

Trilha XP



**Um relato sobre práticas de XP: Como começar?
Quanto tempo até dar resultado? Onde busco
conhecimento?**

Tiago de Mello



O XP não tinha morrido?

“Many Agile adoptions have treated **technical practices as secondary** compared to the management and team practices that some Agile frameworks emphasize.”

OBS: Segundo Google Trends, em Jan 2006 Scrum passou XP.



Kent Beck, Technical coach at Facebook. XP, TDD, patterns, JUnit, 3X, music.

Answered Jun 12, 2012 · Author has 150 answers and 1.1m answer views

It seems to me that I could answer this question at the level of **values, principles, and practices**. At the level of values, are there teams that are actively pursuing communication, simplicity, feedback, courage, and respect? Certainly. At the level of principles, are there teams shortening feedback loops, applying redundancy to complex issues, taking the humanity of participants into account, etc? Sure. At the level of practices, teams are applying all of the practices (and a bunch of new ones like A/B testing and continuous deployment). From this perspective **I'd have to say yes, teams are "doing" XP.**



Kent Beck, Technical coach at Facebook. XP, TDD, patterns, JUnit, 3X, music.

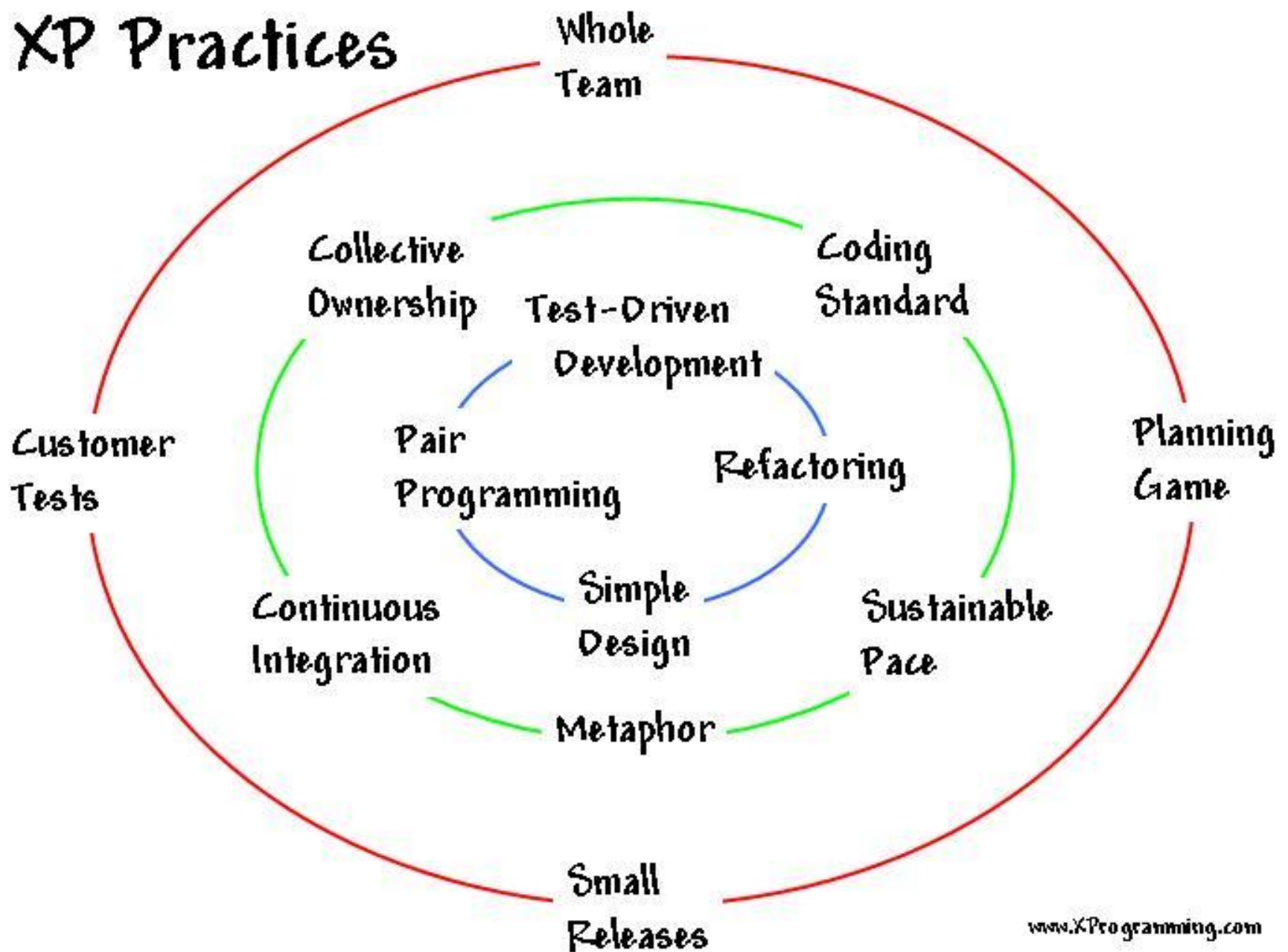
Answered Jun 12, 2012 · Author has 150 answers and 1.1m answer views

