

Como escalar AGILE numa empresa de sistemas legados

Criar Produtos, planejar e executar
Projectos de Software, num tempo em
que as empresas são software!

As empresas com sistemas legados têm-se confrontado nos últimos anos, de forma exponencial, com uma concorrência crescente de quem tendo nascido digital, criou os seus sistemas de forma ágil e flexível, alinhados com os métodos, as ferramentas e práticas do *Continuous Integration / Continuous Delivery*. Estas abordagens garantem uma grande proximidade aos clientes e são uma enorme vantagem competitiva.

Por outro lado, a dificuldade das metodologias tradicionais do IT em garantirem projectos em tempo e em custo, gerou uma pressão que teria de produzir resultados nos métodos, nas ferramentas e na oferta dos principais intervenientes neste mercado.

É neste contexto que o Manifesto AGILE nascido em 2001 é hoje praticamente consensual. Segundo algumas fontes o Agile ultrapassa já em difusão, 80% das empresas.

Como tudo o que é novo, gera novos desafios, novos riscos e dificuldades. A abordagem AGILE que enfatiza a definição contínua dos requisitos, uma colaboração mais forte entre a equipa, antecipação de testes e aumento da sua frequência, bem como ciclos curtos de entrega de software, não está isenta de dificuldades, particularmente em empresas de sistemas legados!

Esta problemática é o tema central deste documento!

Introdução

Temas

1. O estado da arte das TI's, na era do Digital
2. As metodologias Agile, Scrum e Kamban, o Lean e o DevOps
3. Como escalar Agile numa empresa de sistemas legados!
4. O tema do Defice Tecnologico, "Technological debt"
5. Casos e Recomendações
6. Referencias bibliográficas

1. O estado da arte das TI's na era Digital

Continuous Delivery
Continuous Integration
Continuous Provisioning
Continuous Testing

- Ao contrário do paradigma anterior, o das aplicações monolíticas, tudo é feito de forma contínua e holística.
- As empresas de sistemas legados irão cada vez mais usar na construção dos sistemas, engenheiros altamente qualificados, ao contrário da forma tradicional baseada em recursos menos qualificados, em que uns especificam, outros desenvolvem e outros operam.
- O Governo dos Sistemas e do Desenvolvimento Aplicacional passa a ser tema central das empresas com sistemas legados, como o é nas empresas nativas digitais;
- A Cloud, bem como as metodologias assentes no Agile e DevOps, são o novo normal.

2. O Manifesto Agile

Valores do Manifesto AGILE (2001)

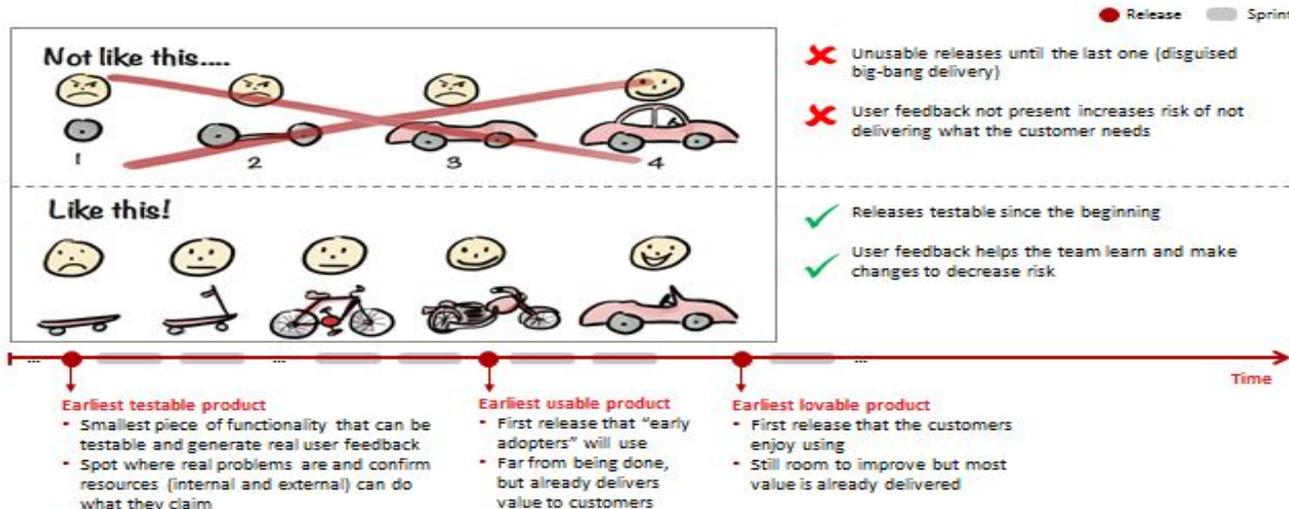
- Os indivíduos e suas interações são mais importantes de que procedimentos e ferramentas
 - Software a funcionar é mais importante que documentação compreensível
 - A colaboração com o cliente é mais importante que a negociação de contratos
 - A capacidade de responder as alterações, é mais importante do que responder a plano
 - Procedimentos, ferramentas, documentação e contratos são úteis, mas não o mais importante.
- As organizações com melhor desempenho ambicionam “be agile” não “do agile” com kpi’s assentes em *Outcomes* (crescimento, rentabilidade, fidelização, qualidade, serviço) em vez de *Outputs* (linhas de código ou número de novos produtos)!
- Colaboração, auto - organização e adaptação contínua são chaves para se ser ágil.

