

TRANS  
FORMAR  
O PAÍS PELA  
EDUCAÇÃO  
É O QUE  
NOS MOVE

ecossistema  
ânima





# Modelos, Métodos e Técnicas da Engenharia de Software



## Aula 05 ESTUDO DE CASO ÁGIL.

Prof. Luis Ybarra



# O Manifesto Ágil

Durante anos, a Engenharia de Software inspirou-se em processos de manufatura para a consolidação de seus métodos de trabalho.

Nascida na segunda metade do século XX, buscou em setores emergentes da indústria da época grande parte das teorias e dos métodos de produção.

Em especial, o campo automobilístico, em ampla ascensão industrial, teve importante papel para a constituição da nova indústria de TI.

Graças ao modelo de produção em série de Henry Ford, altamente inspirado por Frederick Taylor, todo o pensamento tradicional da ciência do desenvolvimento de software desenrolouse com intenso foco na padronização de componentes e processos e na mecanização do movimento.

Já em meados dos anos 90, começaram a surgir processos alternativos de desenvolvimento de software, em resposta àqueles tradicionais, considerados excessivamente regrados, lentos, burocráticos e inadequados à natureza da atividade.

Esses novos processos foram apelidados de “leves” (lightweight), em oposição aos anteriores, “pesados” (heavyweight).



Essas metodologias só passaram a ser chamadas de ágeis após 2001, quando um grupo de 17 especialistas se reuniu na estação de ski Snowbird, em Utah, nos Estados Unidos, para discutir maneiras de desenvolver software de uma forma mais leve, rápida e centrada em pessoas.

Eles cunharam os termos “Desenvolvimento Ágil de Software” e “Métodos Ágeis” e criaram o Manifesto Ágil, amplamente difundido como a definição canônica do desenvolvimento ágil, composto pelos valores e princípios.

# 👉 MANIFESTO PARA OS DESENVOLVEDORES ÁGEIS DE SOFTWARE

O Manifesto para o Desenvolvimento Ágil de Softwares, comumente conhecido como Manifesto Ágil, é uma expressão intencionalmente simplificada dos principais valores do gerenciamento ágil de projetos. Use-o como um guia para implementar as metodologias ágeis em seus projetos.

“Estamos revelando as melhores maneiras de desenvolver softwares, ajudando outras pessoas no processo.

Com esse trabalho, passamos a valorizar:

- Pessoas e interações, acima de processos e ferramentas.
- Software validado, acima da documentação completa.

- Colaboração do cliente, acima da negociação de contratos.
- Resposta à mudança, acima de seguir um plano.

Ou seja, embora haja valor nos itens à direita, valorizamos mais os itens à esquerda.”

©Agile Manifesto Copyright 2001: Kent Beck, Mike Beedle, Arie van Bennekum, Alistair Cockburn, Ward Cunningham, Martin Fowler, James Grenning, Jim Highsmith, Andrew Hunt, Ron Jeffries, Jon Kern, Brian Marick, Robert C. Martin, Steve Mellor, Ken Schwaber, Jeff Sutherland, Dave Thomas.

Essa declaração pode ser copiada sem restrições sob qualquer forma, mas apenas por completo, por meio desse aviso legal.



- 1 - Nossa maior prioridade é satisfazer o cliente através da entrega contínua e adiantada de software com valor agregado.
- 2 - Aceitar mudanças de requisitos, mesmo no fim do desenvolvimento. Processos ágeis se adequam a mudanças, para que o cliente possa tirar vantagens competitivas.
- 3 - Entregar frequentemente software funcionando, de poucas semanas a poucos meses, com preferência à menor escala de tempo.
- 4 - Pessoas de negócio e desenvolvedores devem trabalhar diariamente em conjunto por todo o projeto.
- 5 - Construir projetos em torno de indivíduos motivados, dando a eles o ambiente e o suporte necessários e confiando neles para fazer o trabalho.
- 6 - O método mais eficiente e eficaz de transmitir informações para, e entre, uma equipe de desenvolvimento é por meio de conversa face a face.
- 7 - Software funcionando é a medida primária de progresso.
- 8 - Os processos ágeis promovem desenvolvimento sustentável. Os patrocinadores, desenvolvedores e usuários devem ser capazes de manter um ritmo constante indefinidamente.
- 9 - Contínua atenção a excelência técnica e bom design aumentam a agilidade.
- 10 - Simplicidade: A arte de maximizar a quantidade de trabalho não realizado é essencial.
- 11 - As melhores arquiteturas, requisitos e designs emergem de times auto-organizáveis.
- 12 - Em intervalos regulares, a equipe reflete sobre como se tornar mais eficaz e, então, refina e ajusta seu comportamento de acordo.



