



Topicos Especiais - SAFe

Giovani Salvador

Scaled Agile Framework - SAFe

Pesquisa VersionOne

35%

Com mais de 5000 pessoas na
organização inteira

Com mais de 1000 pessoas na
área de desenvolvimento de
software

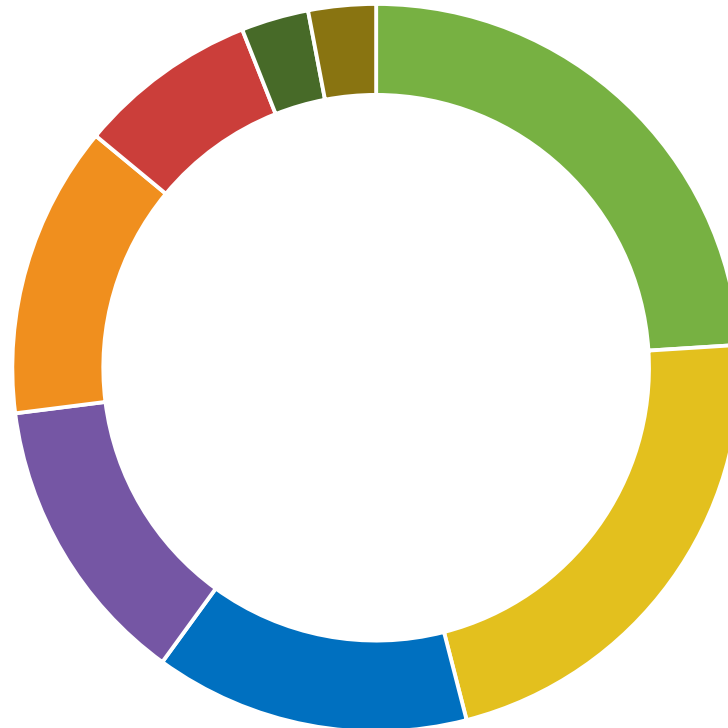
53%

20%

Trabalham em mega organizações,
com mais de 20.000 empregados

Pesquisa VersionOne

Quem respondeu



■ Gerentes de Projeto/Programa ■ Desenvolvedores ■ Dev Lead ■ Scrum Master ou coach
■ Product Owner/BSA ■ Consultor/instrutor ■ IT Staff ■ C-Level (diretores, executivos)

Pesquisa VersionOne



	2014	2013
< 1 ano	15%	8%
1-2 anos	29%	40%
3-5 anos	32%	33%
>5 anos	24%	19%

Pesquisa VersionOne

53%

Afirmam que a maioria dos projetos com ágil tem sucesso. Alguns afirmaram que todos.

O percentual de empresas usando métodos ágeis em times distribuídos, desde 2012

35%

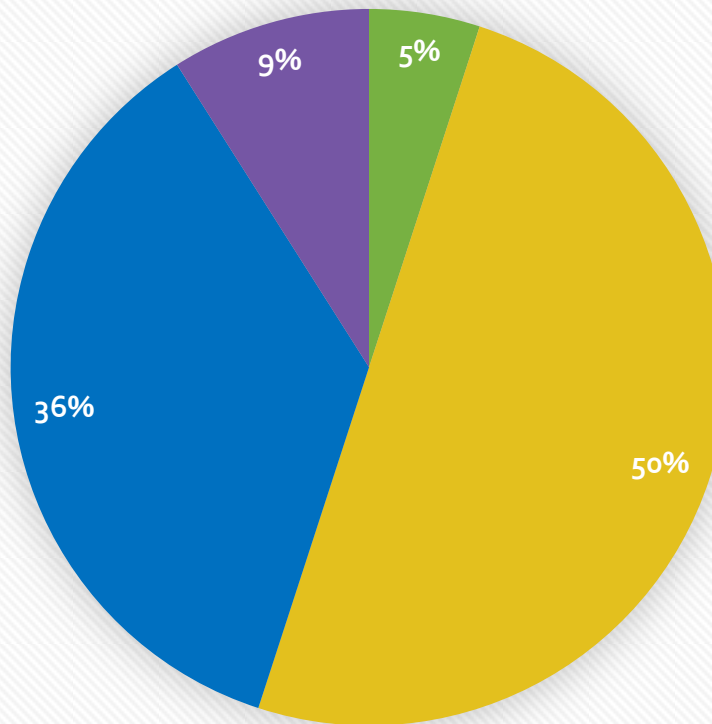


80%

56%

Usam Scrum como framework ágil

Uso de ágil nas empresas



■ Nenhum time usa ágil

■ Menos da metade de nossos times usa ágil

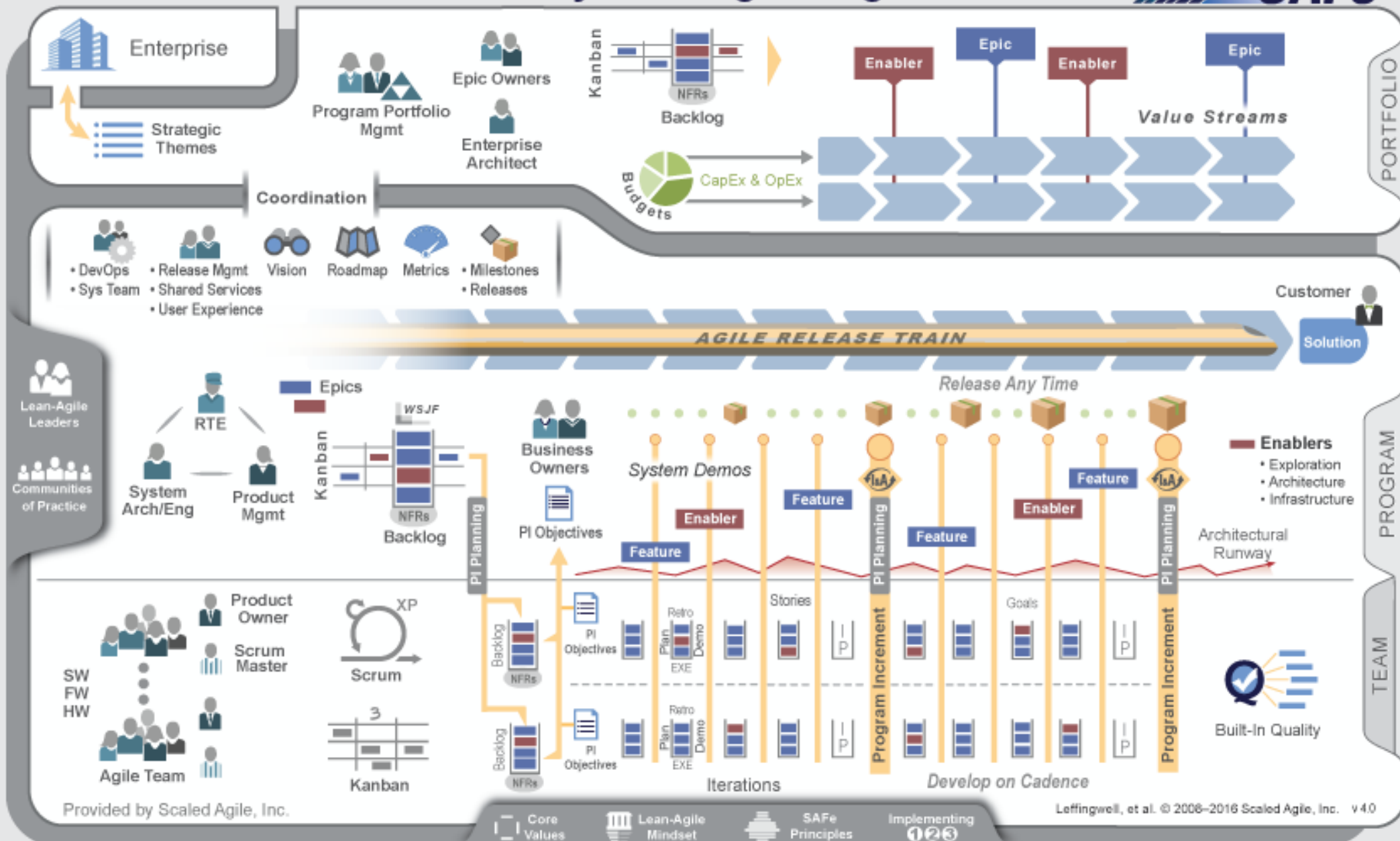
■ Mais da metade de nossos times usa ágil

■ Todos nossos times são ágeis

SAFe

Scaled Agile Framework

SAFe® 4.0 for Lean Software and Systems Engineering



Princípios

#1-Take an economic view

#2-Apply systems thinking

#3-Assume variability; preserve options

#4-Build incrementally with fast, integrated learning cycles

#5-Base milestones on objective evaluation of working systems

#6-Visualize and limit WIP, reduce batch sizes, and manage queue lengths

#7-Apply cadence, synchronize with cross-domain planning

#8-Unlock the intrinsic motivation of knowledge workers

#9-Decentralize decision-making

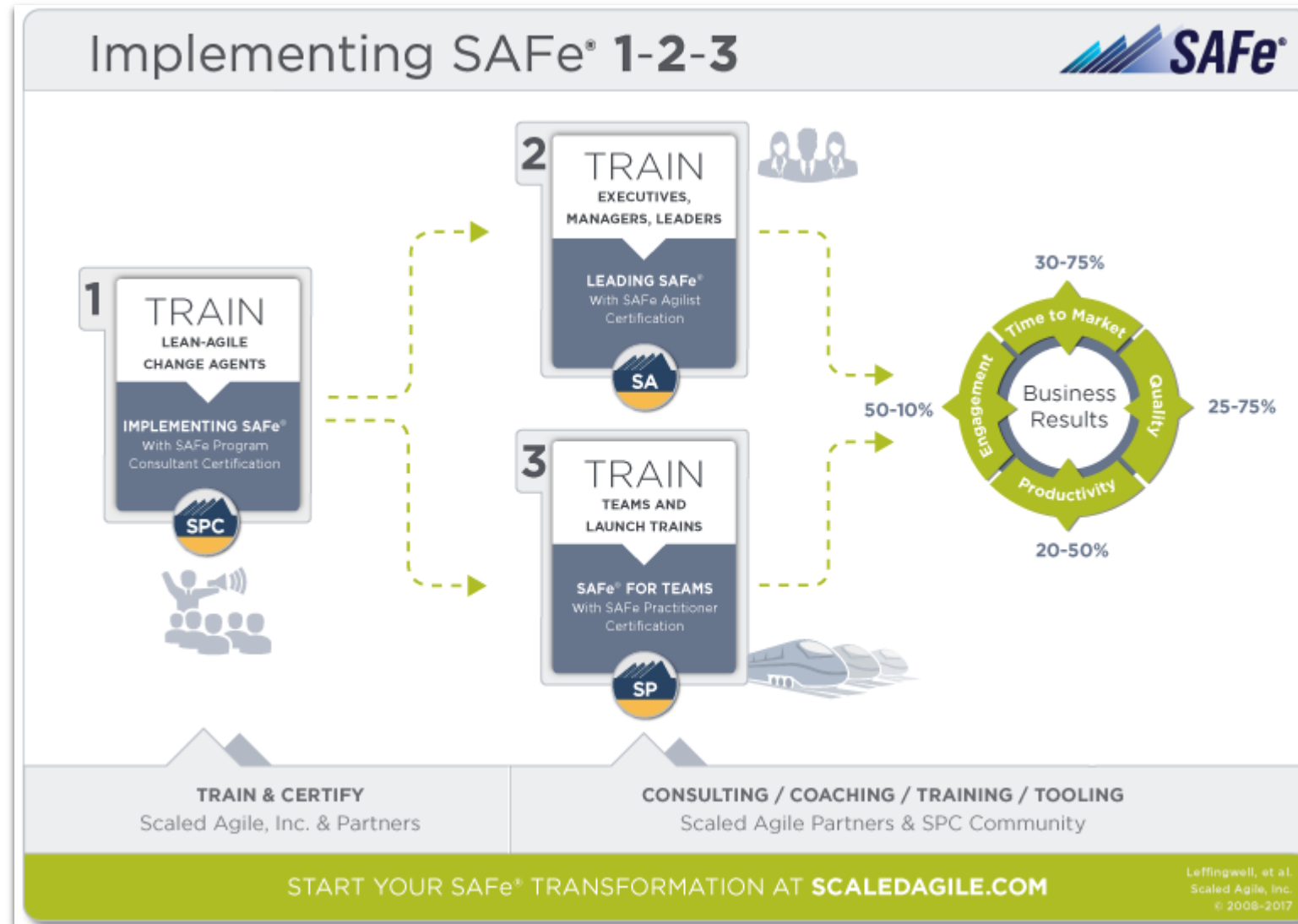
Balanceamento entre custo de desenvolvimento, valor da entrega, visão de longo prazo, etc.

