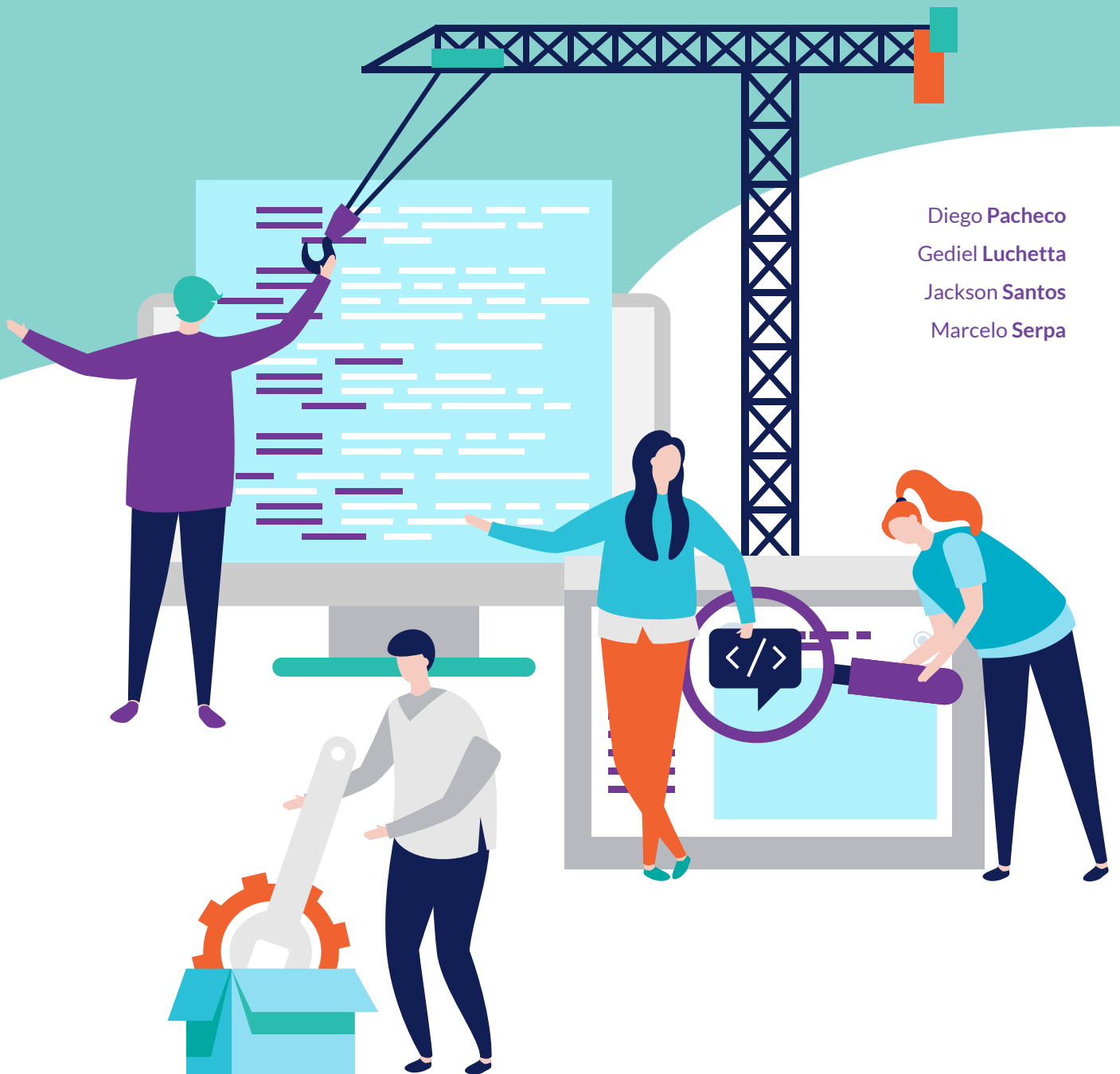


Arquitetura

Valor e Negócio

Diego Pacheco
Gediel Luchetta
Jackson Santos
Marcelo Serpa



Arquitetura de Software é uma disciplina que nasceu há muitos anos e sempre vai existir. Todo *software* possui arquitetura, por pior ou melhor que seja. Hoje em dia, Agile é o mainstream, pois grande maioria das empresas no Brasil e Europa trabalham com algum formato de metodologias ágeis. Nos EUA, especificamente no Vale do Silício, isso não é mais um *big deal* e já foi superado. Porém, no Brasil muitas decisões erradas são tomadas pelo pouco entendimento sobre arquitetura de *software* e a sua influência nos negócios. Por definição, arquitetura de *software* é um reflexo do negócio, porém muitas vezes a área técnica não consegue explicar a importância da arquitetura, certas escolhas técnicas e como isso pode impactar diretamente na experiência do usuário final. Os times de desenvolvimento de *software* e operação sempre tiveram muita dificuldade de mostrar o valor de certas decisões e muitas vezes o negócio percebe a tecnologia como algo complicado, abstrato e alto custo. Acima de tudo, a empresa não possui entendimento do porquê deve investir e se preocupar com certas decisões.

1. Arquitetura Corporativa e Frameworks

	DATA What	PROCESS How	NETWORK Where	ROLE Who	TIMING When	MOTIVATION Why
Scope (contextual)	List of things important to the business	List of processes the business performs	List of locations in which the business operates	List of business responsibilities	List of events significant to the business	List of business goals/strategy
Planner						
Business model (conceptual)	e.g., Semantic Model	e.g., Business Process Model	e.g., Logistics Networks	e.g., Workflow Model	e.g., Master Schedule	e.g., Business Plan
Owner						
System model (logical)	e.g., Logical Data Model	e.g., Application Architecture	e.g., Distributed System Architecture	e.g., Human Interface Architecture	e.g., Processing Structure	e.g., Business Rule Model
Designer						