## ESTUDO DE CASO ÁGIL

O gerenciamento ágil de projetos foca a melhoria contínua, a flexibilidade do escopo, as informações da equipe e a entrega de produtos essenciais de qualidade.

Além disso, aborda a inclusão do scrum como uma estrutura, a programação extrema (XP) para incorporar qualidade no início e o pensamento lean para eliminar desperdícios.

Essas e muitas outras ferramentas e técnicas ajudam organizações, equipes e pessoas a adotar o Manifesto Ágil e os 12 Princípios, que se destinam a pessoas, comunicações, produto e flexibilidade.



The graphic features the word "AGILE" written vertically in large, red, hand-drawn capital letters on the left side. To the right of this, the title "Manifesto for Agile Software Dev." is written in red, underlined, cursive-style text. Below the title, there is a list of four principles, each preceded by a red circle with a diagonal slash. The principles are written in black, hand-drawn capital letters.

**AGILE**

Manifesto for Agile Software Dev.

- INDIVIDUALS AND INTERACTIONS OVER PROCESSES AND TOOLS
- WORKING SOFTWARE OVER COMPREHENSIVE DOCUMENTATION
- CUSTOMER COLLABORATION OVER CONTRACT NEGOTIATION
- RESPONDING TO CHANGE OVER FOLLOWING A PLAN

# ESTUDO DE CASO ÁGIL

## GUIA DE VALOR

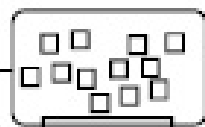
### Estágio 1: VISÃO DO PRODUTO

**Descrição:** Os objetivos do produto e seu alinhamento com a estratégia da empresa  
**Responsável:** Product owner  
**Frequência:** Pelo menos anualmente

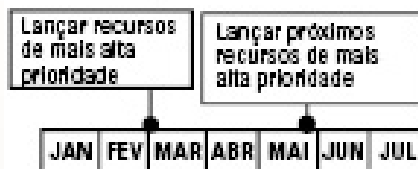


### Estágio 2: GUIA DO PRODUTO

**Descrição:** Visão geral dos recursos do produto que criam sua visão  
**Responsável:** Product owner  
**Frequência:** Pelo menos semestral



### Estágio 3: PLANO DE LANÇAMENTO



(Os estágios 1-3 são práticas comuns fora do scrum)

**Descrição:** Momento certo de lançar a funcionalidade específica do produto  
**Responsável:** Product owner  
**Frequência:** Pelo menos trimestral

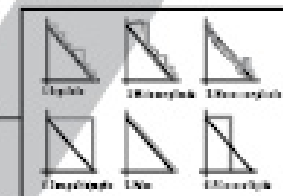
### Estágio 4: PLANEJAMENTO DO CICLO



**Descrição:** Estabelecimento de objetivos e tarefas específicos da iteração  
**Responsável:** Product owner e equipe de desenvolvimento  
**Frequência:** No início de cada ciclo

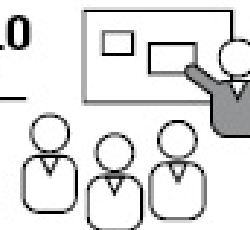
### Estágio 7: RETROSPECTIVA DO CICLO

**Descrição:** Aprimoramento do ambiente e processos pela equipe para otimizar a eficiência  
**Responsável:** Equipe scrum  
**Frequência:** No final de cada ciclo



### Estágio 6: REVISÃO DO CICLO

**Descrição:** Demonstração do produto validado  
**Responsável:** Product owner e equipe de desenvolvimento  
**Frequência:** No final de cada ciclo



### Estágio 5: SCRUM DIÁRIO

**Descrição:** Estabelecimento e coordenação das prioridades do dia  
**Responsável:** Equipe de desenvolvimento  
**Frequência:** Diária

24 horas

1 - 4 semanas

CICLO

Preparação

Execução





## 15.6 COACHING PARA TIMES ÁGEIS

De acordo com a Sociedade Brasileira de Coaching, um bom processo de Coaching é baseado em um Foco, que desperta Ações, que geram determinados Resultados, que são passíveis de serem Melhorados continuamente.