However, if I read your question as, "Are teams following all the practices to the letter," the answer is certainly no. Even teams applying a superset of the practices modify them based on experience.

The more concrete the discussion, the more likely teams are self-aware about using XP. At Facebook where I work there are lots of discussions about tools and frameworks, fewer-but-still-significant about practices, but very few discussions explicitly about principles or values. I would say we aren't "doing XP", but I'm not sure why a binary evaluation even matters.

XP Practices

XP Practices





O Ministério Kent Beck adverte, essas práticas têm um curva de aprendizado.

Tempo:
+3-24
meses

Investimento:

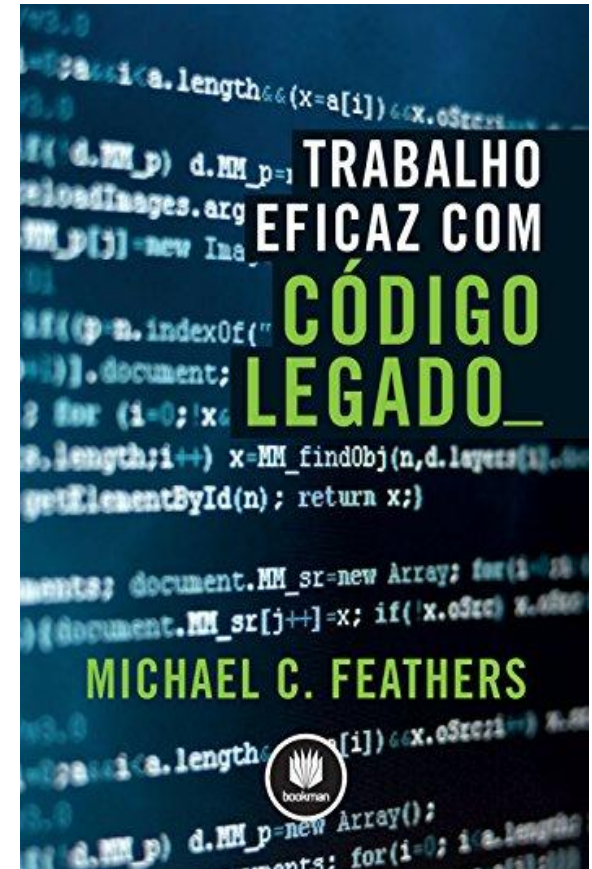
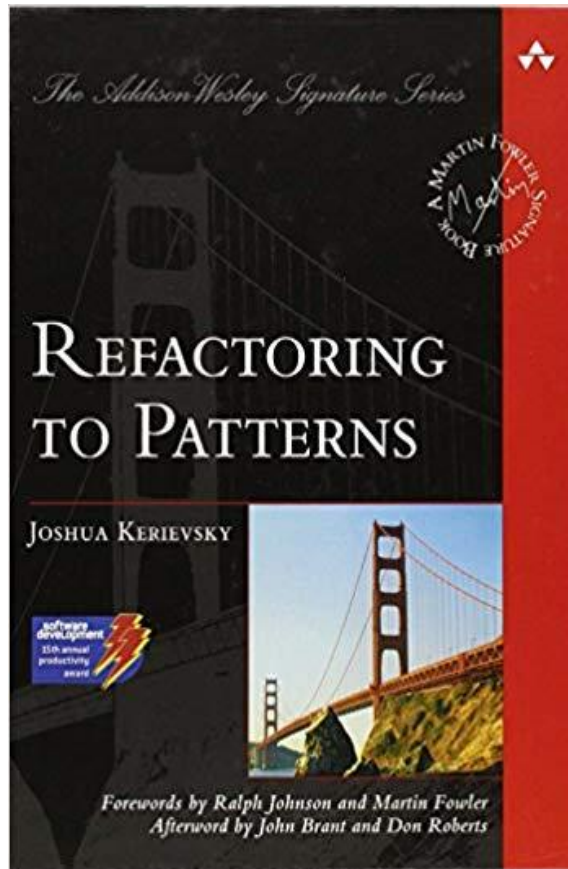
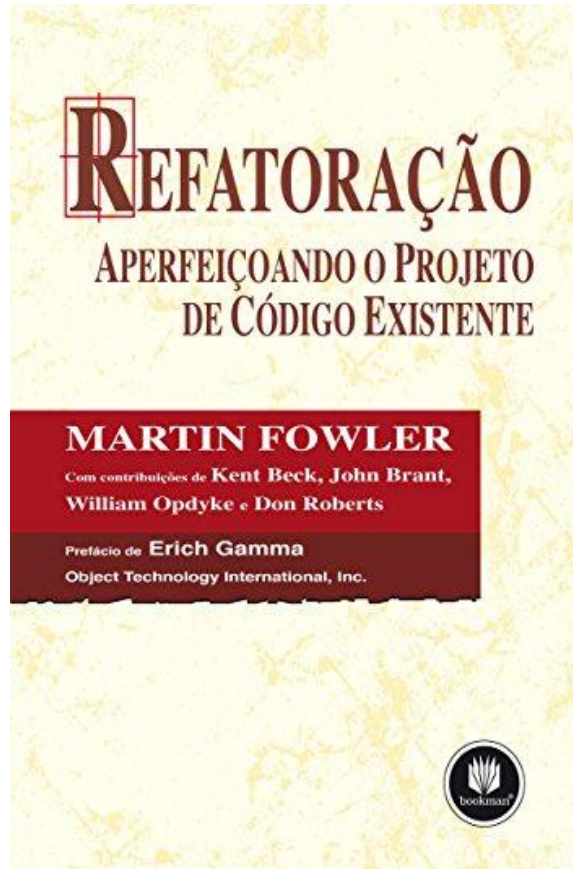
Menor produtividade durante o aprimoramento técnico.