Technology model (physical)	e.g., Physical Data Model	e.g., System Design	e.g., Configuration Design	e.g., Presentation Architecture	e.g., Control Structure	e.g., Rule Design
Detailed representations (out-of-context)	e.g., Data Definition	e.g., Program	e.g., Network Architecture	e.g., Security Architecture	e.g., Timing Definition	e.g., Rule Specification
Functioning enterprise	e.g., Data	e.g., Process	e.g., Network	e.g., Organization	e.g., Schedule	e.g., Strategy

Zachman Framework - Ontologias

Surgiu nos anos 80 através da IBM, o *framework* de arquitetura corporativa chamado de Zachman Framework. Nesse *framework* existem passos e públicos alvos como Desenvolvedores nas linhas mais inferiores e executivos nas linhas superiores. Esse *framework* foi um dos poucos que sobreviveu e ainda é utilizado. Existem outros *frameworks* como o TOGAF do Opengroup, que também é utilizado nos dias de hoje e possui cases. Esses 2 *frameworks* têm como propósito a questão importante de alinhar o negócio como a TI, porém muitas vezes leva o gestor das empresas a tomar a decisões erradas, de alto custo e que tecnicamente apresentam muitas falhas.

No início dos anos 2000, quando aconteceu a revolução dos métodos ágeis, ficou claro que a TI precisava entregar valor de forma mais rápida e contínua. Porém, esses *frameworks* de arquitetura corporativa, muitas vezes focam apenas em processo, pois é difícil para uma empresa de fora dizer como o negócio deve funcionar e replicar isso para os desenvolvedores. O resultado que se obteve nos anos 80 e ainda acontece nos dias de hoje é um custo elevado e alto índice de insatisfação dos usuários finais, bem como dos desenvolvedores, uma vez que esses processos *top down* são lentos e não focam nas 2 questões mais importantes para cada negócio, independente da indústria:

1. Satisfação do cliente final; 2. Progreção.

Priorizar somente os processos demanda muita energia com coordenação interna, deixando em segundo plano os resultados que vem de fora (*mindset* de produto).