Iterações curtas

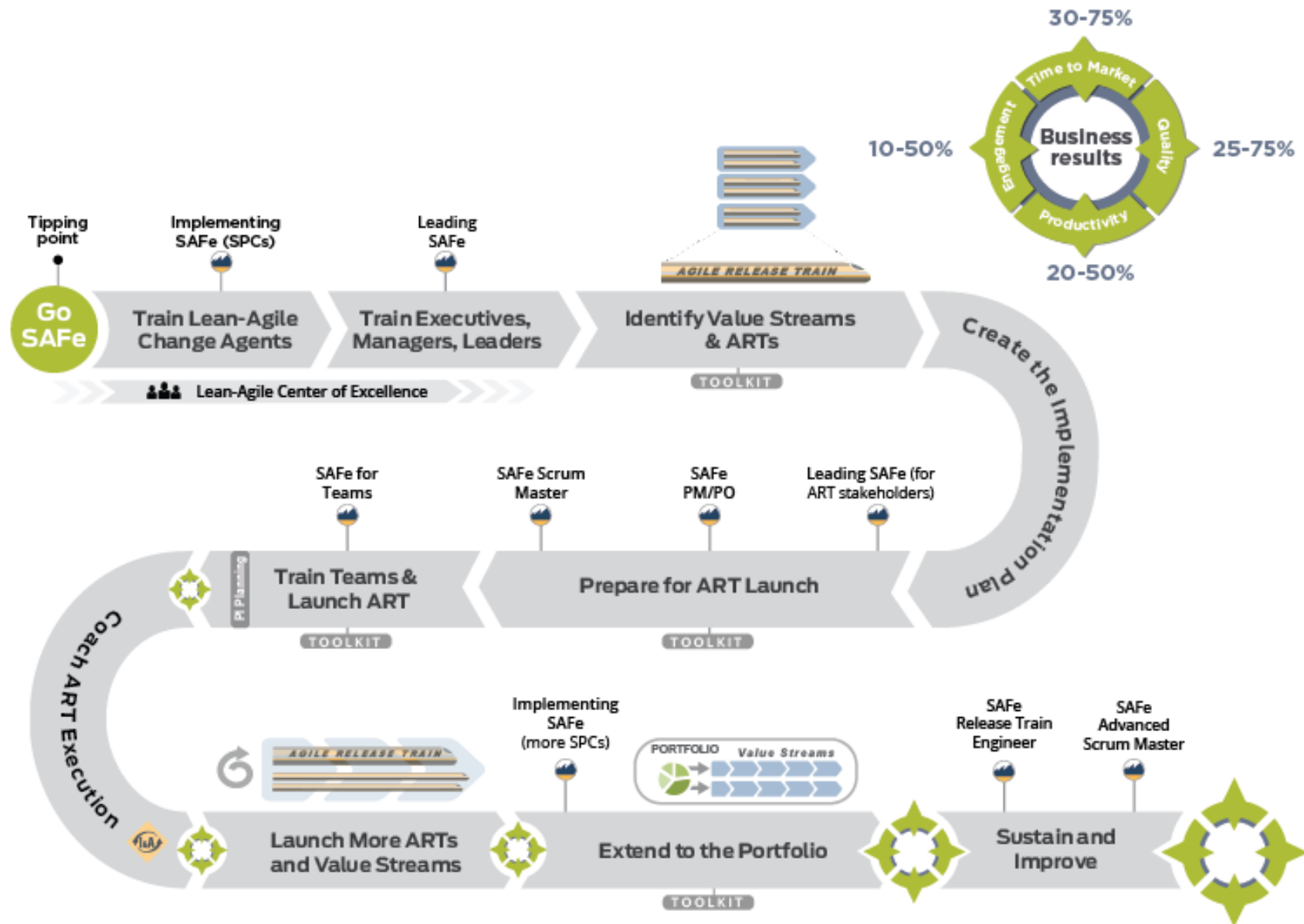
Validações mais frequentes entre domínios de trabalho