Benefícios:

Menos defeitos e maior produtividade.

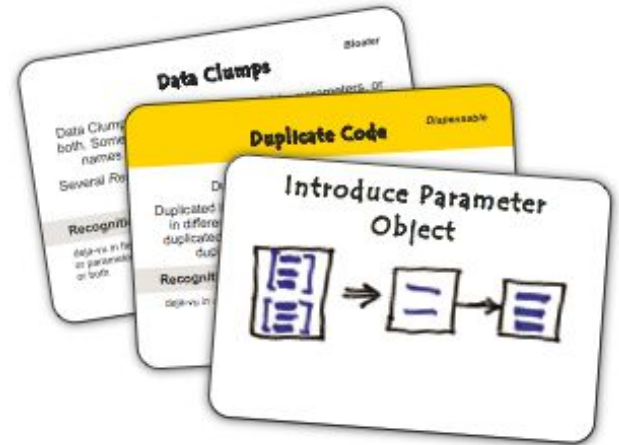
Segundo The Agile Fluency Model:
<https://martinfowler.com/articles/agileFluency.html>

Refatorar - Conhecimento



Refatorar - Como começar


- Clean Code;
- Dojos e Katas em time;
- Refactoring cards (QWAN);
- SOLID;
- Sonar, codecity e afins;



★ Dica: colocar regras de qualidade de código para quebrar o pipeline (se você não tem pipeline, volte duas casas).

Refatorar - Medir resultados

- Complexidade ciclomática;
- Quantidade linhas de código (por método);
- Quantidade linhas de código (por arquivo);
- Fat;
- Tangle;
- Métricas do Sonar;

 Dica: ter uma baseline, e criar cerimônias mensais ou trimestrais para acompanhamento.