2. CMMI & Certificações

Characteristics of the Maturity levels



CMMI - Níveis de Maturidade

CMMI tentou algo parecido, bem intencionado, mas também não deu certo, pois o processo muito lento, caro e não tem foco onde é necessário. Muitas vezes o código produzido por empresas que fazem CMMI é pior do que as que não o fazem. Hoje em dia isso praticamente sumiu do *mainstream*, graças aos métodos Ágeis. Contudo, ter uma certificação não é garantia de sucesso, ter uma entidade dizendo que sua empresa está no nível 1 ou 3 pode ser ilusório: não aumenta a receita, nem deixa os usuários finais mais satisfeitos, assim como não estimula os colaboradores, pelo contrário, rigidez de processo é algo mau visto pela geração dos *millennials*. Hoje em dia precisamos pensar em uma experiência para o desenvolvedores assim como pensamos para os clientes (isso se chama EX - Employee Experience). Certificações contribuem pouco para o resultado final de uma empresa. Fazer produtos, inovar e atender bem os clientes requer tempo e trabalho duro, logo não tem como existir uma fórmula mágica para resultado instantâneo, o pensamento imediato contribui para não se ter resultados. Qualquer transformação ou evolução precisa de tempo para ser entendida, testada, experimentada e evoluída.

3. Transformação Digital

Enterprise Digital Transformation Accelerates, Fortune Global 500 Become Trendsetters

64%

64% of enterprises become the explorers and practitioners of digital transformation.

Source: IDC

67%

67% of CEOs of Global 2000 enterprises have digital transformation at the center of their corporate strategy.

Source: Gartner

G100

Fortune Global 100 enterprises will work out plans for digital transformation by the end of 2017.

Source: IDC

Transformação digital - Números (2017)

Toda empresa financeira, fintech, health tech e similares nos dias de hoje, buscam a culturalização da transformação

digital. O digital é melhor, mais rápido, atende o cliente com eficácia e reduz custos. Apesar de ser um termo já existente desde o ano 2000, isto se tornou tão desejado somente em 2019, pois houve mudança na percepção de valor em relação a DevOps, Cloud e Inovação. Está aliado ao fato de Startups que estão utilizando Lean Startup e outras formas ágeis de descobrir o Product Market Fit. Ou seja, levamos quase 20 anos para viabilizar ao negócio que, utilizando a tecnologia de forma diferente, podemos inovar e sair à frente do mercado.

Muitas empresas do Vale do Silício tiveram essa visão, por isso, elas dominam o mundo atualmente. Porém, ainda demoramos para perceber a importância do *software* nos negócios. Claro, a TI tem 50% ou mais nessa equação, uma vez que temos dificuldades de vender ideias e mostrar aos executivos (C-Level) o futuro.

4. Aspecto negativo dos métodos Ágeis



Agile Manifesto - Início dos anos 2000

Métodos ágeis trouxeram muitas evoluções para o desenvolvimento de *software*, tais como foco em entregas frequentes, excelência técnica, integração contínua, entre outras. Historicamente, os times ágeis sempre menosprezaram a disciplina de Arquitetura de Software.

O resultado negativo é percebido em softwares de larga escala, por exemplo, em empresas como Spotify, Netflix, Uber e Apple, que trabalham em times de plataforma e *times core*, que são focados em entregar *softwares*, ferramentas e serviços para os desenvolvedores. Dessa forma o *gap* de arquitetura é resolvido. Porém, em empresas de médio porte e até mesmo na indústria nacional onde cada vez é mais difícil ter profissionais qualificados, muitas vezes essa falta de arquitetura tem um efeito mais alarmante, resultando em desperdício financeiro, soluções ruins de experiência para o usuário final, escolhas técnicas inferiores que desmotivam os desenvolvedores e aumentam o *turn over*.

5. O poder da visualização



Copo da Starbucks - Exemplo de Kanban

Métodos como Kanban e Design Thinking são mais visuais. Não apenas como forma de entender melhor, mas concretizar ideias e ter certeza de que todos estão falando a mesma língua. Kanban significa sinal visual. No copo da cafeteria starbucks podemos ver um excelente exemplo, pois quando você faz o seu pedido e efetua o pagamento, o barista faz o café e você retira no balcão. Isso é um fluxo e as decisões ou parâmetros do cliente estão presentes no copo de forma visual.