Escalar decisões leva tempo

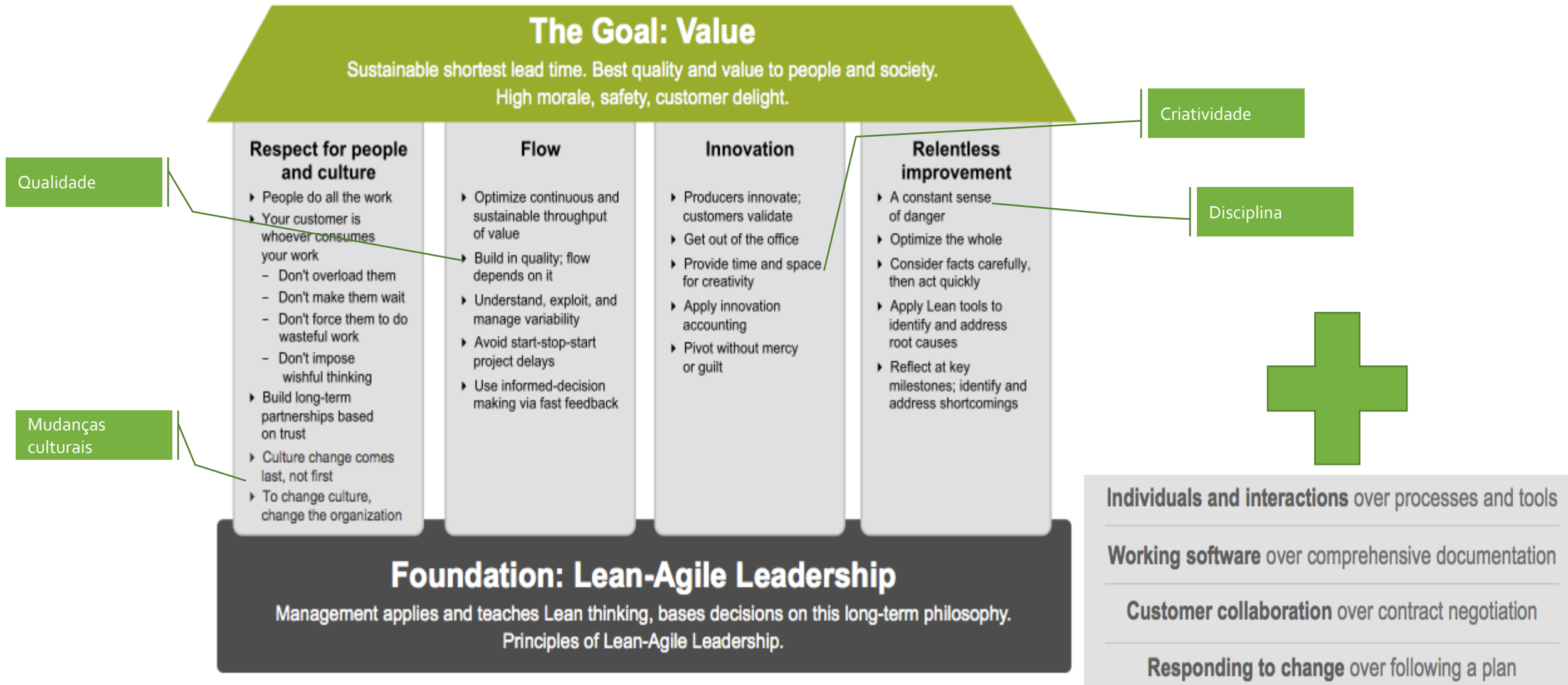
implementando



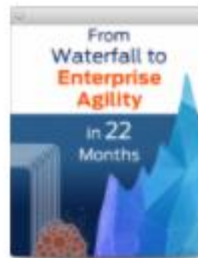
implementando

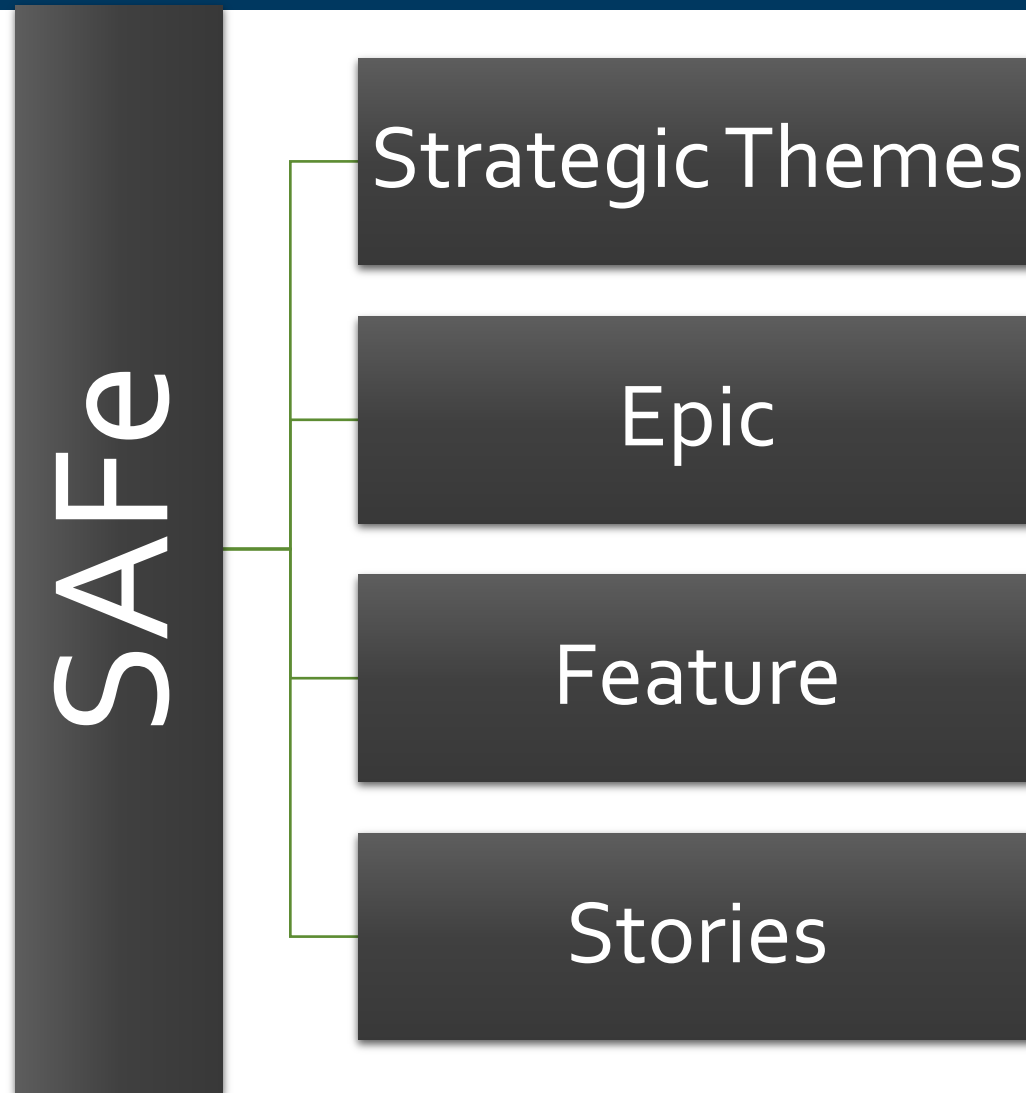


Lean Agile Mindset

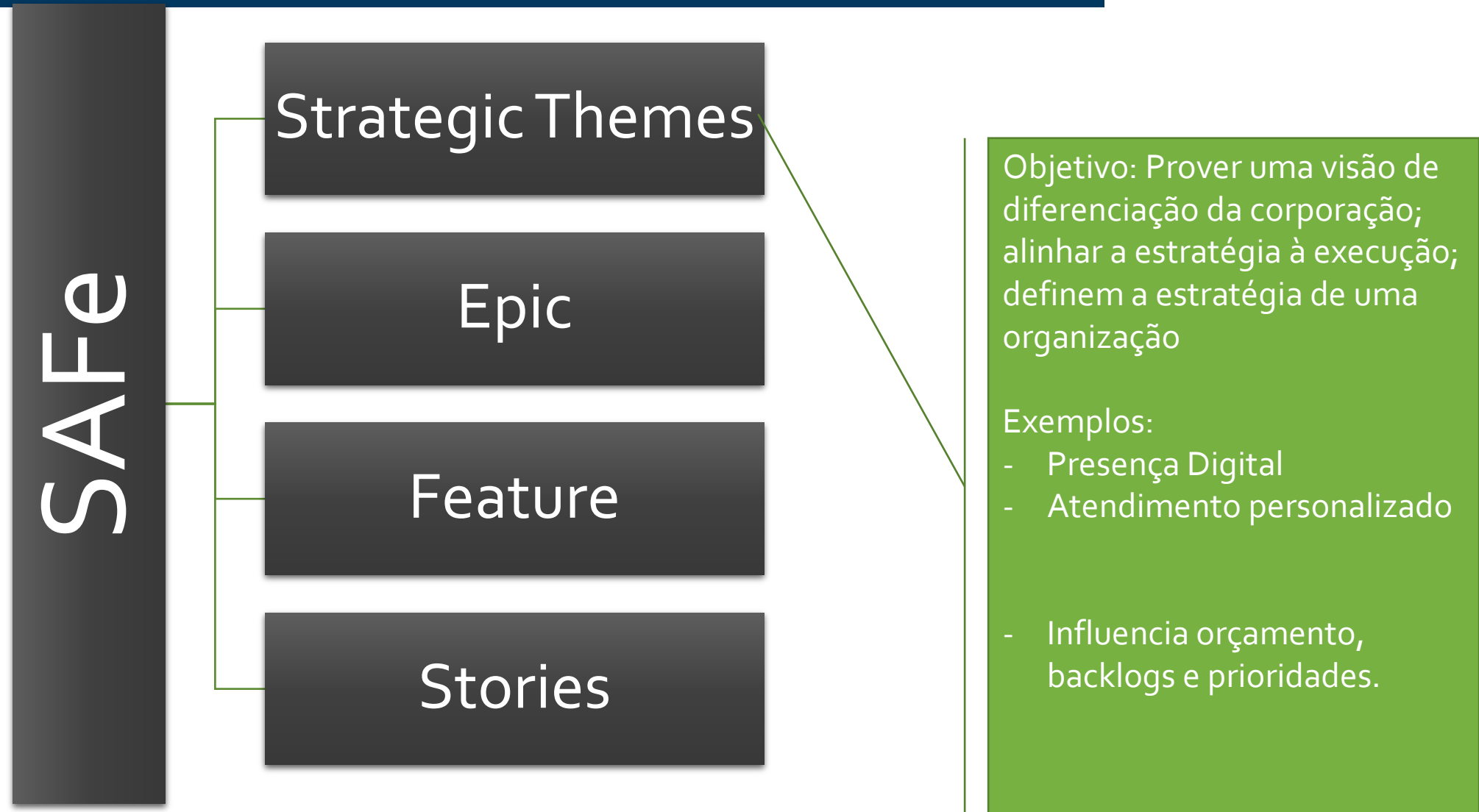


Cases

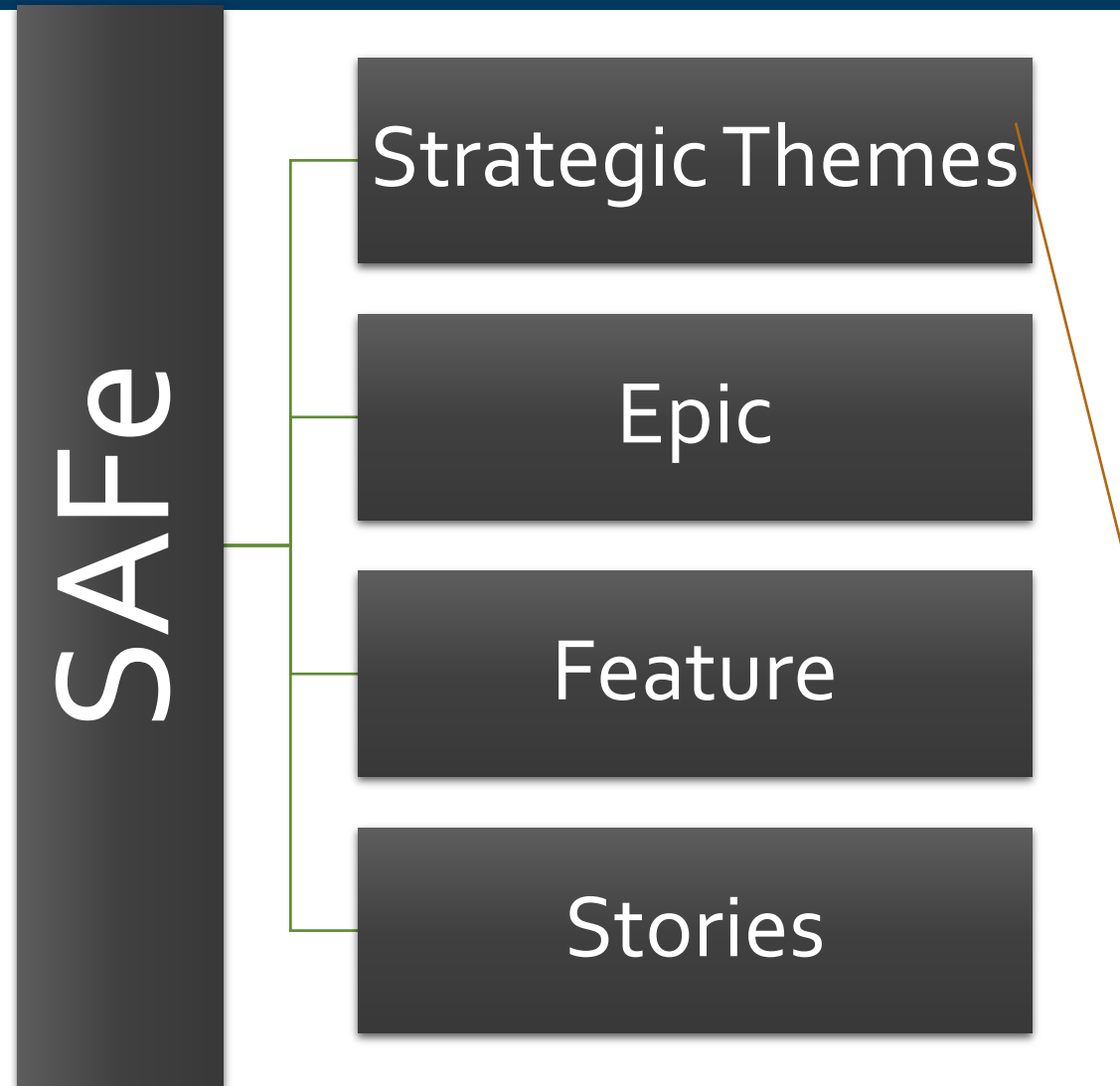




Fonte: Adaptworks / Agile Academy



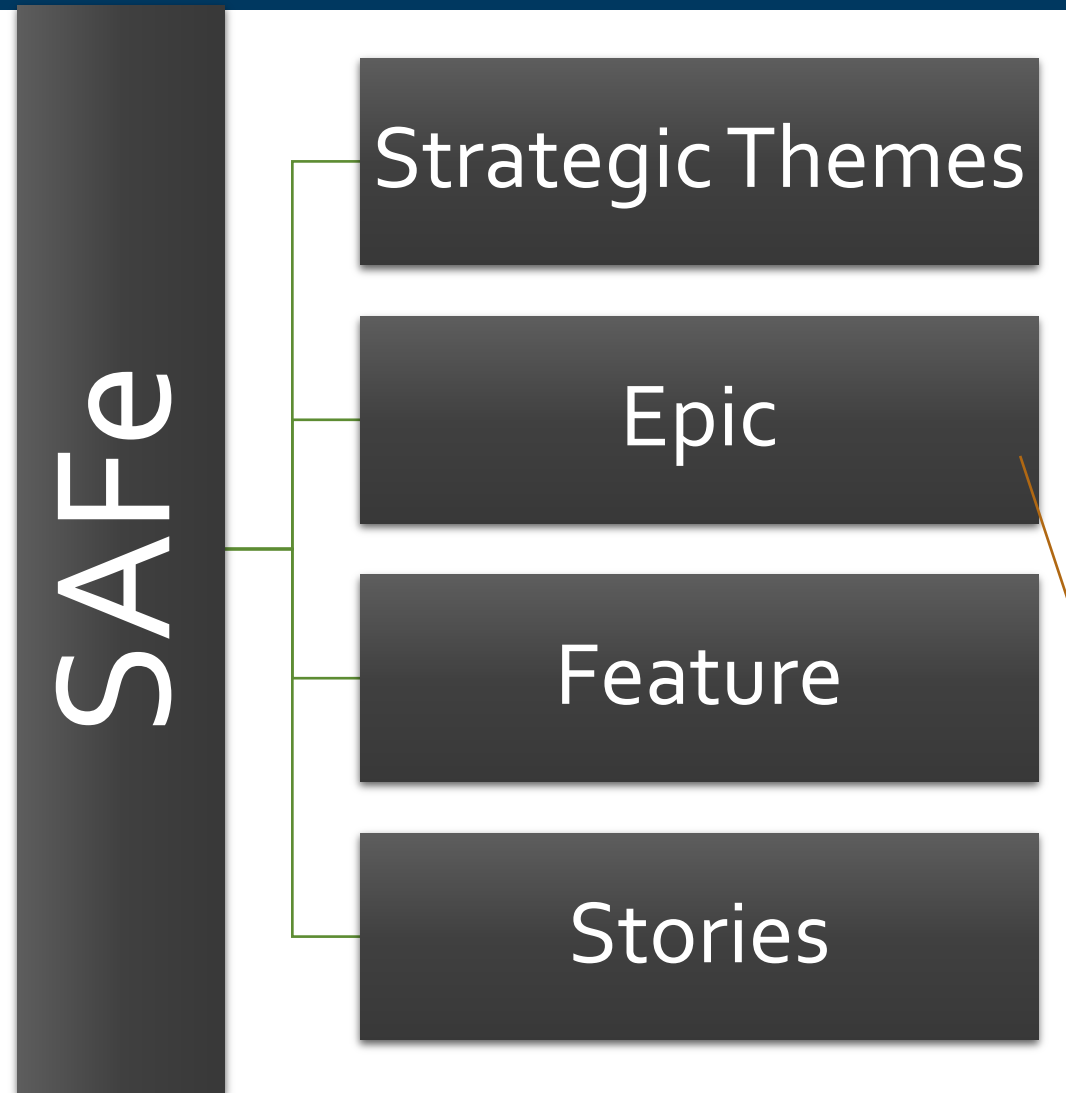
Fonte: Adaptworks / Agile Academy



- 1 – Redução de 150k dólares/ano automatizando tarefas repetitivas e de longa duração.
- 2 – Reduzir o tempo de lançamento da estratégia de marketing para novos produtos (tratores e colheitadeiras).
- 3 – Globalizar sistemas, reduzindo em x% o custo total de licenciamento e manutenção de software.