TDD - Conhecimento

TDD DESENVOLVIMENTO GUIADO POR TESTES

KENT BECK



DESENVOLVIMENTO DE SOFTWARE ORIENTADO A OBJETOS, GUIADO POR TESTES

STEVE FREEMAN
NAT PRYCE



Prefácio por Kent Beck
Posfácio por Tim Mackinnon



WORKING EFFECTIVELY WITH UNIT TESTS

Jay Fields

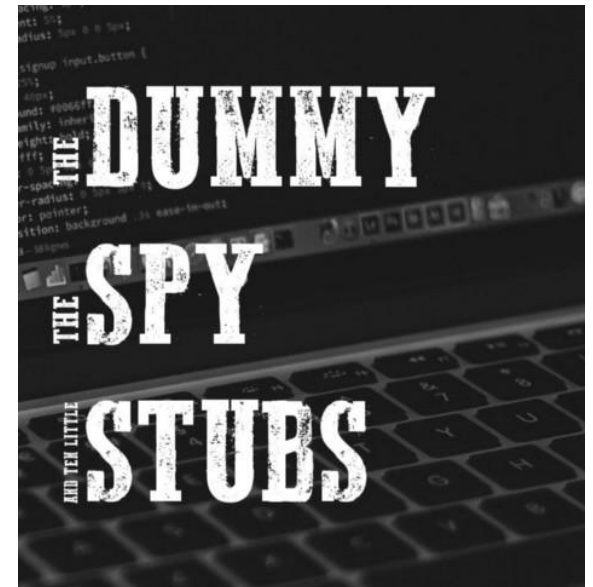
foreword by Michael Feathers



TDD - Como começar

- Testes automatizados;
- Dojos e Katas;
- Testando o que tiver valor a curto prazo;
- Testar estilos: Mockista VS Classista;
- Testes por mutação;
- Entender Mocks;

★ Dica: série Clean Coders - London VS Chicago.



TDD - Medir resultados

- % Cobertura de testes;
- % Cobertura de testes por mutação;
- % de testes VS alterações de código;
- distribuição da pirâmide de testes;

Testes automatizados:

- Essencial para trabalho em time;
- Essencial para CI e CD;

Pair programming - Conhecimento

- Prática;
- Cheat sheets:

<https://github.com/BillSchofield/pair-programming-advice>

<https://www.industriallogic.com/blog/a-few-tips-for-mob-programming/>

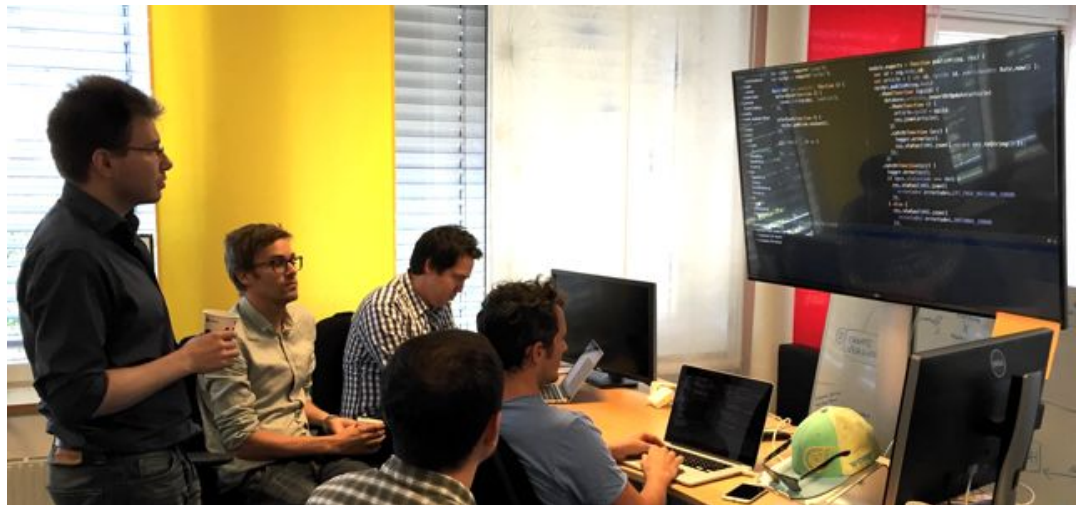
<http://blog.gdinwiddie.com/2007/08/18/pair-programming-techniques/>

Pair programming - Como começar

- Repasses, integração de desenvolvedores;
- Tarefas alta complexidade;
- Fazer os rodízios;
- Limitar o WIP;