Observe que esse fluxo básico do processo de Coaching funciona como um espelho ao Coachee, para visualizar quais resultados ele está obtendo e, principalmente, aprender de forma empírica a melhorar seu comportamento para alcançar resultados ainda melhores.

Para atuar como base a esse ciclo de melhoria contínua de um Coachee, uma boa sessão de Coaching normalmente segue a seguinte estrutura de discussão:

- os aprendizados desde a última sessão;
- os avanços que o Coachee teve desde a última sessão;
- o que podemos desenvolver (ou melhorar) a partir da sessão atual;
- os aprendizados da sessão atual
- quais tarefas o Coachee identificou e se responsabilizou em desenvolver até a próxima sessão

Essa característica evidencia uma forte sinergia com a filosofia ágil.



## 15.7 RESUMINDO O COACHING EM ALGUNS PASSOS

O processo de Coaching inicia quando um indivíduo (que se tornará um Coachee) possui um sonho (que se tornará uma meta).(Prikladnicki, 2014).

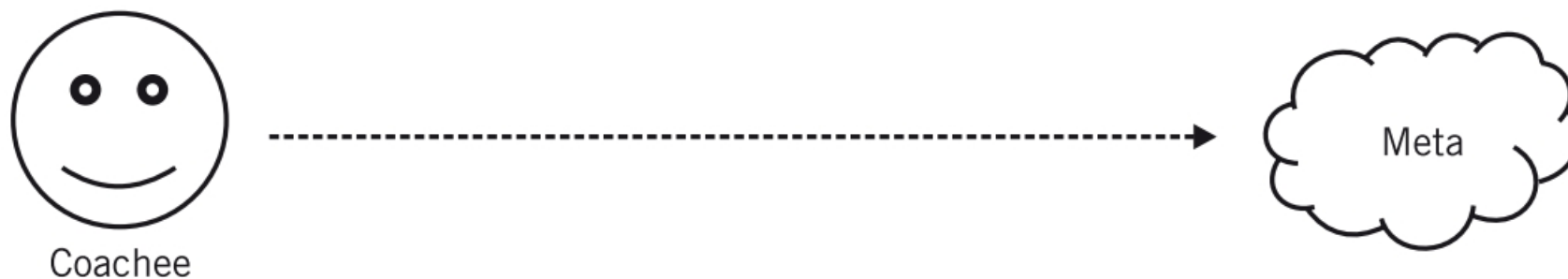


Figura 15.3 Um Coachee com senso de responsabilidade sobre uma meta.



Em seguida, acontece o entendimento da realidade atual e de quão difícil ou fácil ela pode ser para o Coachee.