Dentro do Kanban acredita-se que um fluxo visual é mais fácil de gerenciar, evoluir e tomar decisões. Historicamente, quando falamos de arquitetura de *software* visual, o mercado respondeu como UML (*Unified Modeling Language*) sendo uma notação que teve o seu valor, porém foi superestimado ao fazer diagramas para tudo (por conta do mau entendimento de RUP, consultorias e fábricas de *software* focadas em CMMI). Além disso, UML não deixa claro porque certas decisões de *software* precisam ser tomadas. Alguém poderia fazer um diagrama de use cases ou até mesmo um diagrama de DFD em nível zero de análise estruturada, entretanto, este tipo de diagrama é focado em fluxo e processo. O problema ocorre em certas decisões que vão além de um fluxo de negócio - logo existe um grande espaço em branco na visualização da arquitetura de *software*.

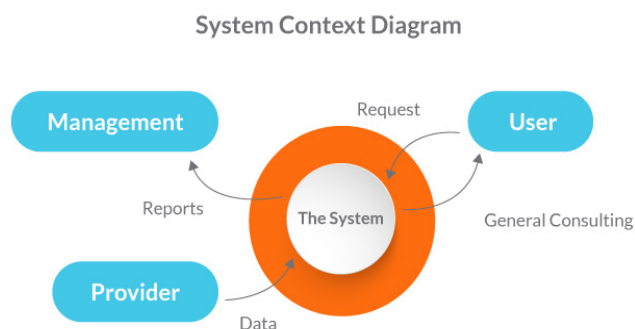
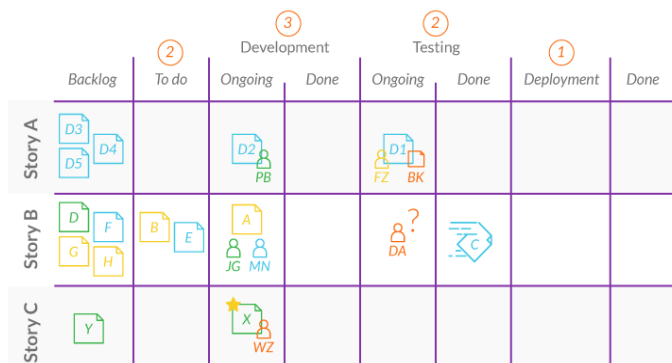


Diagrama DFD - nível ZERO



Lean Business Model Canvas

Kanban board: 8 columns with swim lanes



Os parâmetros usados são exemplos, sendo alterados conforme a necessidade da empresa

Quadro de Kanban - Fluxo de dev

7. Avaliação com Score Visual



O modelo de score visual desenvolvido pela ilegra tem como pilar o modelo 4x4 também da ilegra.

Ao nos basearmos no modelo 4x4 garantimos que não estamos dando uma cartilha pronta para nenhum cliente e sim levando em conta o seu contexto e com isso tomando as melhores decisões técnicas, dando o melhor acompanhamento para a evolução do cliente – totalmente voltado para o aprendizado e entendimento do porquê ir ou não para certos caminhos.

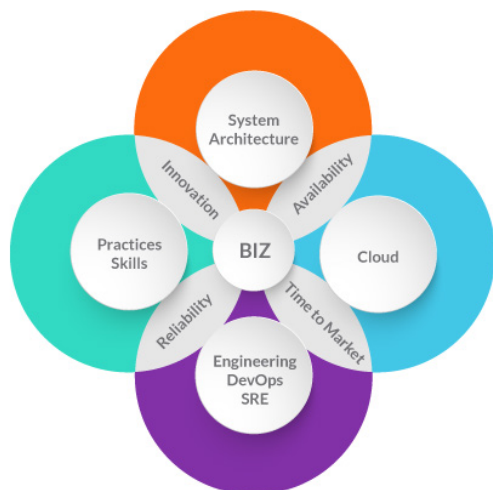
8. Evolução Contínua

Score corporativo, por sistemas ou serviços (dependendo do tamanho do arcabouço de *software* de cada empresa) é um *snapshot*, ou seja, uma foto atual da realidade, que se for medido de forma recorrente, mostrará a evolução da TI no sentido de atender as estratégias do negócio e se constituirá em uma abordagem para melhoria contínua.