2. Os 12 princípios da metodologia Agile

1. Garantir satisfação cliente, através da entrega rápida e contínua de software funcional;
2. Até mesmo mudanças tardias ao âmbito do projecto são bem vindas;
3. O software funcional é entregue frequentemente;
4. Cooperação constante entre as pessoas que entendem do negócio e os “developers”;
5. Projectos desenvolvidos em torno de indivíduos motivados;
6. O método mais eficiente de comunicar é a conversa cara a cara;
7. A medida do sucesso é o software funcional em operação;
8. O ritmo deve ser constante e continuado ao longo do tempo;
9. Atenção constante à excelência técnica e ao bom *design*;
10. Simplicidade , a arte de maximizar a quantidade de trabalho não feito;
11. As melhores arquiteturas, requisitos e projetos emergem de equipes auto-organizadas;
12. A intervalos regulares a equipa reflete sobre como tornar-se mais eficaz e em seguida ajusta o seu comportamento;

2. A mentalidade de quem desenvolve em Agile

Mindset



As good practice, many times, there can be a 1w "Sprint 0", where the simplest functionality is implemented E2E, to confirm the team is able to deliver at proper quality and time span

2. Quality at speed



Negócios e produtos diferentes, justificam um ênfase diferente , numa ou outra prioridade. Mas Qualidade e Velocidade não são conceitos antagónicos num contexto Agile!

TEST AUTOMATION é a fórmula que permite o melhor dos dois mundos, incluindo o conceito de Parallel Testing. A automação dos testes, associada a uma abordagem de CONTINUOUS TESTING é a resposta para a agilidade com qualidade, ou seja, garante um feedback permanente e antecipado dos problemas

A equipa Agile assegura que a automação irá acontecer em toda a pirâmide de testes, dos Testes Unitários, aos Testes do GUI, passando pelos Componentes, Integração e API's.

Testes deixa de ser sinónimo de Testes Funcionais de Utilizador no final do projecto.

2. DevOps

A metodologia DevOps, visa também maximizar o retorno no mais curto espaço de tempo, mas num contexto de máxima fiabilidade e ausência de disrupção.

Em vez de minimizar os tempos de desenvolvimento, pretende minimizar tempos de indisponibilidade, interrupções, erros e problemas de sistemas, através da automação e estreita colaboração numa mesma equipa.

O DevOps é central para as empresas com sistemas legados e grandes operações institucionalizadas, onde a agilidade e a rapidez de desenvolvimento não são suficientes para garantir uma operação eficiente.

Sempre que o DevOps não está firmemente ancorado na cultura da empresa, e na metodologia Agile, ou seja é desvalorizado, a prazo, os sistemas podem regredir e tornarem-se de novo complexos e instáveis!

2. O Agile, o Lean e o Dev/Ops

O **Agile** foca no “*how, when, who and where*”, expressas em interações rápidas e eficientes.

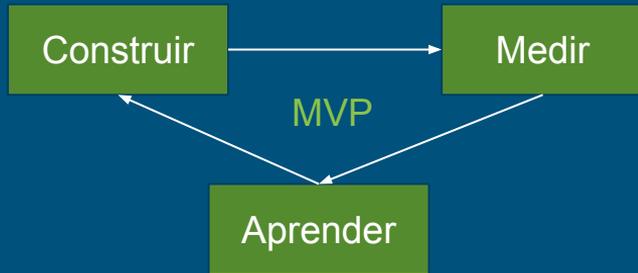
A metodologia **Lean**, focam mais em validação do Produto, ou seja no “*what and why*” das user stories e no seu sucesso. Lean está mais centrada no que pomos em frente aos clientes e nos dados. Visa delimitar o âmbito desenvolvendo apenas o que é absolutamente essencial; é a fonte do conceito de MVP (Minimum Viable Product)

O DevOps visa também maximizar o benefício para o cliente. Mas ao contrário do desenvolvimento rápido, privilegia a fiabilidade dos sistemas e a minimização do tempo de indisponibilidade. Usa a automatização e minimiza o Défice Tecnológico que potencialmente advém das fases de desenvolvimento.