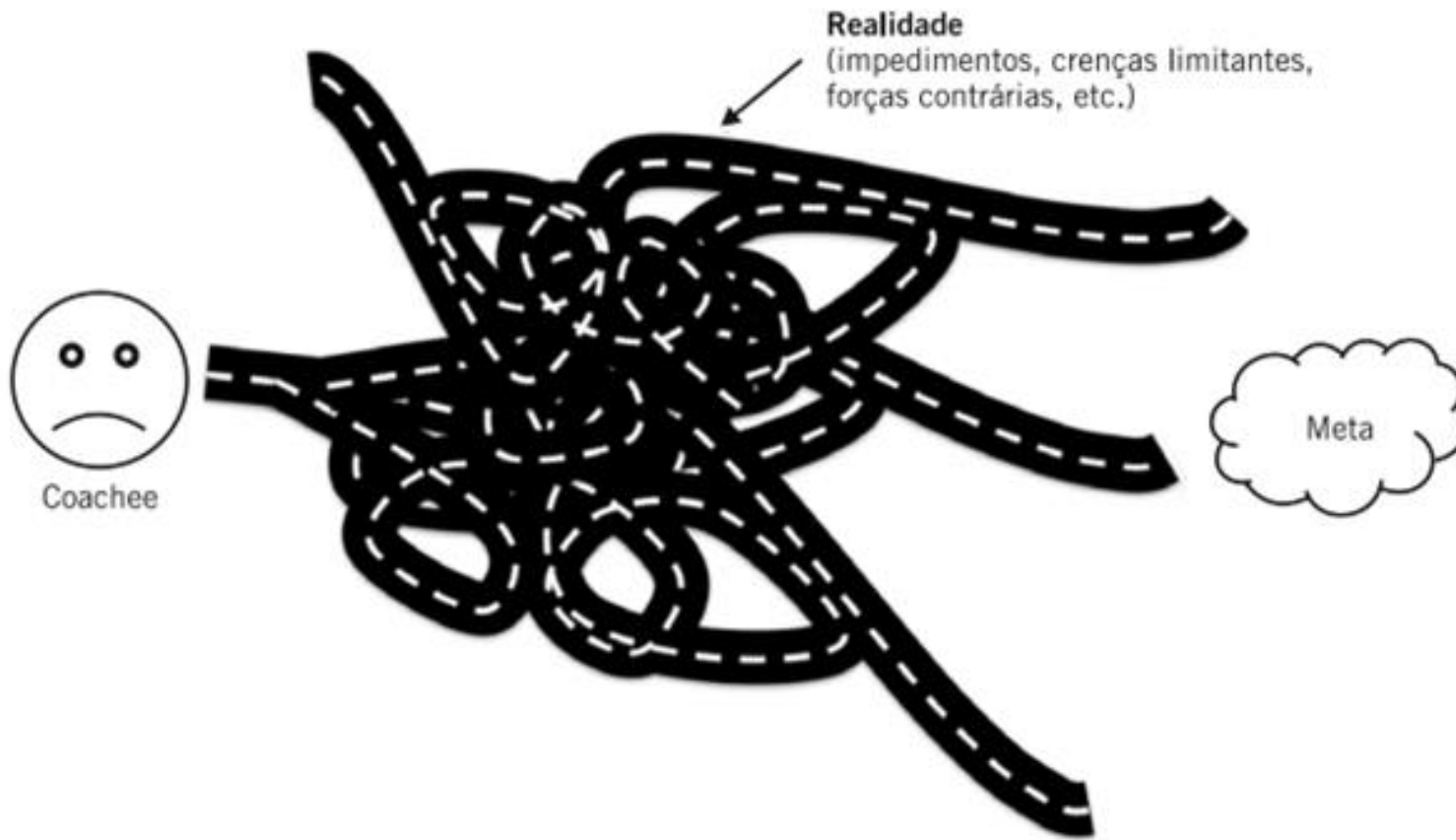


Figura 15.4 Como é a percepção da realidade por um Coachee.

O Coach ajuda o Coachee a buscar novos ângulos de visão e novos pensamentos (inclusive crenças) sobre sua realidade.