9. Nenhuma regra é universal

Toda regra possui um contexto. Assim como cada negócio tem as suas verdades e os seus valores. Acreditamos que a evolução de qualquer *software* em uma empresa precisa ser sob-medida, ou seja, contextual. Conforme os valores e a realidade de cada mercado, é necessário olhar para práticas e modelos diferentes de arquitetura antes de optar pela utilização de microsserviços e entender se a arquitetura está correta. O valor gerado para o negócio são entregas mais frequentes, porém isso nem sempre ocorre, porque muitas vezes microsserviços sem Devops, Chaos Engineering ou Cloud/Containers não funcionam. Então, não se

6. Modelo de Arquitetura 4x4



Os parâmetros usados são exemplos, sendo alterados conforme a necessidade da empresa

Modelo Arquitetura 4x4 - by ilegra.com

O modelo de Arquitetura 4x4 da ilegra resgata a visão de negócio alinhado a estratégia de arquitetura de *software* e tecnologia de uma empresa. Quatro valores centrais são definidos pelo cliente e os outros quatro são definidos pelo time de arquitetura, ajudando sua empresa a entender porque certos movimentos tecnológicos são necessários. O modelo é 100% customizável, ou seja, as principais estratégias (4) de cada negócio são levadas em consideração para determinar as principais estratégias da TI (4).

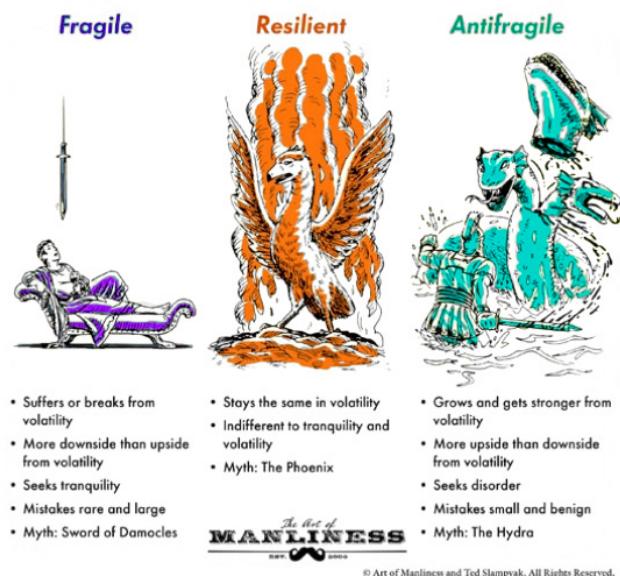
trata de uma tecnologia ou arquitetura específica, mas sim a direção e o seu aprendizado durante esse processo. Falar de aprendizado é muitas vezes visto como algo amador se uma empresa de consultoria não tiver respostas, mas as respostas são reveladas através de um processo de aprendizado ao invés de uma cartilha pronta e imutável. O score tem como resultado 5 possíveis notas que são A, B, C, D e E. A nota A é o score mais alto e E tem muitos pontos a evoluir.

10. Entendimento é Poder! Mais importante do que fazer

Qualquer empresa pode fazer qualquer coisa, usar qualquer arquitetura e qualquer *software*, mas o mais importante é saber o porque ela toma todas essas decisões e essa consciência muitas vezes não existe em algumas empresas, que estão aprendendo a fazer *software* de forma moderna e estão aprendendo a fazer produtos, daí a necessidade da cópia – porém, isso não funciona, pois copiar significa pegar apenas o que o é visível ou seja o produto final e não os princípios que levaram a esse resultado. Esse é um conceito velho em Lean e o Dr Deming já dizia que as empresas que copiassem a Toyota iam falhar e não virariam a Toyota. Quer um exemplo mais novo? Muitas empresas copiam o modelo do Spotify. Será que todas estão se tornando um Spotify? O Spotify não virou o que é, copiando outra empresa, ele foi original e inovador. Então, a forma de pensar é outra.