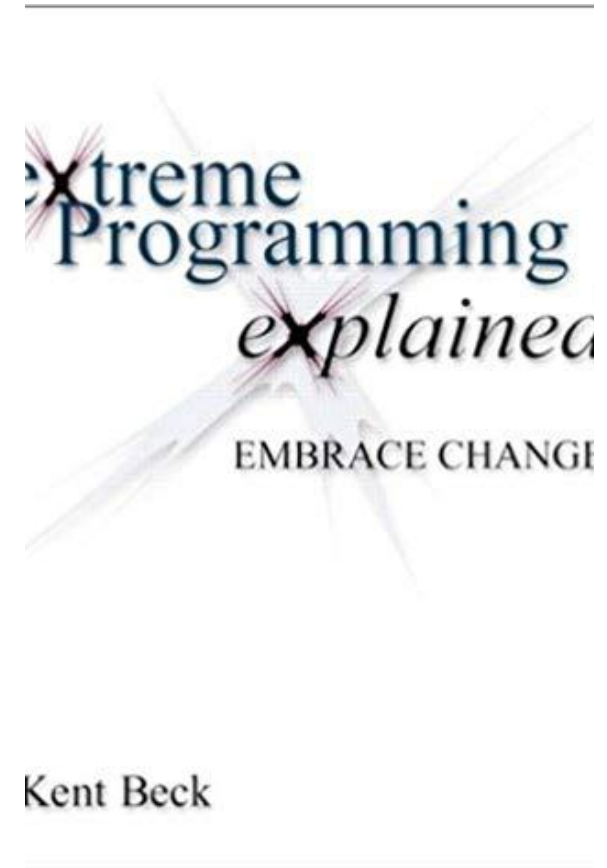
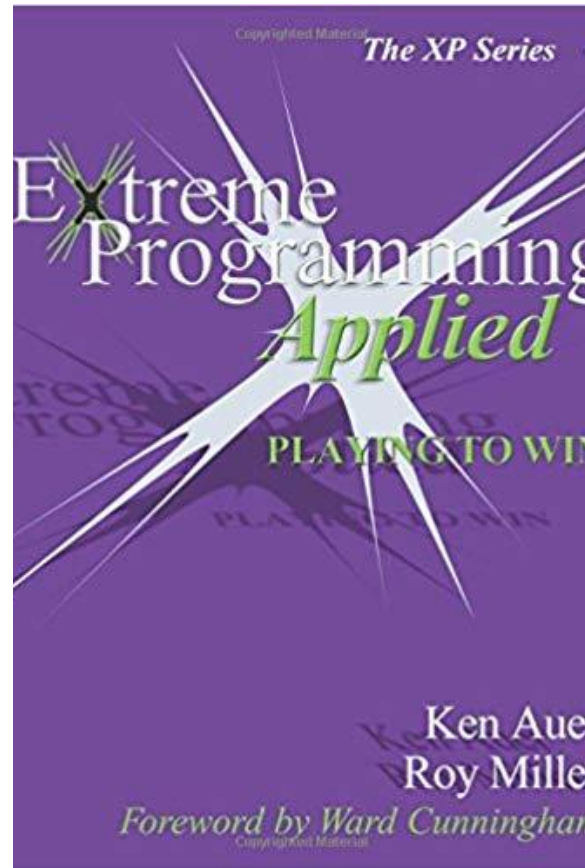
Dica: Mob Programming



Pair programming - Medir resultados

- Repasse de conhecimento;
- Lead time;
- WIP;
- Validado cientificamente
(<http://bit.ly/2UV9H8o>)
- Sustentabilidade da empresa com trabalho em time
(<https://twitter.com/RonJeffries/status/1114808214648967168>)