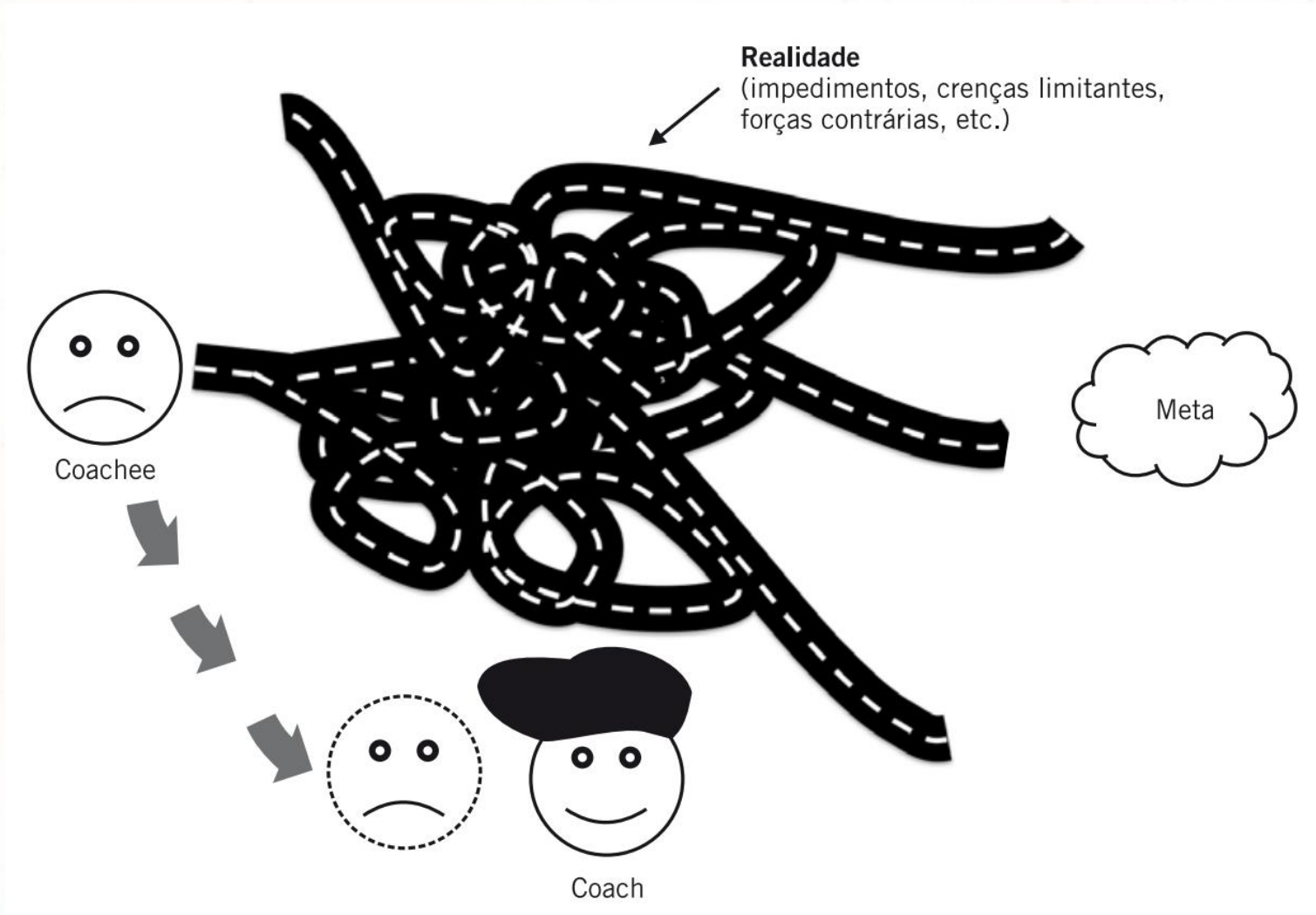


Figura 15.5 O Coach estimulando o Coachee a olhar para o problema por um novo ângulo.