11. Necessidade de aceitar a falha

Amazon.com é famosa por aceitar a falha em diversos níveis, desde arquitetura de *software* e sistemas distribuídos na *cloud*, bem como nos negócios através de inovação. É praticamente impossível fazer inovação disruptiva sem falhas, porque no final das contas tudo é sobre descobertas. Não existe uma fórmula mágica anti-falhas, mas é possível aprender rápido e utilizar o que faz sentido para cada contexto com o melhor que existe em tecnologia. Para uma empresa que não tem uma estrutura tecnológica de peso é impossível sair de 0 para 1 sem falhas, mesmo com ajuda de uma consultoria. Mas por que? Porque quem tem que aprender é a empresa e não a consultoria. É como aprender a andar de bicicletas, os tombos fazem parte do aprendizado. Existe um conceito moderno de aceite e recuperação de falhas chamado de Anti-fragilidade que pode ser aplicado a sistemas bem como para negócios. A premissa é muito similar a premissa de Lean Startup onde aceitamos que vamos falhar e isso é normal, desta forma não estamos buscando evitar as falhas, mas sim ter certeza que nós conseguimos contorná-las e nos recuperar delas, esse é um *mindset* completamente diferente.



Anti-Fragilidade - Conceito

12. Oxigenando Negócios através da Arquitetura de Software

Acreditamos em oxigenação dos negócios através de novas tecnologias, arquitetura de *software* e boas decisões. Porém, essa oxigenação possui um contexto para cada cliente, colocado de forma visual no modelo de arquitetura 4x4 e no score visual dos sistemas. É necessário que haja educação do negócio, ele deve entender porque tomar certas decisões tecnológicas, bem como seu impacto na corporação, como diferencial competitivo na era digital, através de uma abordagem que muda a percepção de valor sobre a tecnologia com *mindset* de produtos, aceitação de falha e muito aprendizado. Não trabalhar a arquitetura de *software* de forma correta pode levar a vários problemas como:

- *Sistemas que não escalam (usuários e nem desenvolvedores).*
- *Sistemas que não tem uma experiência ideal para o usuário final (por ser lento, cair toda hora, ou ocorrem falhas que impactam em outras funções por falta de isolamento)*
- *Complexidade e lentidão para entregar novas funcionalidades (quando a arquitetura é complexa, ou tem um design ruim).*
- *Custo elevado (pela falta de um bom design e de boas práticas de engenharia de software).*

Arquitetura de *software* é crucial nortear as decisões técnicas que irão alavancar o negócio e entregar experiências únicas, inovadoras e duráveis.

13. Referências

<https://medium.com/@adamsconsulting/huawei-digital-transformation-fortune-global-500-companies-e907d92b245d>
https://en.wikipedia.org/wiki/Capability_Maturity_Model_Integration
<https://pt.wikipedia.org/wiki/TOGAF>
https://pt.wikipedia.org/wiki/Zachman_Framework
<https://www.opengroup.org/togaf>
<https://medium.com/@ahuijsen/scrum-and-the-agile-manifesto-f35fbbd4d0c7>
<https://www.scrum.nl/blog/to-scrum-to-kanban-or-to-scrumban-that-is-the-question/>
http://www.macoratti.net/13/08/net_dfd2.htm
https://pt.wikipedia.org/wiki/Business_Model_Canvas
<https://www.artofmanliness.com/articles/beyond-sissy-resilience-on-becoming-antifragile/>

David J. Anderson - *Kanban*

Corey Ladas - *Scrumban: Essays on Kanban Systems for Lean Software Development*

Sidney Dekker - *Drift into Failure*

Alexander Osterwalder - *Business Model Generation: A Handbook for Visionaries, Game Changers, and Challengers*

Eric Ries - *The Lean Startup: How Today's Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses*

W. Edwards Deming - *Out of the Crisis (The MIT Press)*

James Coplien - *Lean Architecture: For Agile Software Development*

Oxigenamos negócios para um mundo inquieto.

Somos uma empresa global de design, inovação e software.
Desenvolvemos estratégias e experiências de cliente que
transformam negócios.



Porto Alegre +55 51 3212-8444 | São Paulo +55 11 3171-0555

comercial@ilegra.com

www.ilegra.com