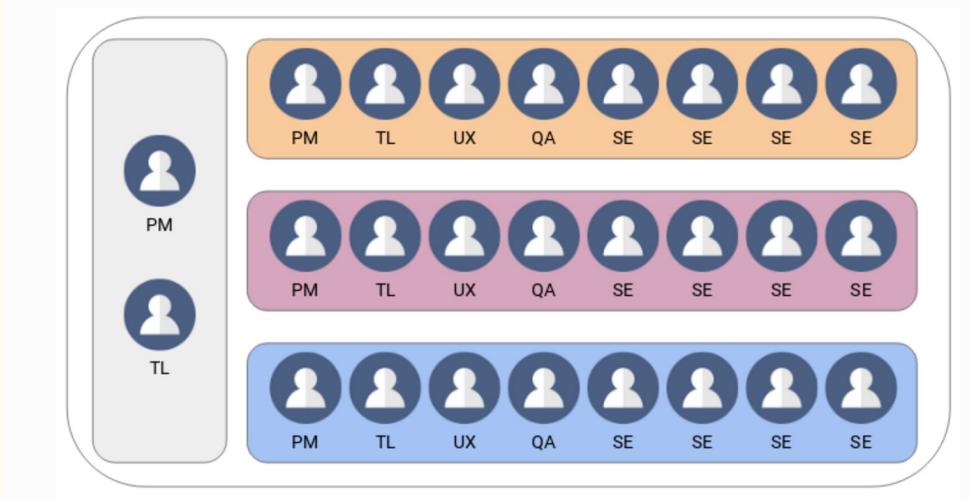
- A estrutura de Squads foi inspirada no modelo Spotify e serviu de base para a escalabilidade e evolução da empresa
- Em 2016 a Resultados Digitais já tinha 12 times na área de produto e 12 PMs no time

Novos Problemas

- Mesmo com esses benefícios, conforme a equipe aumentava, novos problemas apareciam:
 - Times com a mesma temática
 - Times compartilhando problemas, soluções, roadmaps, tickets de suporte e até pessoas em algum momento

Tribo

- A partir desses problemas, criaram-se as tribos, uma estrutura que agrupa times de uma mesma temática. Cada tribo é liderada por um Tech Leader de Tribo e um Product Manager de Tribo.



Objetivos da Estrutura de Tribos

- Garantir a autonomia da tribo em todos os sentidos:
 - Metodologias
 - Processos
 - Cerimônias
 - Budget e contratações
- Deixar os roadmaps alinhados e coerentes
- Manter uma gestão próxima e sem criar novos níveis hierárquicos

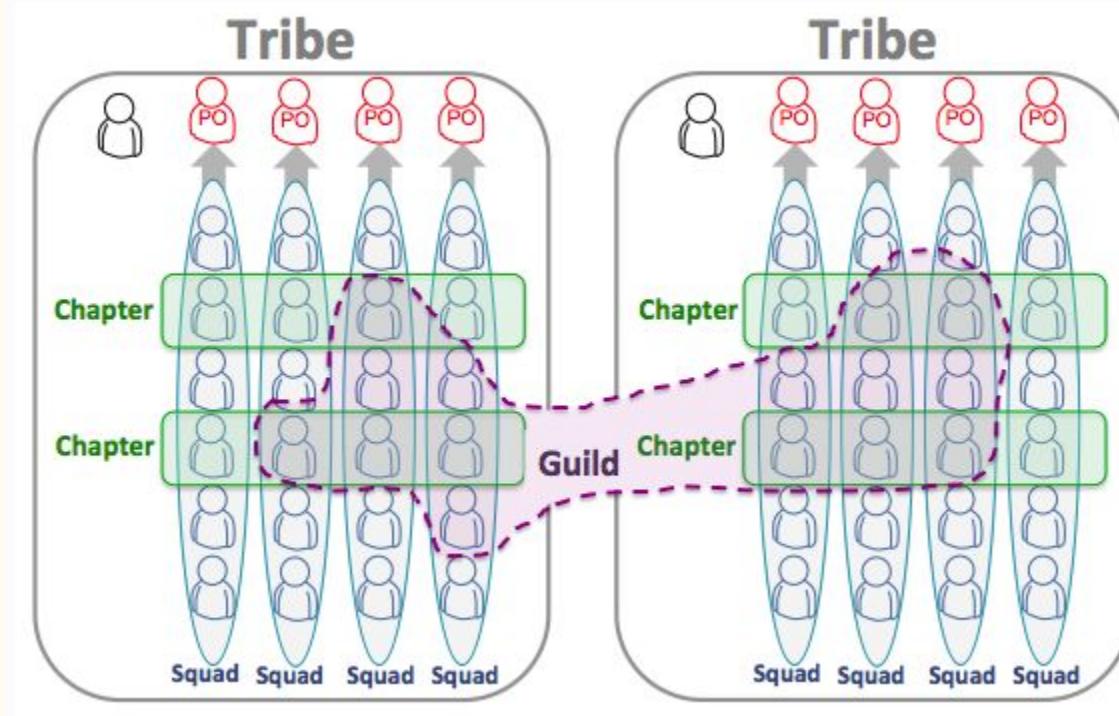
Chapters

“Na Spotify foi criado o conceito de “Chapter” (Divisão), um conjunto de profissionais com as mesmas habilidades e dentro da mesma área de competência, dentro da mesma tribo.

Existem quatro divisões, que por ventura podem ser relacionados a diferentes funções, por exemplo: uma divisão é composta por programadores, outra formada por analistas de teste, outra podendo ser os Agile Coaches e, por último, a dos Product Owners.”

Fonte: [Escalando o Agile na Spotify](#)

Chapters



Chapter

1 Chapter que reúne todos os QAs do time de desenvolvimento
de Produto

Chapter

Necessidade de interação entre os grupos de QAs para trocar conhecimento e experiências a ponto de evoluir toda uma área de Qualidade dentro da empresa.

definido em intervalos regulares para uma discussão de “Experts” de Qa e suas dificuldades

Futuro

A empresa funciona atualmente neste formato contando com 13 times de produto e 1 chapter de QAs, mas é provável que essa estrutura se altere com o surgimento de novos problemas e desafios.