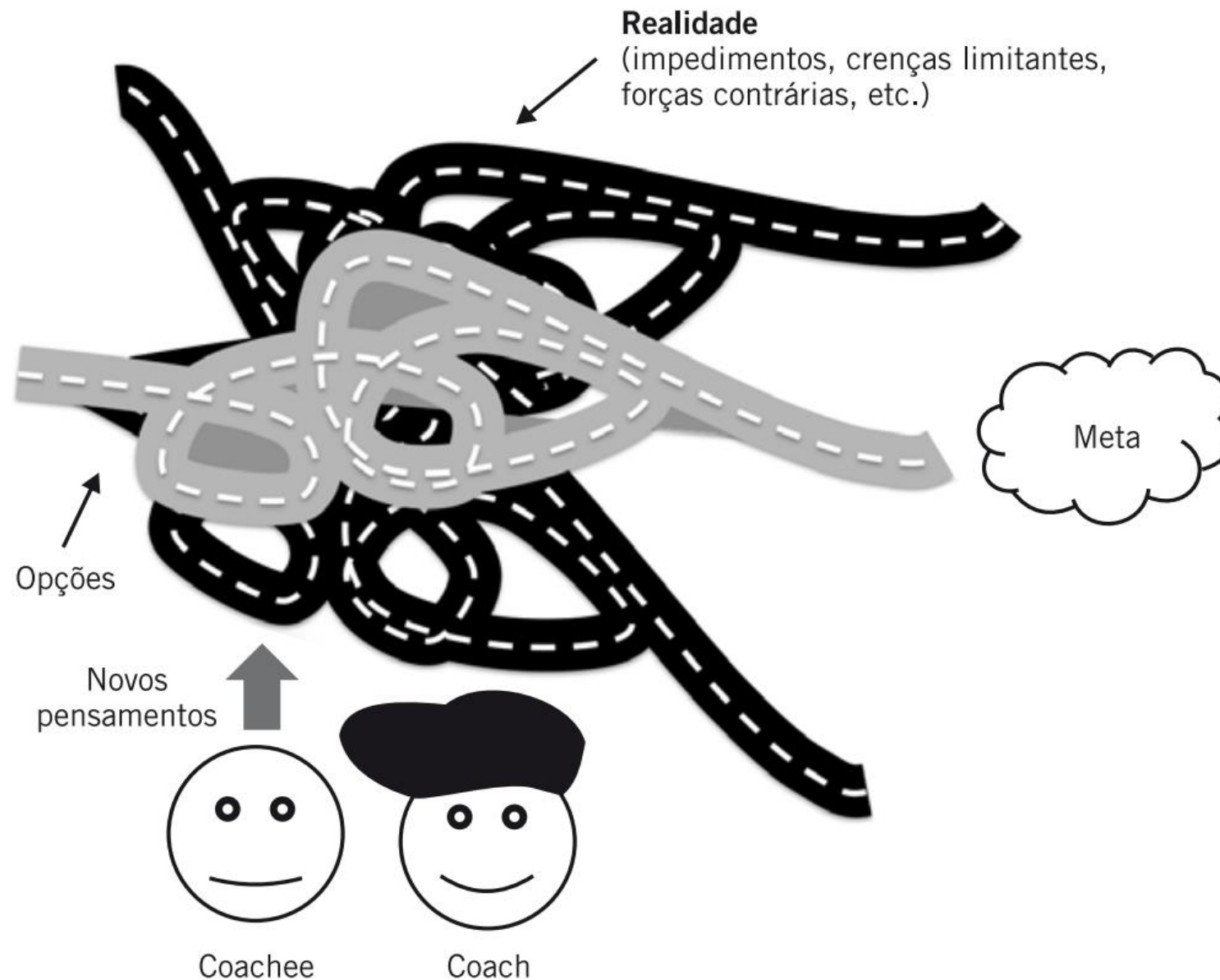


Figura 15.6 O Coachee com uma compreensão diferente a partir do seu novo ângulo de visão.