Design simples - Conhecimento



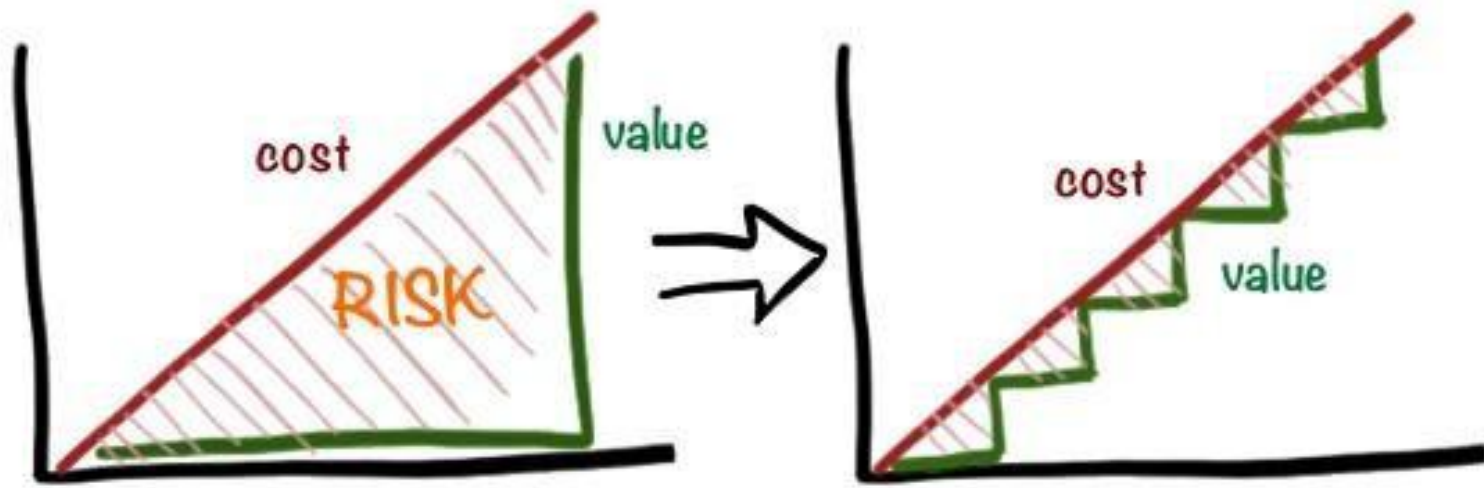
Design simples - Como começar

- Keep it Simple;
- You Are Not Gonna Need It;
- Não centralizar decisões de design de código ou arquitetura de código;
- Ter claro no código o domínio de negócio (com DDD, Clean Architecture...)
- Compartilhar com pessoas do negócio as decisões de código;



Design simples - Medir resultados

- Throughput;
- Produtividade
- Maior autonomia aos desenvolvedores;



Conclusão

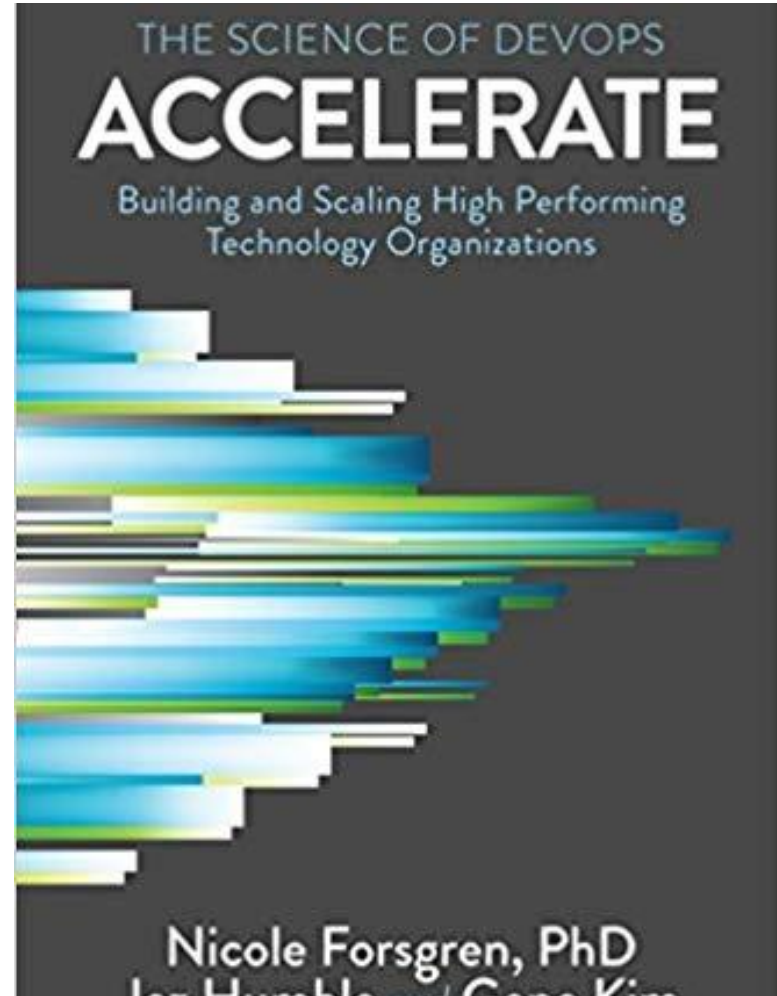
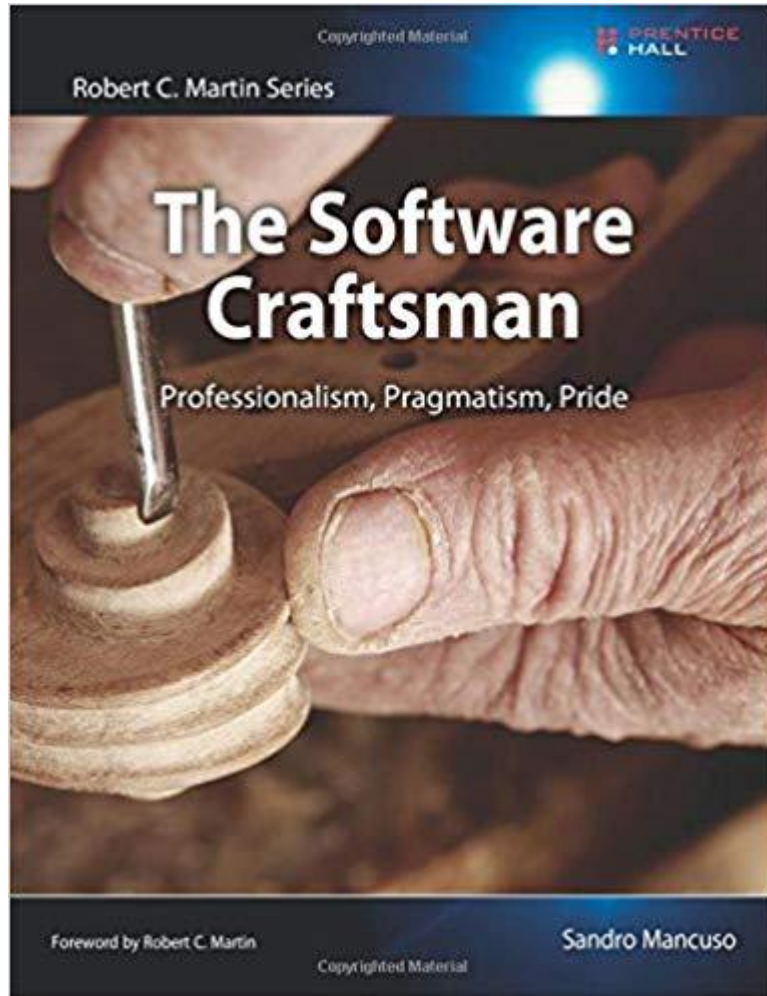