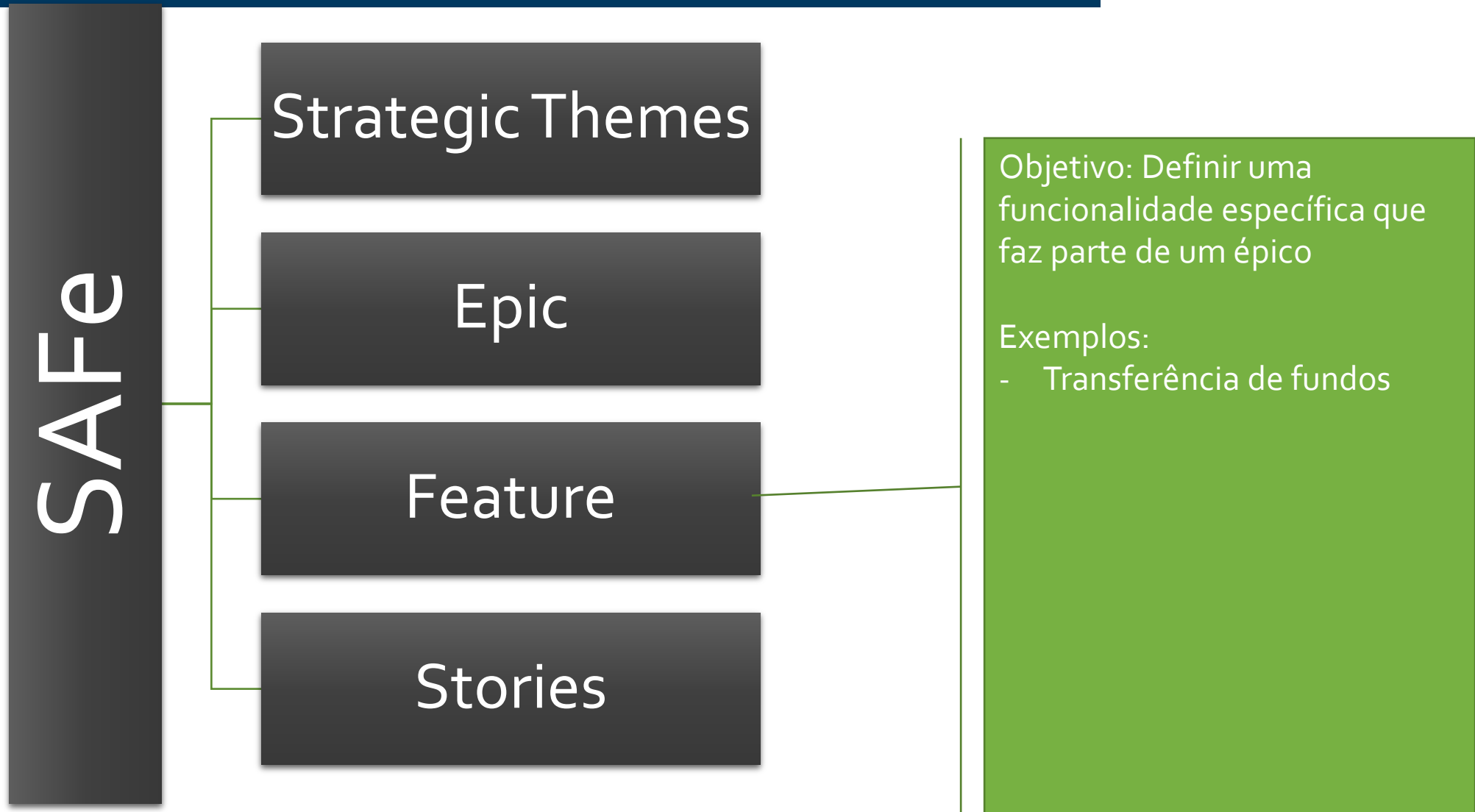


Fonte: Adaptworks / Agile Academy

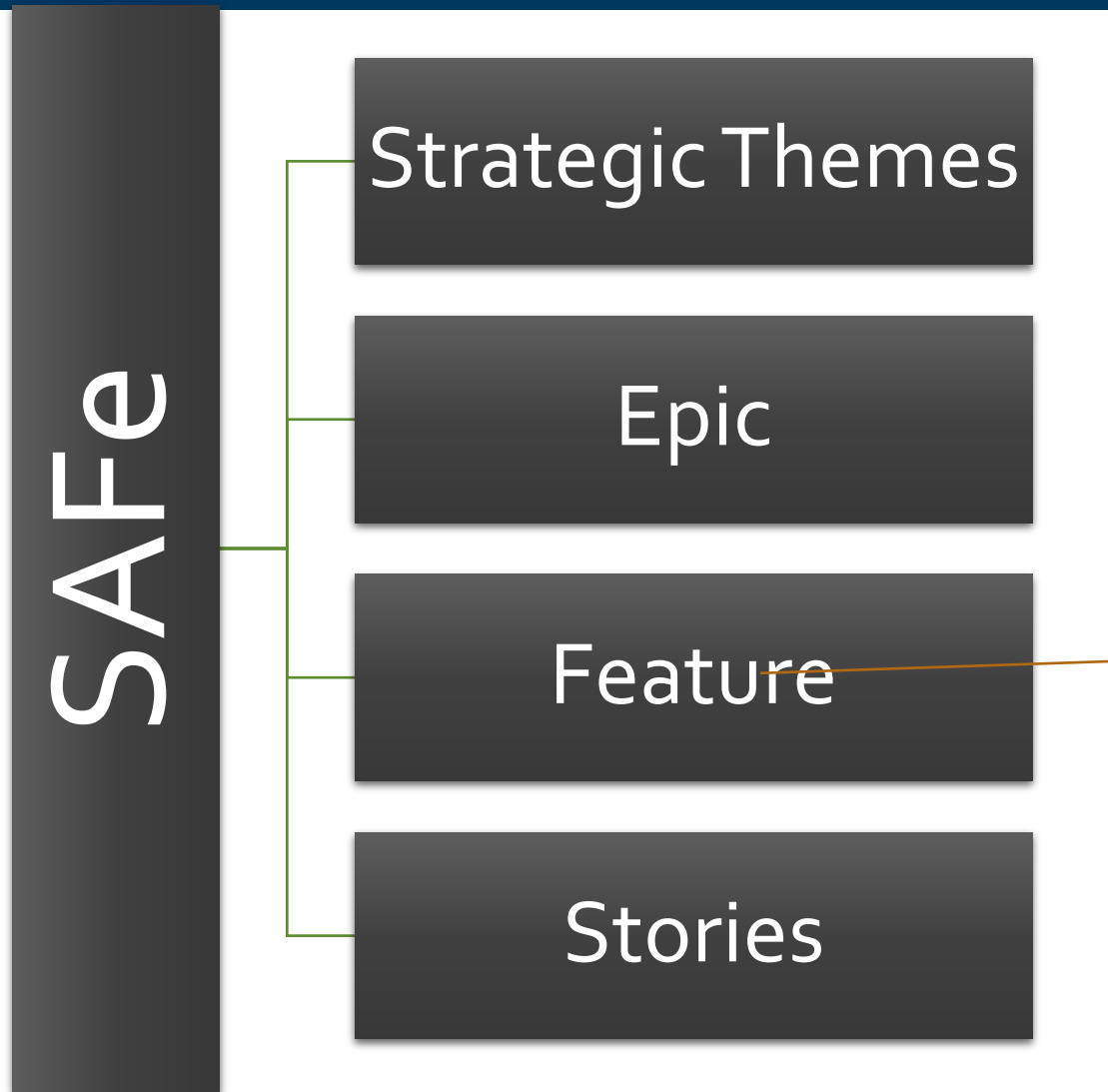


1.1

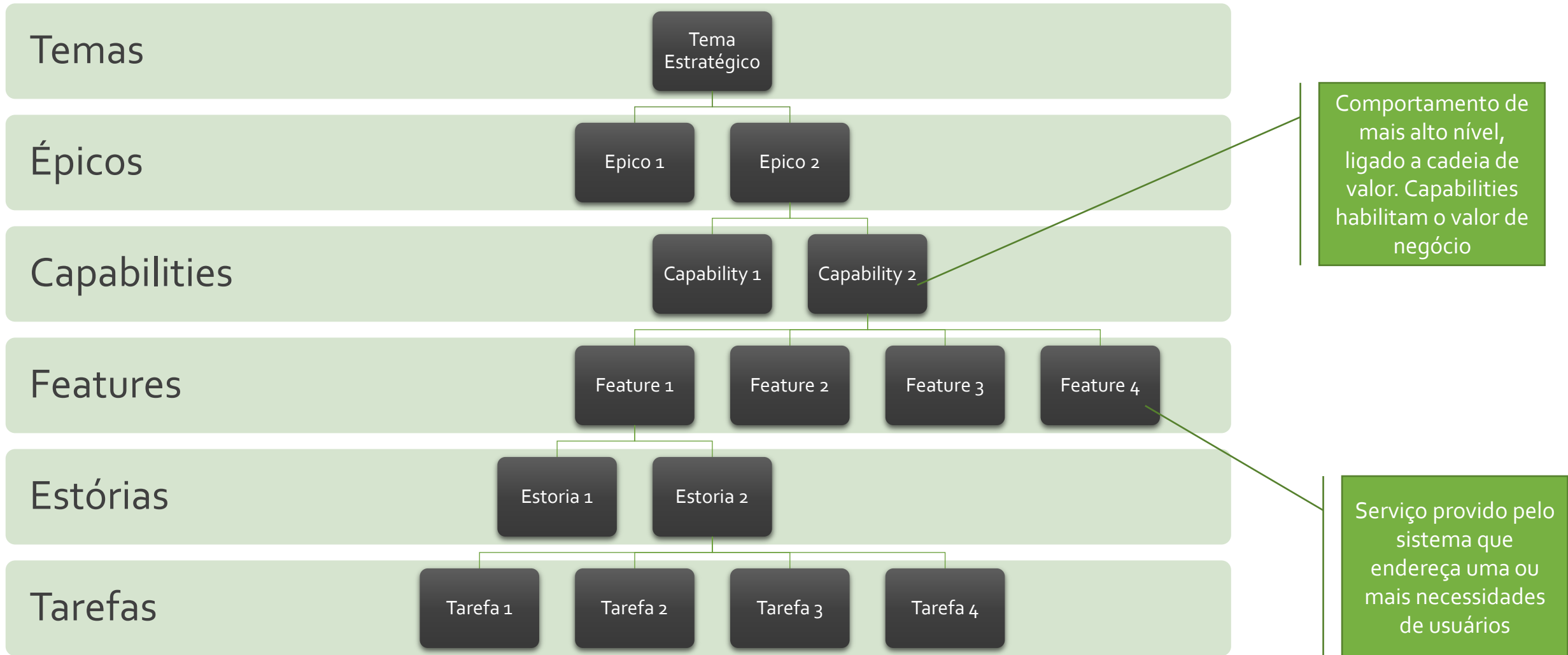
Para gestores que precisam decidir entre manufaturar um produto novo ou vender algum de estoque, a solução de Aging de Pátio é um sistema de tomada de decisão que vai agregar rapidamente dados de outros departamentos e mostrar um dashboard com os produtos no pátio e seu tempo de espera.

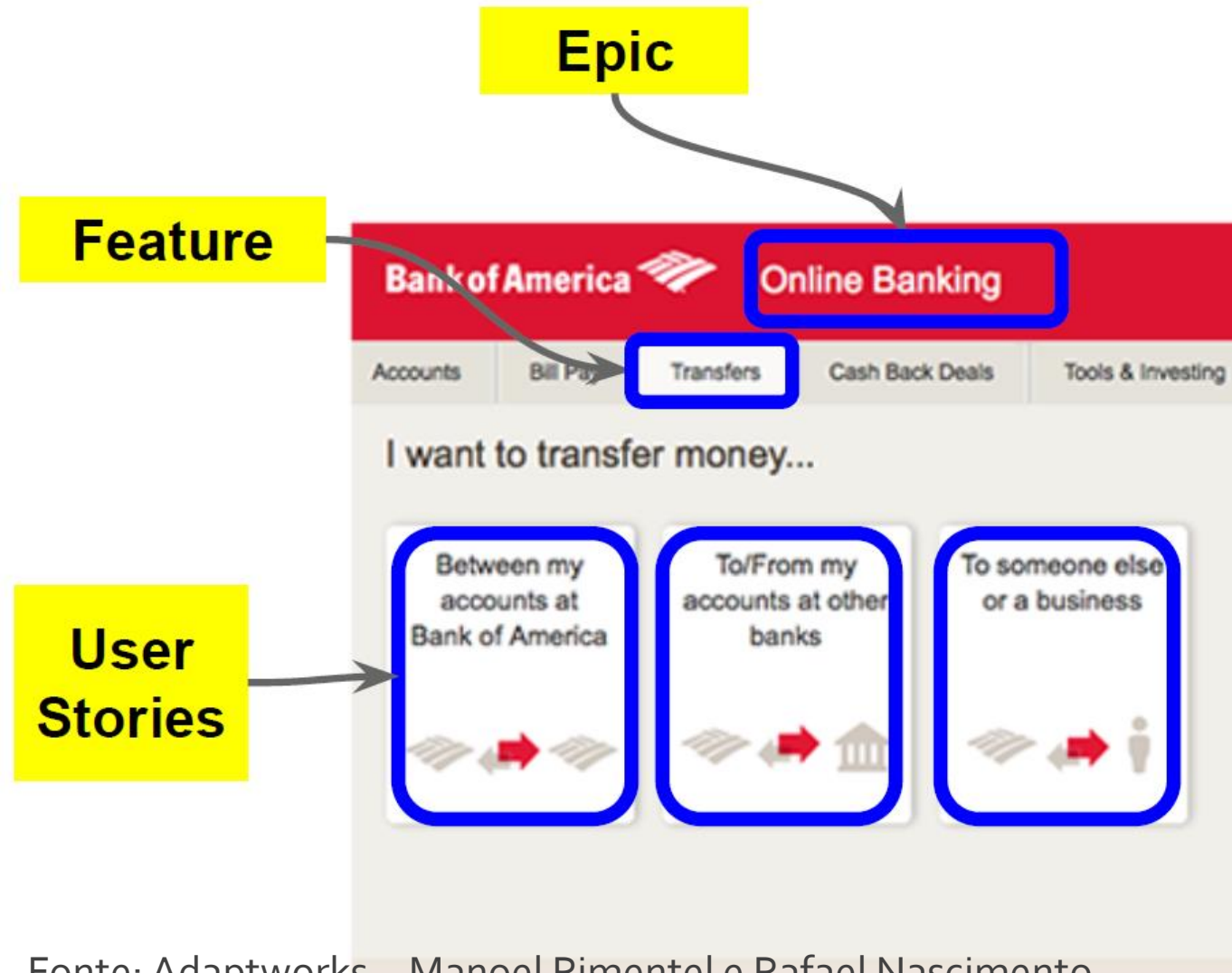


Fonte: Adaptworks / Agile Academy



- 1.1.1 – Funcionalidade de carga dos dados dos produtos da área de marketing
- 1.1.2 Funcionalidade de carga dos dados de produtos vindos da manufatura
- 1.1.3 Funcionalidade de exibição dos dados em um dashboard





Fonte: Adaptworks - Manoel Pimentel e Rafael Nascimento

Por atuar no segmento financeiro,
precisamos ofertar um sistema de
provisionamento de crédito para clientes em
sites de vendas online, para que os clientes
possam escolher os produtos e parcelá-los
com nossa instituição, pagando seus
parcelamentos e validando seu crédito
disponível através de um portal para o cliente
(ao estilo *home banking*)

Epics

- Possui um Epic Owner.
- Afetam diretamente a área de negócio de 3 formas:

Tempo

- Não se completa em um único sprint

Escopo

- Pode afetar múltiplas equipes (Release Trains/Feature Teams)

Organizacional

- Múltiplos segmentos de uma organização são afetados

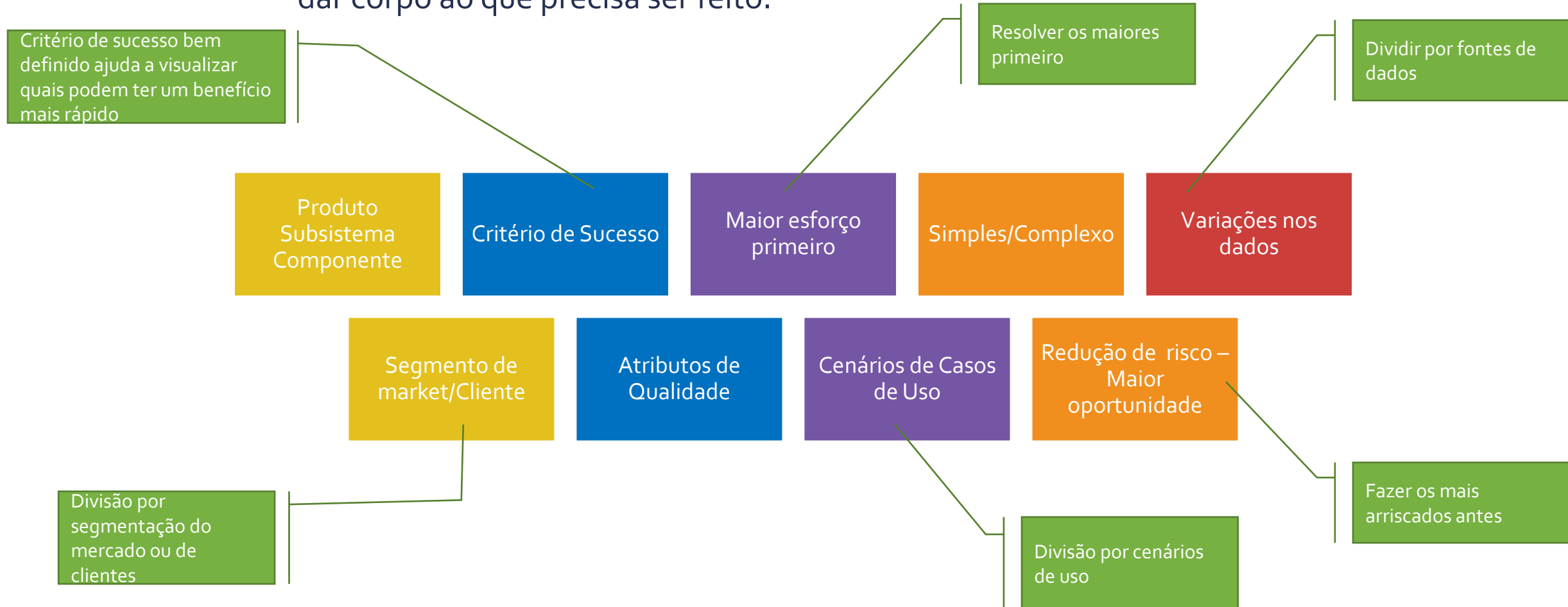
Epics

Por atuar no segmento financeiro, precisamos ofertar um sistema de provisionamento de crédito para clientes em sites de vendas online, para que os clientes possam escolher os produtos e parcelá-los com nossa instituição.