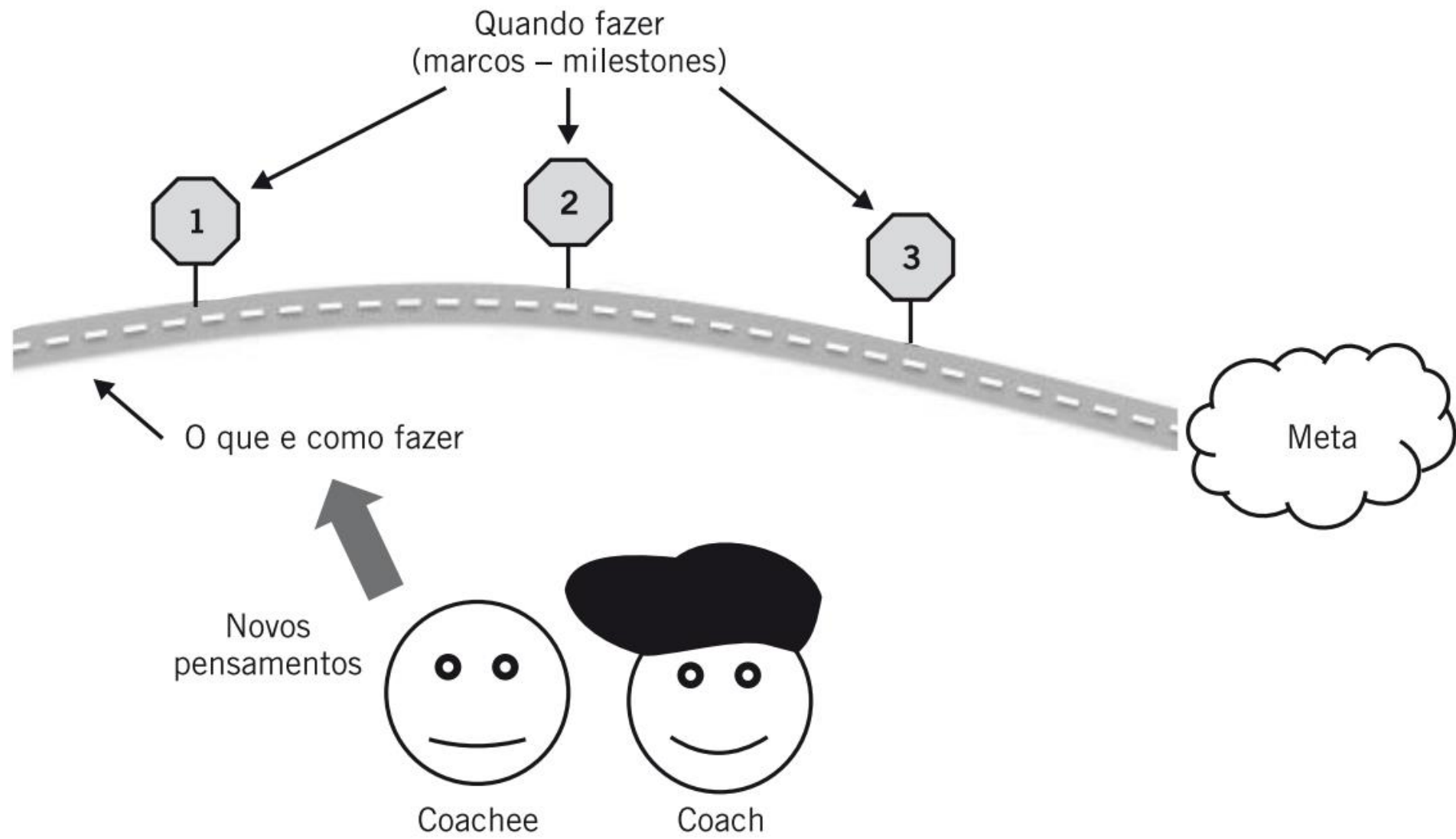


Figura 15.7 Visão do caminho traçado pelo Coachee.

Munido de novos pensamentos, o Coachee começa a visualizar como e quando trilhar o caminho rumo a uma meta.

Uma vez esboçado o caminho, o Coach ajudará o Coachee a fazer a “caminhada” em direção a seu objetivo.  
(Prikladnicki, 2014).