“Any fool can write code that a computer can understand. Good programmers write code that humans can understand.”
— Martin Fowler

Bônus I - Mais formas de demonstrar resultado

- Índice de retrabalho;
- Reaproveitamento código;
- Índice de falhas em alteração;
- Satisfação e facilidade de devs.

Bonus II



Bônus III

Links de artigos ou vídeos que possam ser interessantes:

<https://martinfowler.com/bliki/UnitTest.html>

<http://blog.cleancoder.com/uncle-bob/2017/05/05/TestDefinitions.html>

<http://butunclebob.com/ArticleS.UncleBob.TheThreeRulesOfTdd>

<http://blog.wingman-sw.com/archives/16>

<https://8thlight.com/blog/uncle-bob/2014/05/14/TheLittleMocker.html>

<https://martinfowler.com/articles/mocksArentStubs.html>

<https://www.thoughtworks.com/insights/blog/mockists-are-dead-long-live-classicists>

<http://www.everydayunittesting.com/2015/08/the-ugly-truth-about-emergent-design.html>

<http://www.ustream.tv/recorded/86157842>

<https://www.youtube.com/user/codurance/videos>

Pago:

<https://cleancoders.com/>

Bônus IV

Links de alguns katas específicos:

<http://butunclebob.com/ArticleS.UncleBob.ThePrimeFactorsKata>

<http://butunclebob.com/ArticleS.UncleBob.TheBowlingGameKata>

<http://thecleancoder.blogspot.com/2010/10/craftsman-62-dark-path.html>

Sites que têm mais katas:

<http://katas.softwarecraftsmanship.org/>

<http://codekata.pragprog.com/>

Questões

Palestra:

<http://bit.ly/XP-STARTER-TDC>

Contato:

 **[/in/tiago-de-mello/](https://www.linkedin.com/in/tiago-de-mello/)**

 **[@tiago_d_mello](https://twitter.com/tiago_d_mello)**

 **tiagodemello@mail.com**