Por atuar no segmento financeiro, precisamos disponibilizar aos nossos clientes um portal para pagamento de suas parcelas, validação de saldo disponível, atualização cadastral

Epics – Como dividir

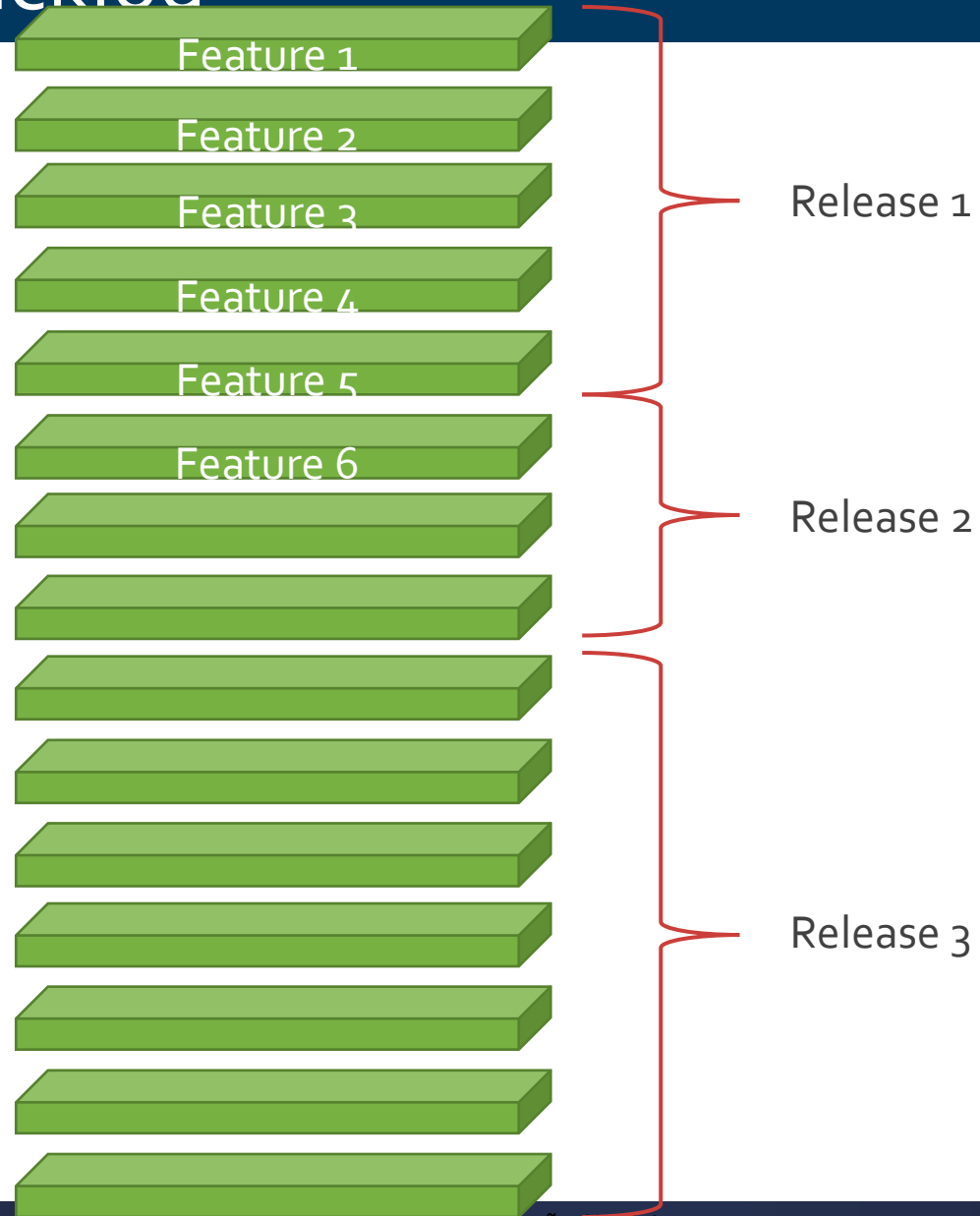
- É preciso dividir um Epic em funcionalidades (Features) para começar a dar corpo ao que precisa ser feito.



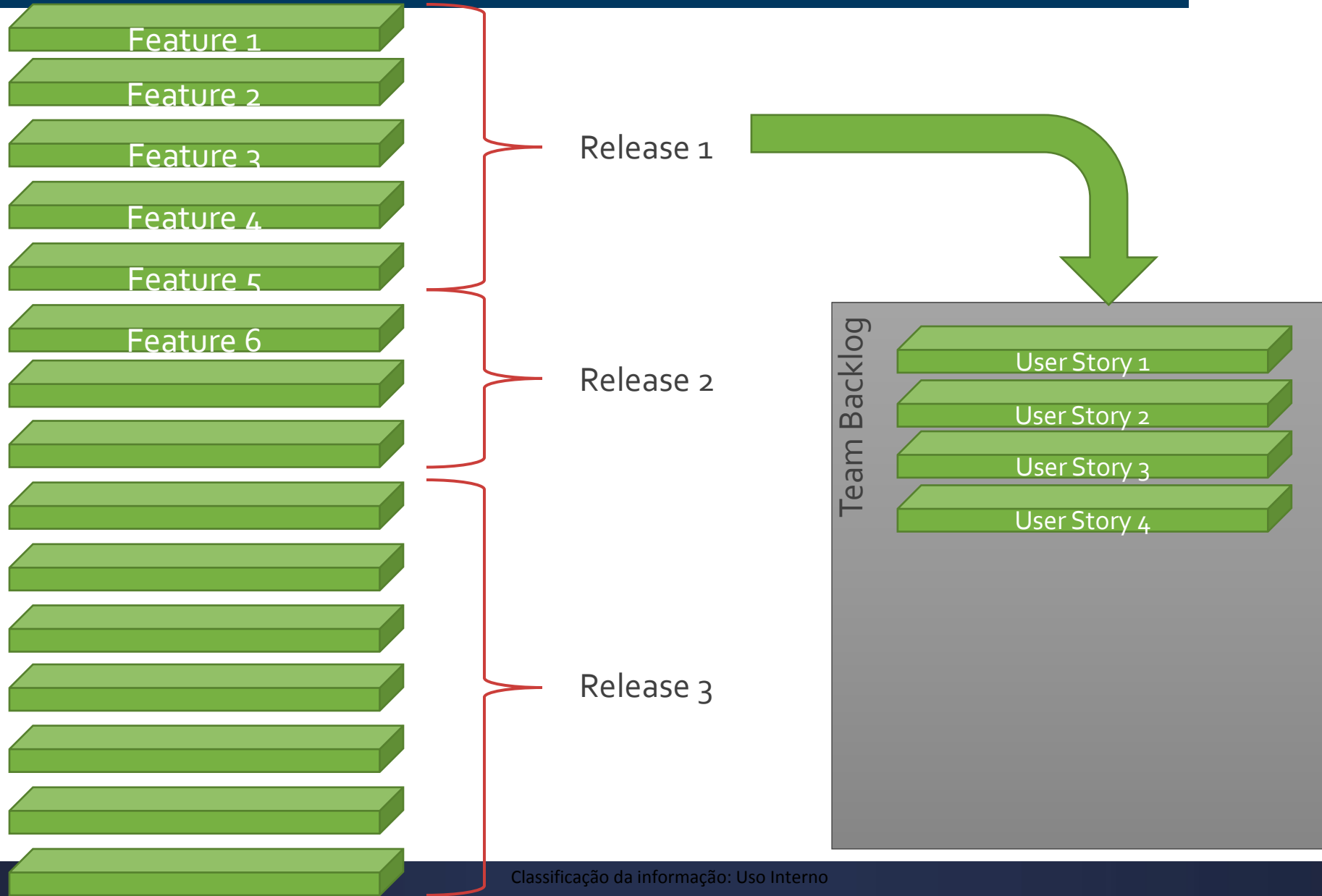
Program/Product Backlog

Maior prioridade

Menor prioridade

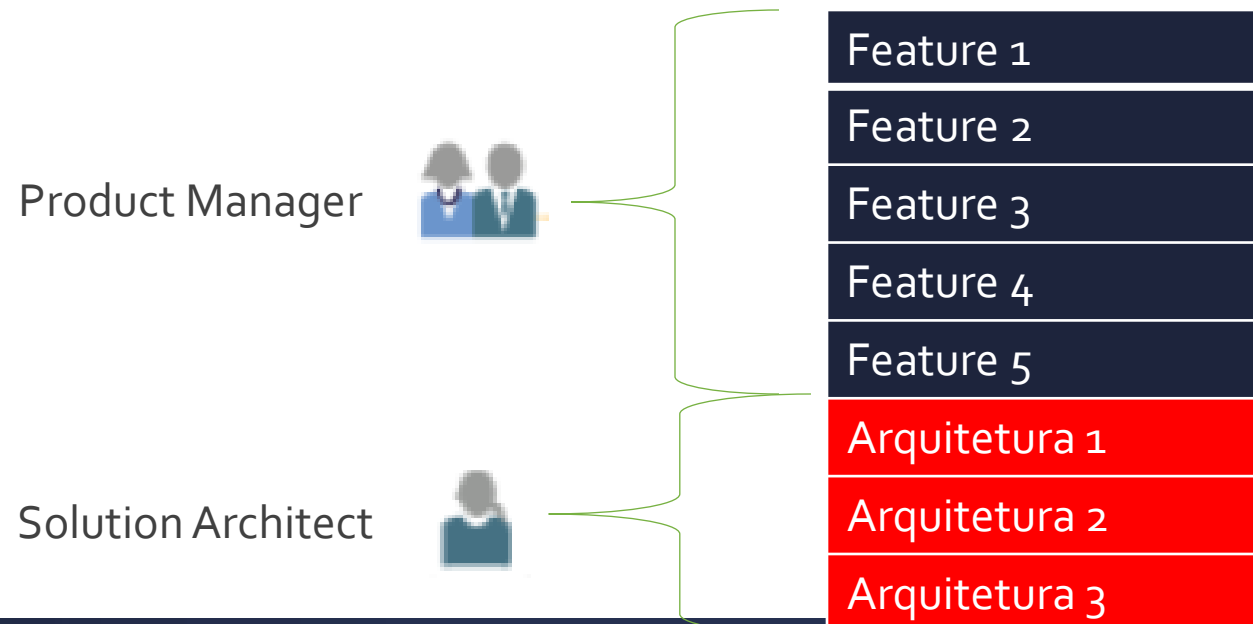


Program/Product Backlog



Program Backlog

- Mantido pelo Product Manager
- A partir de um Epic, múltiplos itens de backlog serão criados e são chamados de **Features**.
- Mescla itens de arquitetura e requisitos funcionais



Features

- Uma funcionalidade que pode ser composta por várias user stories.
- Não descrita na forma de user story.
- Um release contém várias features.
- Pode se confundir com Themes.
- Possuem critério de aceite.
- Não é um termo novo – Feature Driven Development?

Features - Matriz Feature/Benefit

Feature	Benefício
Provisão de Crédito	Provimento de crédito para os clientes online, observando riscos de inadimplência e posicionamento de um cliente em uma faixa de crédito.
Consulta Saldo	Cliente possui habilidade de consultar saldo disponível para financiar novos produtos.
Transferência	Cliente possui habilidade de transferir valores de sua conta corrente para contas correntes cadastradas; Cliente pode cadastrar pessoas para transferir valores.

Features

- O momento adequado para discussão de Features é no Release Planning.
- O máximo benefício no menor prazo (**WSJF**) ganha prioridade em SAFe.
- Sugestão SAFe é apresentar as Top Ten features na reunião de release planning.

Features

- O momento adequado para discussão de Features é no Release Planning.
- O máximo benefício no menor prazo (**WSJF**) ganha prioridade em SAFe.
- Sugestão SAFe é apresentar as Top Ten features na reunião de release planning.

$$WSJF = \frac{Cost\ of\ Delay}{Duration}$$

$$WSJF = \frac{Cost\ of\ Delay\ (Valor\ de\ Negócio + Criticida\ de\ tempo + (Redução\ de\ risco\ futuro\ | oportunidade\ de\ negócio\ futuro))}{Duration}$$

- Usuários preferem essa funcionalidade?
- Qual o impacto financeiro ao negócio?
- Existem penalidades envolvidas?

- Existe um prazo fixo?
- O negócio pode decair com o tempo se a funcionalidade não for entregue logo?
- Usuários terão de usar outra solução enquanto esperam?

Tamanho do item

- Essa funcionalidade previne riscos futuros ou é algo mais imediatista?

- Essa funcionalidade habilita novas oportunidades de negócio?

Features

- O momento adequado para discussão de Features é no Release Planning.
- O máximo benefício no menor prazo (**WSJF**) ganha prioridade em SAFe.
- Sugestão SAFe é apresentar as Top Ten features na reunião de release planning.

$$WSJF = \frac{\text{Cost of Delay}}{\text{Duration}}$$

$$WSJF = \frac{\text{Cost of Delay (Valor de Negócio + Crítica de tempo + Redução de risco futuro | oportunidade de negócio futuro)}}{\text{Duration}}$$

- Dica:

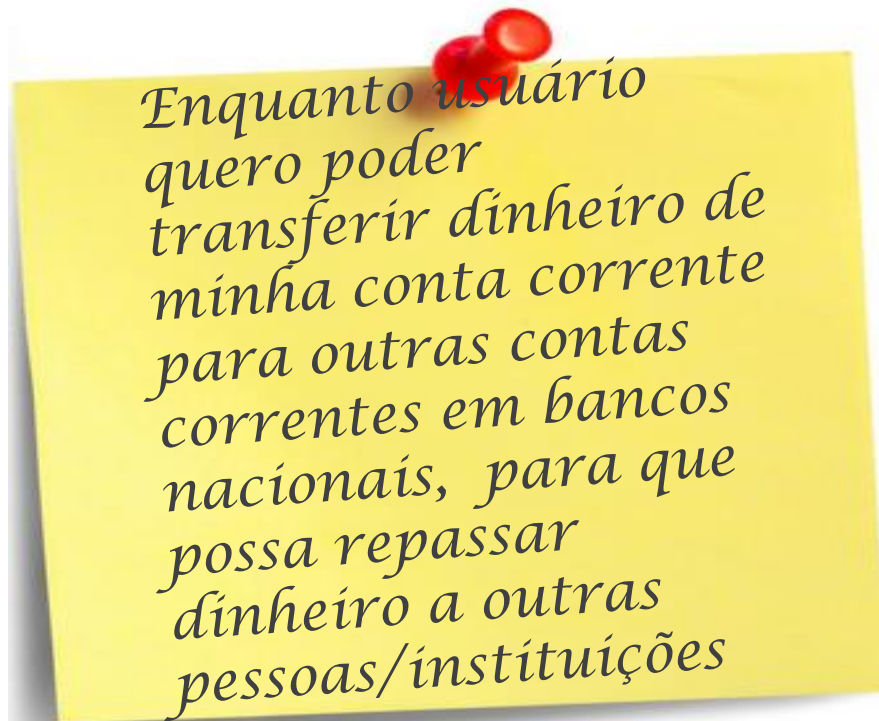
Olhar uma coluna por vez
Colocar 1 para o menor item

Feature	Valor de neg.	Critic. Tempo	Risco fut/oport. Neg.	Duração/ Job size	WSJF
A	20	3	13	13	???
B	1	3	5	3	???
C	5	5	1	5	???