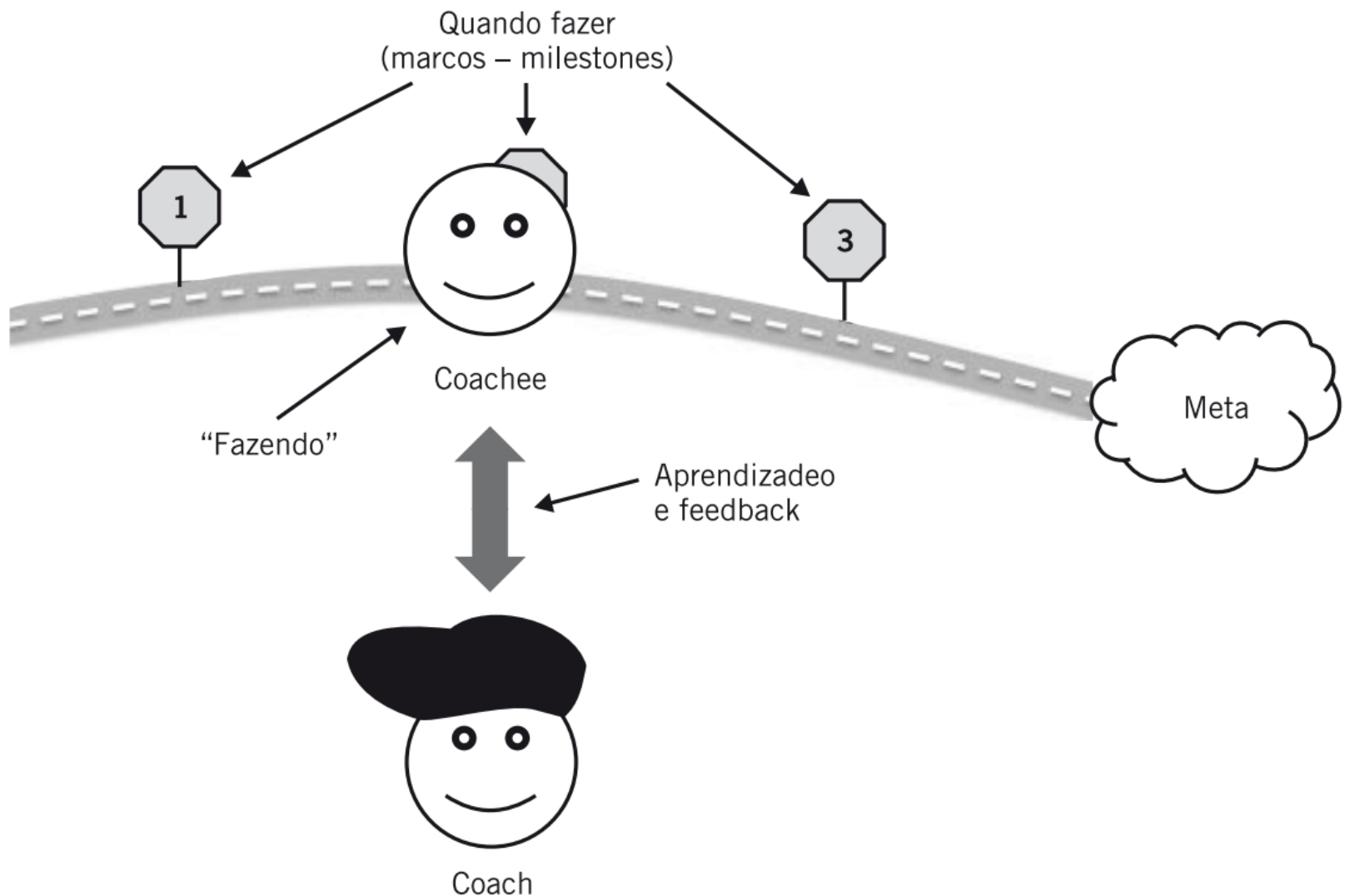


Figura 15.8 Continuação do processo de Coaching através dos ciclos de aprendizagem rumo a uma meta.

É importante observar que essa atuação do Coach estimulará uma aprendizagem contínua no Coachee, conseguida por meio de seus erros e acertos durante o caminho, de maneira que esse fluxo contínuo seja efetivo até o alcance da meta final.

## CONCLUSÃO

**É disso que Coaching trata:** ajudar um indivíduo ou time num processo de mudança para chegar a uma meta.

Esse processo de mudança gera uma necessidade de uma performance melhor e, para isso, é preciso desenvolver uma nova forma de pensar, de sentir e, principalmente, de agir.(Prikladnicki, 2014).



## **BIBLIOGRAFIA**

Pressman, Roger S. Engenharia de software : uma abordagem profissional [recurso eletrônico] / Roger S. Pressman, Bruce R. Maxim ; [tradução: João Eduardo Nóbrega Tortello ; revisão técnica: Reginaldo Arakaki, Julio Arakaki, Renato Manzan de Andrade]. – 8. ed. – Porto Alegre : AMGH, 2016.

Morais, Izabelly Soares de. Engenharia de software [recurso eletrônico] / Izabelly Soares de Moraes, Aline Zanin ; revisão técnica : Jeferson Faleiro Leon. – Porto Alegre : SAGAH, 2017.

PRESSMAN, Roger; MAXIM, Bruce. Engenharia de Software. Uma abordagem profissional. 8a. Ed. Bookman, 2016.

<https://integrada.minhabiblioteca.com.br/#/books/9788580555349/cfi/3!/4/2@100:0.00>

SOMMERVILLE, Ian. Engenharia de Software. 9. ed. São Paulo: Pearson Prentice Hall, 2011.

[https://bv4.digitalpages.com.br/?term=engenharia%2520de%2520software&searchpage=1&filtro=todos&from=busca&page=\\_14&section=0#/legacy/276](https://bv4.digitalpages.com.br/?term=engenharia%2520de%2520software&searchpage=1&filtro=todos&from=busca&page=_14&section=0#/legacy/276)

LARMAN, Craig. Utilizando UML e padrões: uma introdução à análise e ao projeto orientados a objetos e desenvolvimento iterativo. 3. ed Porto Alegre: Bookman, 2007.

<https://integrada.minhabiblioteca.com.br/#/books/9788577800476/cfi/0!/4/2@100:0.00>

Prikladnicki Rafael, Willi Renato, Milani Fabiano. Métodos ágeis para desenvolvimento de software / Organizadores, Rafael Prikladnicki, Renato Willi, Fabiano Milani. – Porto Alegre : Bookman, 2014.

IFSC. Ciclo de Vida Iterativo e Incremental. Wiki Instituto Federal de Santa Catarina, São José, out. 2006. Disponível em: <[https://wiki.sj.ifsc.edu.br/wiki/index.php/Ciclo\\_de\\_Vida\\_Iterativo\\_e\\_Incremental](https://wiki.sj.ifsc.edu.br/wiki/index.php/Ciclo_de_Vida_Iterativo_e_Incremental)>. Acesso em: 31 ago. 2017.