Backlog e Stories

User Stories e Team Backlog

- Menor ítem disponível em um backlog
- Devem ser o mais atômicas possível

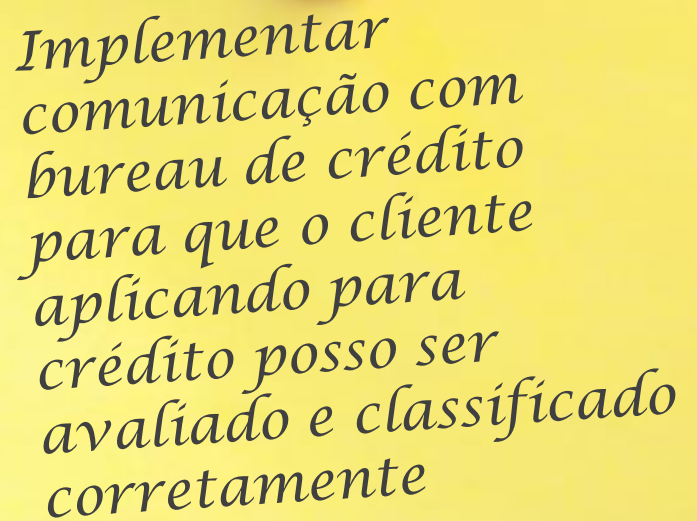


Critério de aceitação (Acceptance Criteria)

- Conta de origem com saldo suficiente
- Conta destino com valor creditado
- Conta origem com valor debitado

User Stories e Team Backlog

Podem existir histórias técnicas também.



*Implementar
comunicação com
bureau de crédito
para que o cliente
aplicando para
crédito posso ser
avaliado e classificado
corretamente*

Critério de aceitação (Acceptance Criteria)

- Dados do cliente disponíveis.
- Transmissão via HTTPS com certificado digital
- Comunicação com o bureau via web service.
- Resposta do bureau armazenada no banco XYZ.

User Stories e Team Backlog

Definição de READY

- Definido pelo time
- O critério usado pelo time para definir quando uma user story está pronta para ser desenvolvida (consumida do backlog)

Definição de DONE

- Definido pelo time
- Critério para definir quando uma user story está completa

User Stories e Team Backlog

I_{ndependent}

- Independência entre as user stories
- O time deve estar apto a pegar qualquer user story
- Ninguém deveria depender de outra pessoa

N_{egotiable}

- Uma história não é um contrato, é um convite para uma conversa.
- Satisfaça as necessidades do usuário, não da user story

V_{aluable}

- Parece óbvio, mas a user story precisa agregar valor ao usuário/cliente

E_{stimatable}

- Está difícil de estimar? Então é preciso investigar melhor a história

S_{mall}

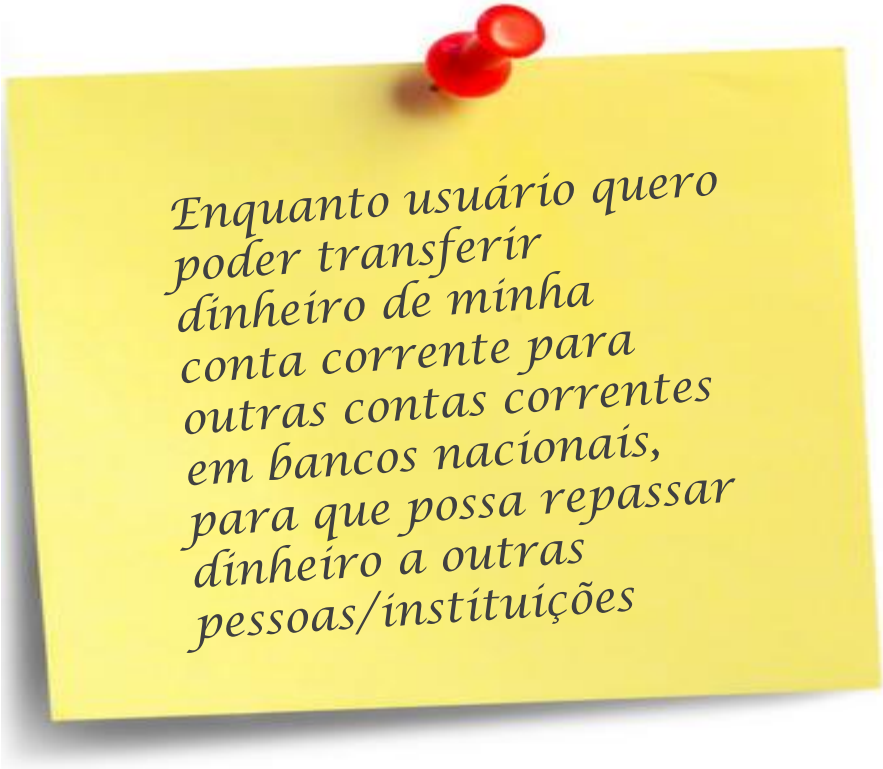
- User stories tem de ser pequenas. Talvez uma semana total.

T_{estable}

- Critérios de aceitação claros; fácil de testar.

Tasks - Tarefas

- O menor item que um time trabalha



*Enquanto usuário quero
poder transferir
dinheiro de minha
conta corrente para
outras contas correntes
em bancos nacionais,
para que possa repassar
dinheiro a outras
pessoas/instituições*

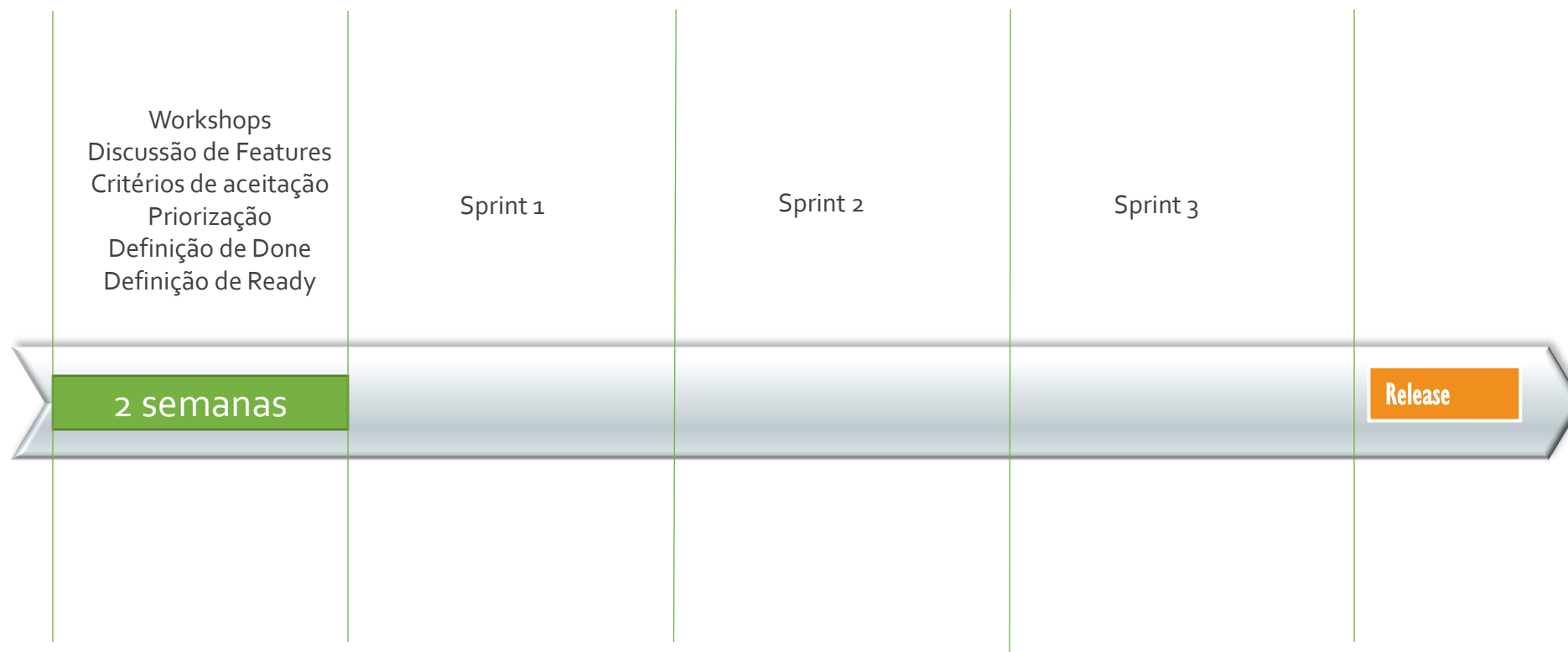
- Criar página transfer.aspx.
- Criar componente XYZ para realizar a transferência.
- Adicionar chamada da página no menu principal.
- Criar tabela HISTORICO para armazenar transferência efetuada.
- Criar rotina de arquivamento dos historicos com mais de 2 anos.

Pare de Começar – Comece a Terminar

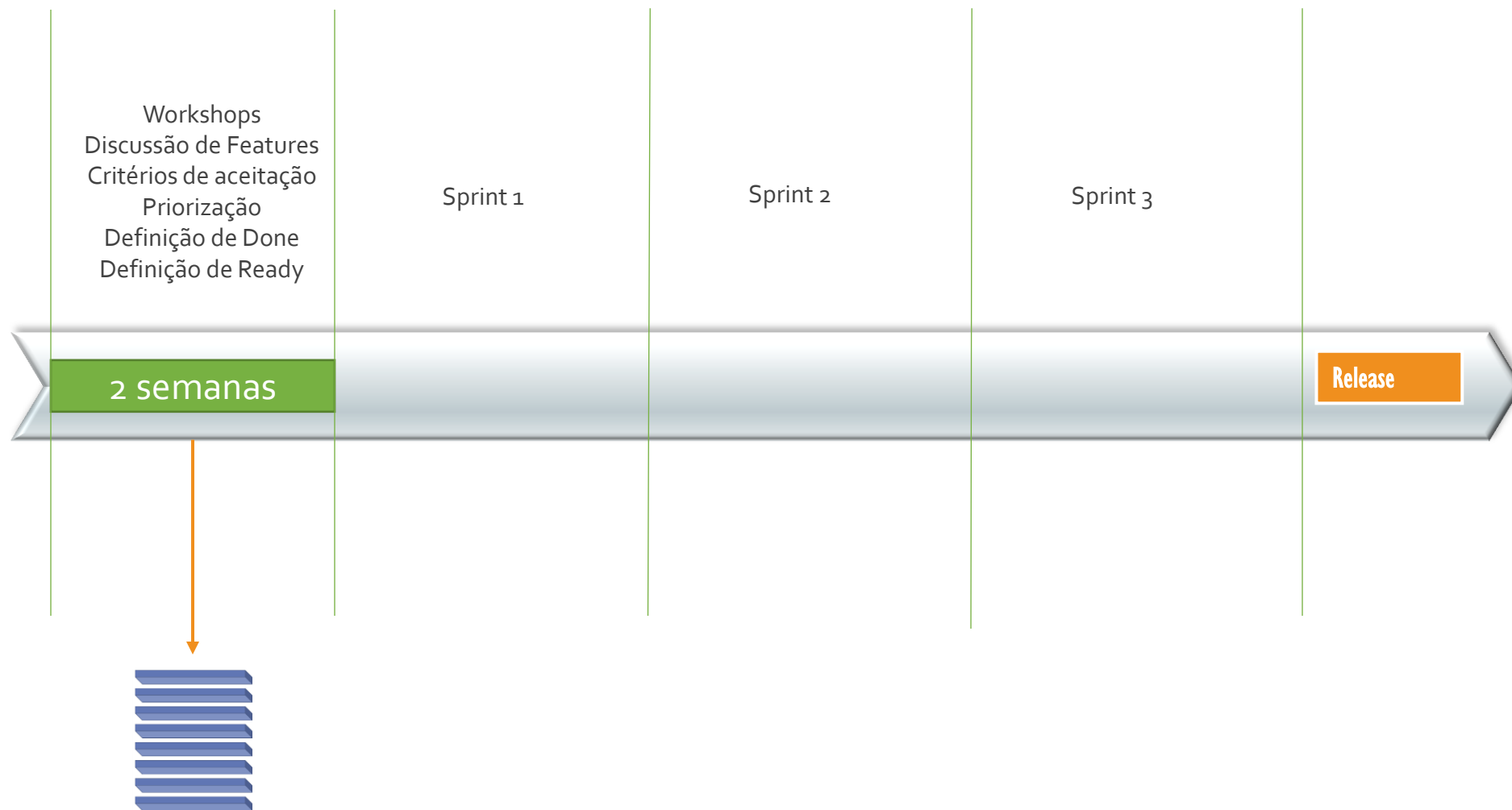
Release Planning



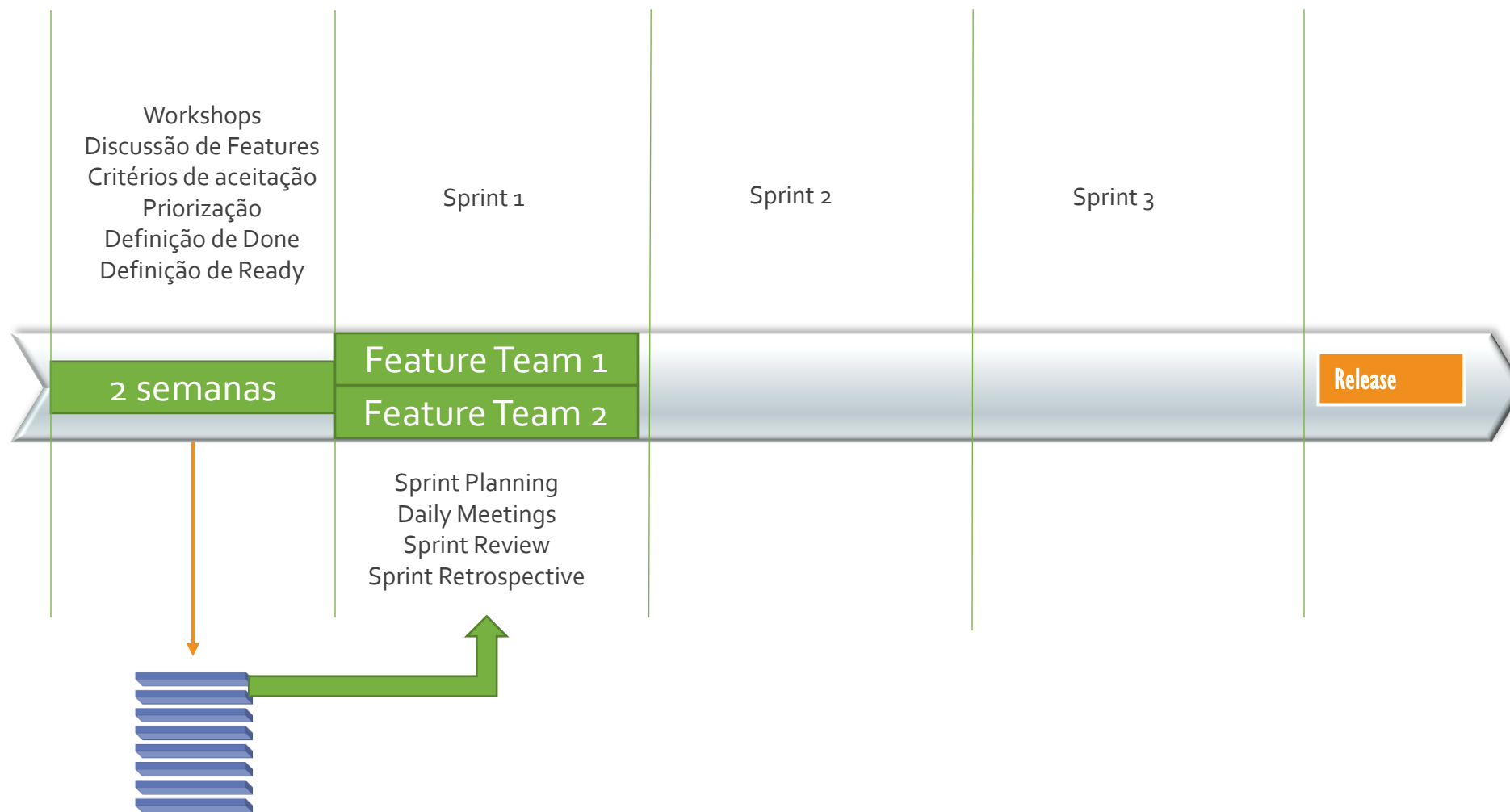
Release Planning / Program Increment



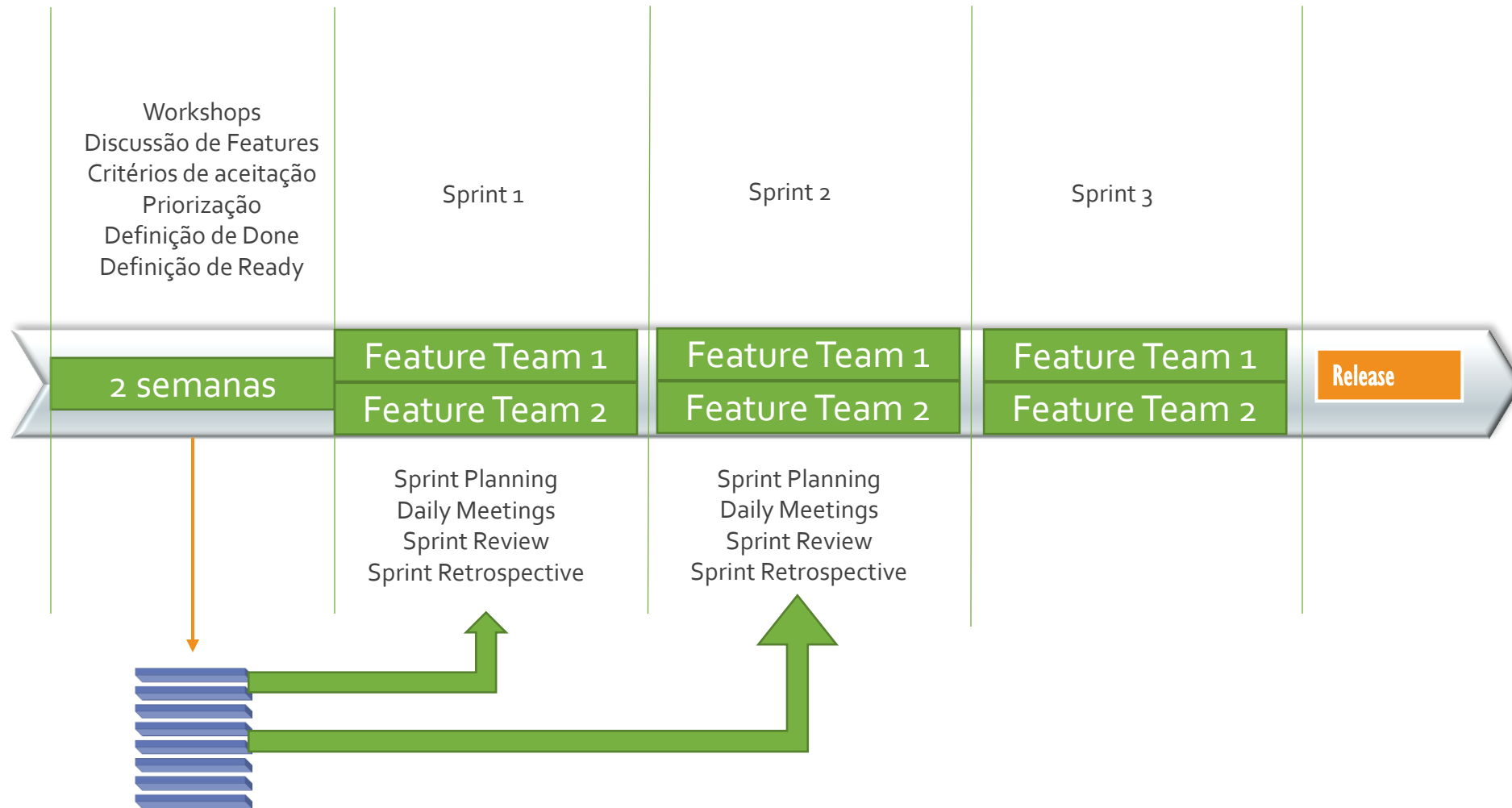
Release Planning / Program Increment



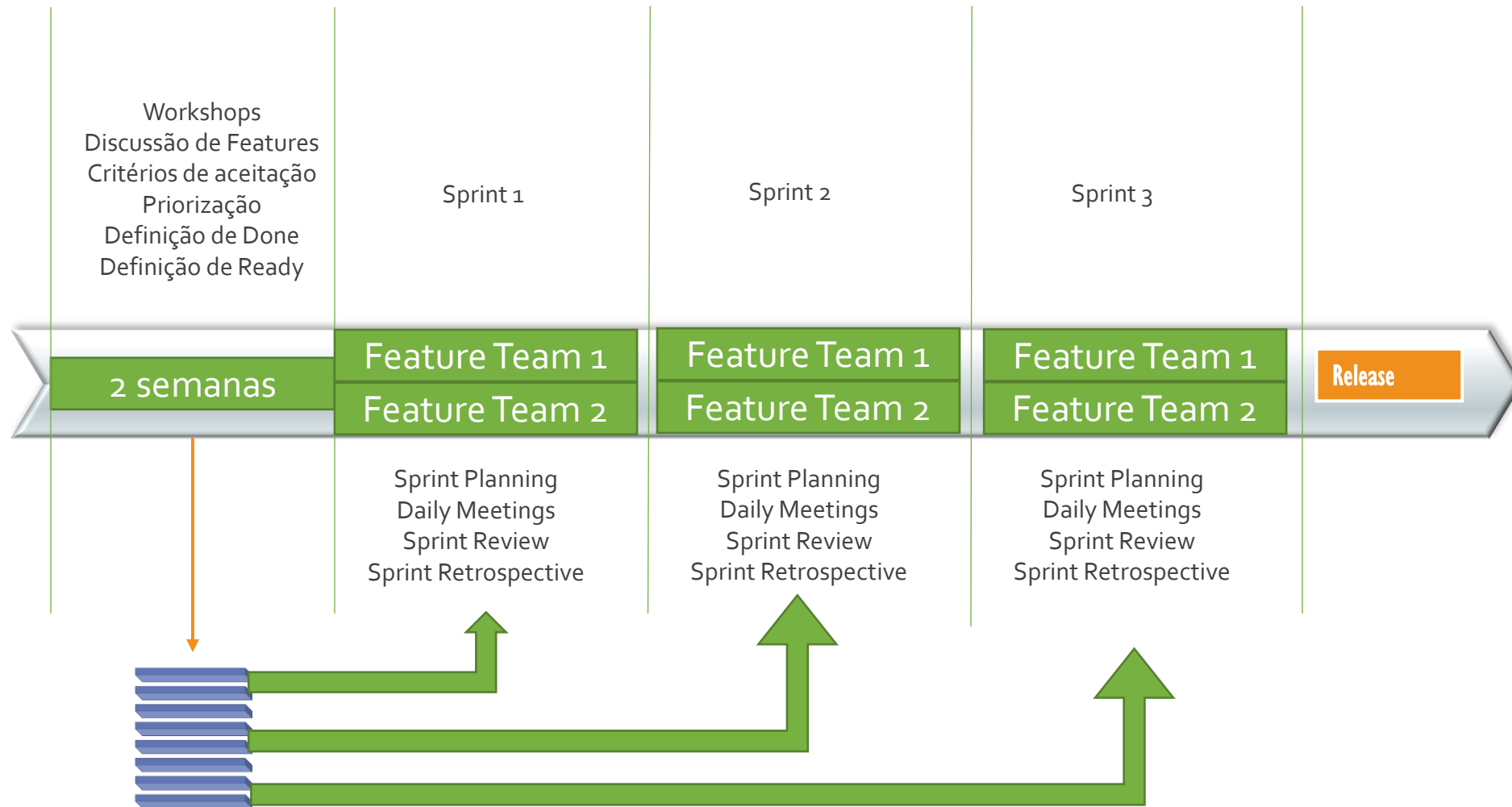
Release Planning / Program Increment



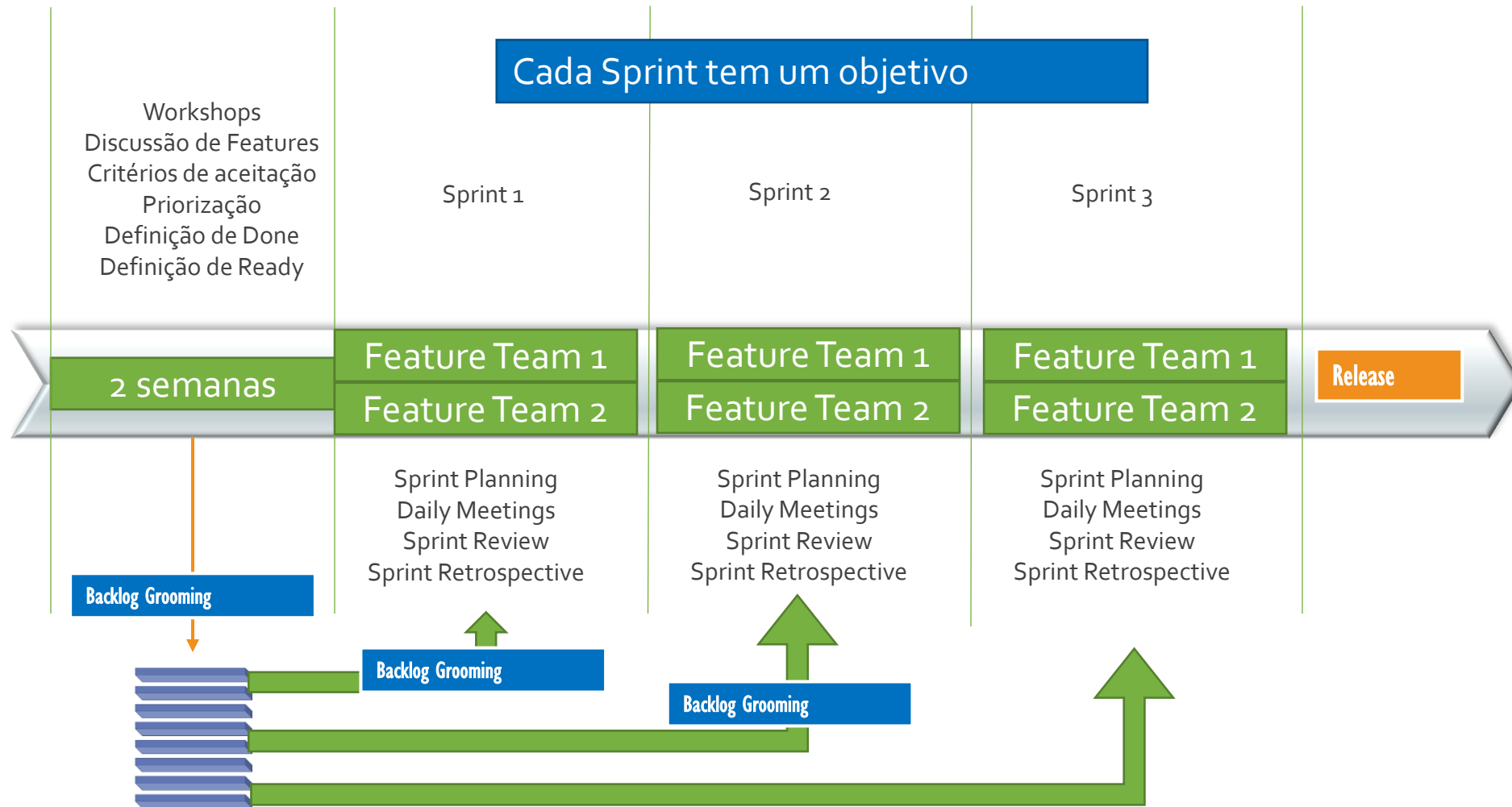
Release Planning / Program Increment



Release Planning / Program Increment



Release Planning / Program Increment



Release Planning / Program Increment

- O que entra em cada release?



Release Planning – Agile Release Train (ART)



- Principal fonte de entregas dentro de SAFe
- Sugestão de no máximo 10 semanas
- Pode entregar a qualquer momento

Abordagem

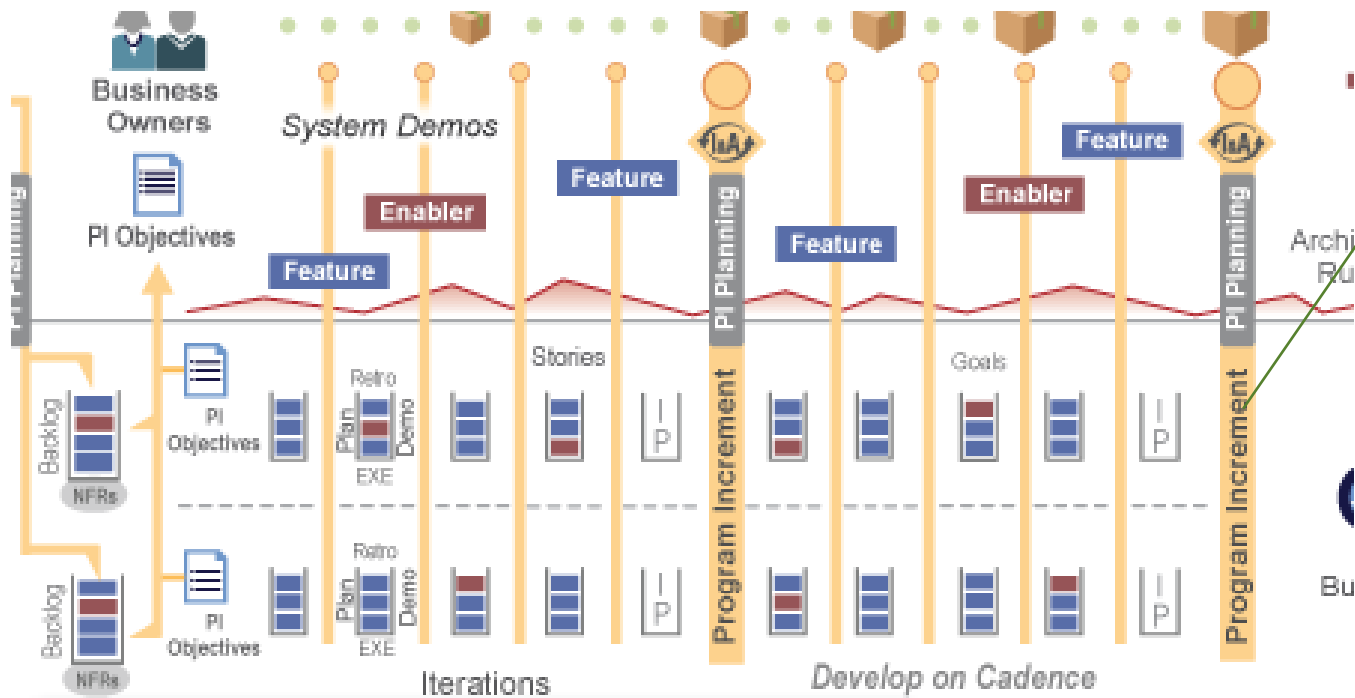
- Qualidade é fixa
- Data é fixa
- Escopo é variável

Lema
Always on Track –
Always on Time

Time

- ART contém de 5 a 12 agile teams
- Sugestão de no máximo duas áreas geográficas
- 3 horas de diferença no máximo
- 4 dias na semana no mínimo

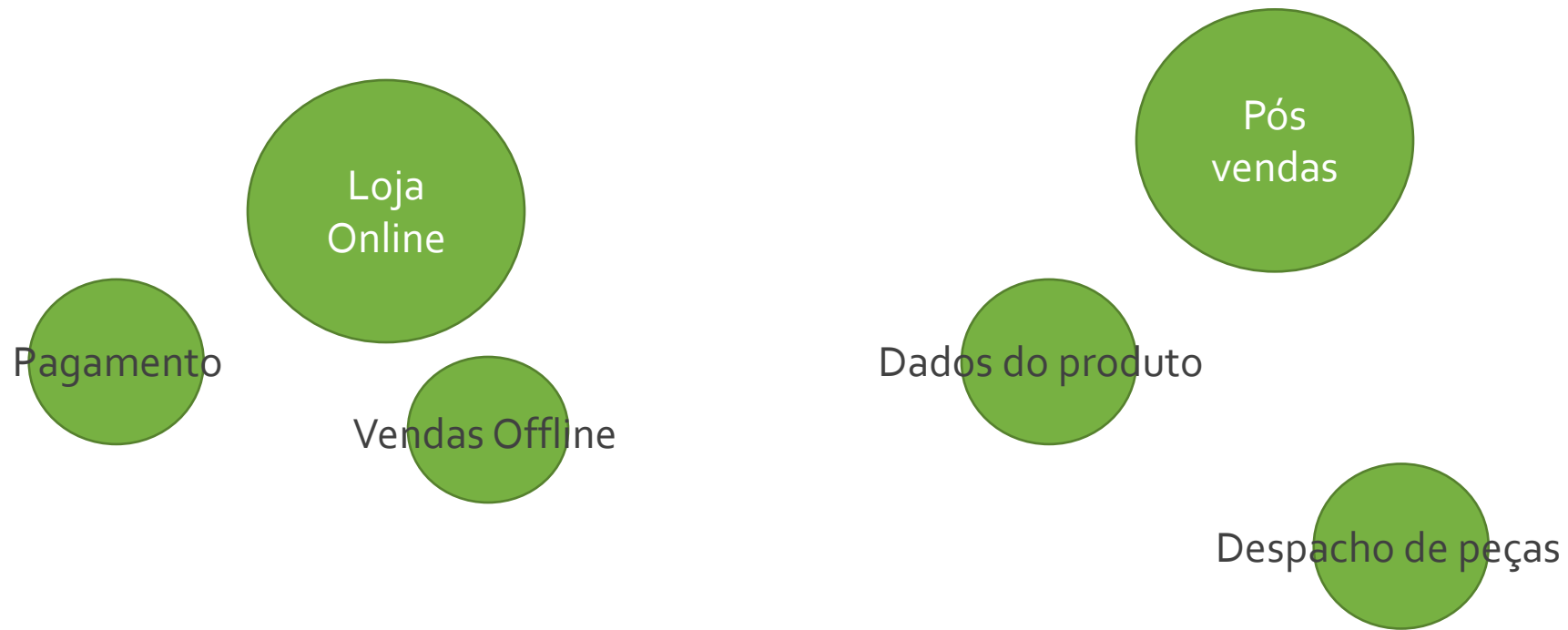
Program Increment



ciclos de PDCA

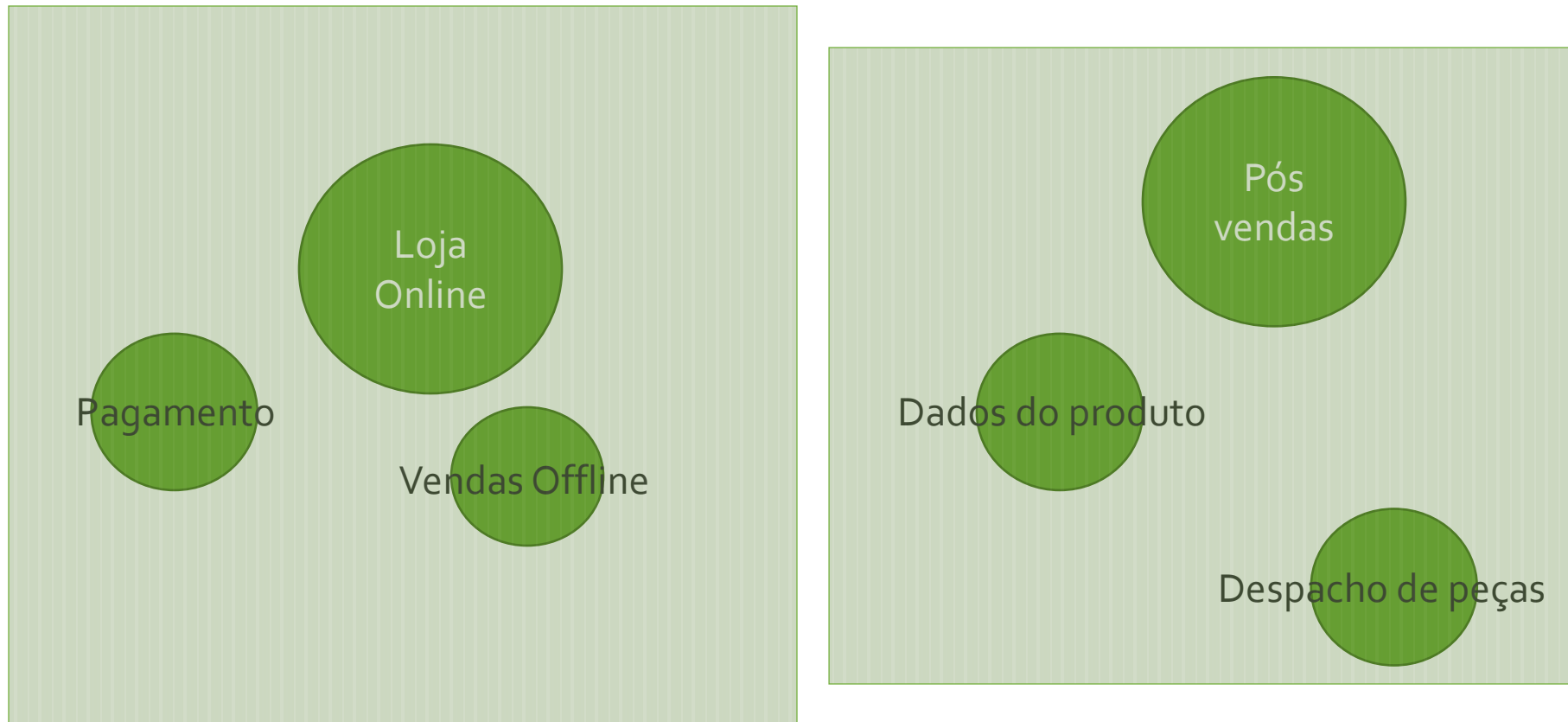
O que são Feature Teams

A feature team is “is a **long-lived, cross-functional, cross-component** team that completes many end-to-end **customer features—one by one.**”



O que são Feature Teams

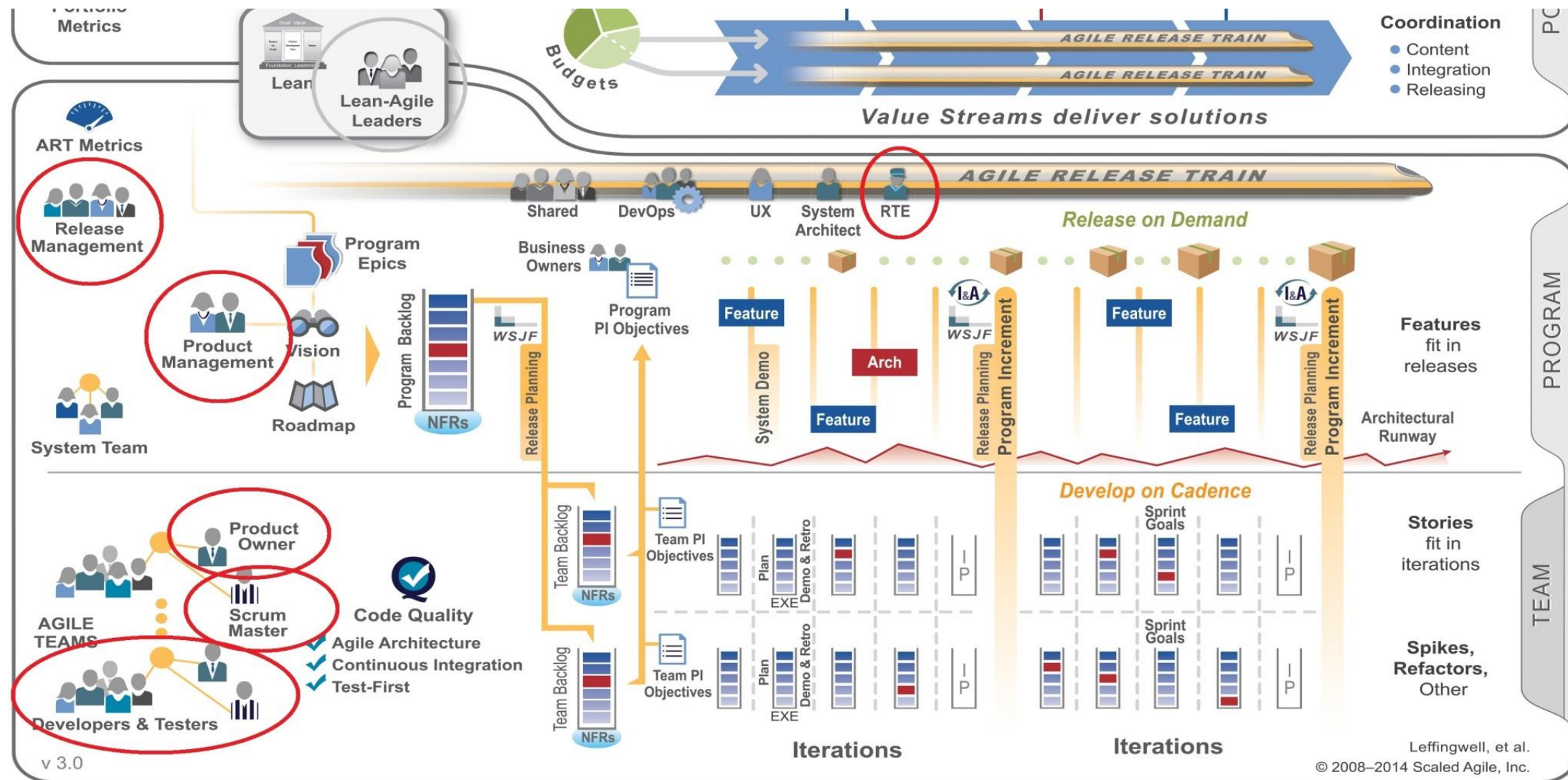
A feature team is “is a **long-lived, cross-functional, cross-component** team that completes many end-to-end **customer features—one by one.**”



Perguntas

- Quando se discute arquitetura no modelo de sprints? Porque não Big Design Up Front?
 - Tarefas podem entrar nos sprints
 - Um estudo revela que quanto menos dedico a discussão de arquitetura, mais retrabalho tenho (Livro Software Architecture in Practice)
 - Entretanto, quanto mais dedico à discussão up-front, menos efetivo meu código é: Mais linhas de código, menos benefício para o usuário.
 - A resposta é: Faça arquitetura suficiente
 - Pode-se fazer um Sprint Zero
- Como vocês dividiriam os releases para a construção de uma casa?

Uma proposta lightweight SAFe



Referências

- [Adaptworks.com.br](https://adaptworks.com.br)
- Agile Academy
- Livro Software Architecture in Practice