Agile, Lean e DevOps, diferentes no foco e na aproximação metodológica, têm em comum possibilitarem o Continuous Integration e o Continuous Delivery (CI/CD), através de uma intensa colaboração, eliminando o desperdício e otimizando as interações

2. O MVP - Minimum Viable Product

- Cada vez que um novo produto ou novas funcionalidades são construídas e disponibilizadas, é exigida a criação de mecanismos que permitam o acompanhamento e a medida do impacto nos clientes e nas suas interações
- A definição de um MVP na fase de construção da visão, permite a gestão do processo de “construir, medir e aprender”



- Em vez de só ter um processo para aprender com os dados em fase subsequente ao lançamento, pretende-se aprender rapidamente após o lançamento da nova funcionalidade e se não adequada, repetir com novo desenvolvimento tão rápido quanto possível, para acelerar a decisão
- O fracasso é aceite como normal

2. Scrum versus Kanban

Scrum e Kanban são as duas *framework* Agile mais conhecidas.

Na *framework* Agile (SCRUM), existe um tempo fixo (sprint) para o desenvolvimento de um conjunto de User Stories (âmbito). As que não forem terminadas voltam para o *backlog* das user stories a implementar.

Na *framework* Agile (KANBAN) não existe a componente de um tempo limitado e fixo. Pelo contrário, o plano é feito pela equipa (âmbito fixo) tendo em vista a operacionalização de um conjunto determinado de histórias que tem de ficar prontas nessa fase.

Em ambos os casos o tempo inclui tudo, descoberta, desenvolvimento, testes e operação.

Em aplicações legadas em que não seja fácil a construção de User Stories, a *framework* Kanban é porventura mais útil do que a Scrum

2. Continuous Integration and Delivery-CI/CD

Continuous Delivery é o conjunto de processos, práticas e ferramentas que possibilitam com regularidade adaptar o software às solicitações dos clientes, mudanças bruscas no mercado ou na estratégia, ou corrigir erros, sem qualquer impacto na operativa da empresa;

Construir um contexto de Continuous Delivery exige profissionais com as competências adequadas, mudança de processos, práticas e arquiteturas e um elevado nível de automatização. É necessária uma grande disciplina nas tarefas de engenharia para garantir a automatização completa do pipeline, do desenvolvimento à operação.

Os mecanismos de gestão financeira e de pessoal, terão de ser alterados. É diferente dizer que um projecto está concluído, quando este é constituído por elementos em que cada um pode ser alterado e sofrer *updates* a qualquer momento, de forma independente.

A adoção do AGILE e do DevOps é insubstituível num contexto CI/CD.

3. Lançamento de um processo Agile

Nenhum processo Agile deve ser lançado, antes de se estar preparado. A saber:

- Focado numa oportunidade de negócio com uma visão consistente;
- Com *outcomes* claros;
- Com recursos suficientes, formados e envolvidos com os valores Agile;
- Com delegação para operar com autonomia;
- Comprometida com os clientes;
- Capaz de criar rapidamente um MVP com ciclos curtos de desenvolvimento;
- Apoiada de forma efectiva pela gestão;
- Com um Product Owner com capacidade de liderança;
- E um orçamento com alguma flexibilidade.

Atenção à Arquitectura de Sistemas e ao Talento.

Atenção às expectativas exageradas nos primeiros ciclos (sprints), pois a experiência é determinante para a eficiência da equipa.

3. Como escalar Agile!

Escalar no uso do Agile, coloca questões pertinentes que terão de ser resolvidas antes!

Se a empresa lançar dezenas ou centenas ou até milhares de equipas Agile na organização, o que acontece?

Será que o Agile se aplica em todas as áreas da empresa ou apenas na produção de software?

E a metodologia não tem também os seus aspectos negativos?

Um estudo apresentado na HBR de Junho 2018, afirma ser possível escalar de forma efectiva e que ao fazê-lo as empresas criam benefícios substanciais. Mas também que os líderes têm de ser realistas (ver casos no 8.)

Há áreas onde o Agile pode não ser o mais indicado, mas todas terão de sofrer alterações agilizando os processos de trabalho para acompanharem as partes mais Ágeis!

Para as áreas com tarefas mais rotineiras, existem metodologias porventura mais adequadas, nomeadamente o Lean!

3. Como escalar Agile!

Desenvolver
fluencia no AGILE

Aumentar a
maturidade AGILE

Automatizar
infraestrutura e
DevOps

Aumentar a
cadencia de
entrega

- Nas área onde por qualquer razão não é seguida a metodologia AGILE e a organização é mais tradicional, todas as áreas tem de assumir os valores do AGILE e colaborarem de forma mais efectiva.
- A abordagem a duas velocidades gera frustração e distanciamento
- Transições do tipo big - bang foram ensaiadas por várias empresas, mas são difíceis. A generalidade das empresas desenvolve abordagens de acordo com uma plano que toma em consideração as prioridades, com um quadro de referência temporal para toda a empresa, bem estabelecido.

3. Como escalar Agile!

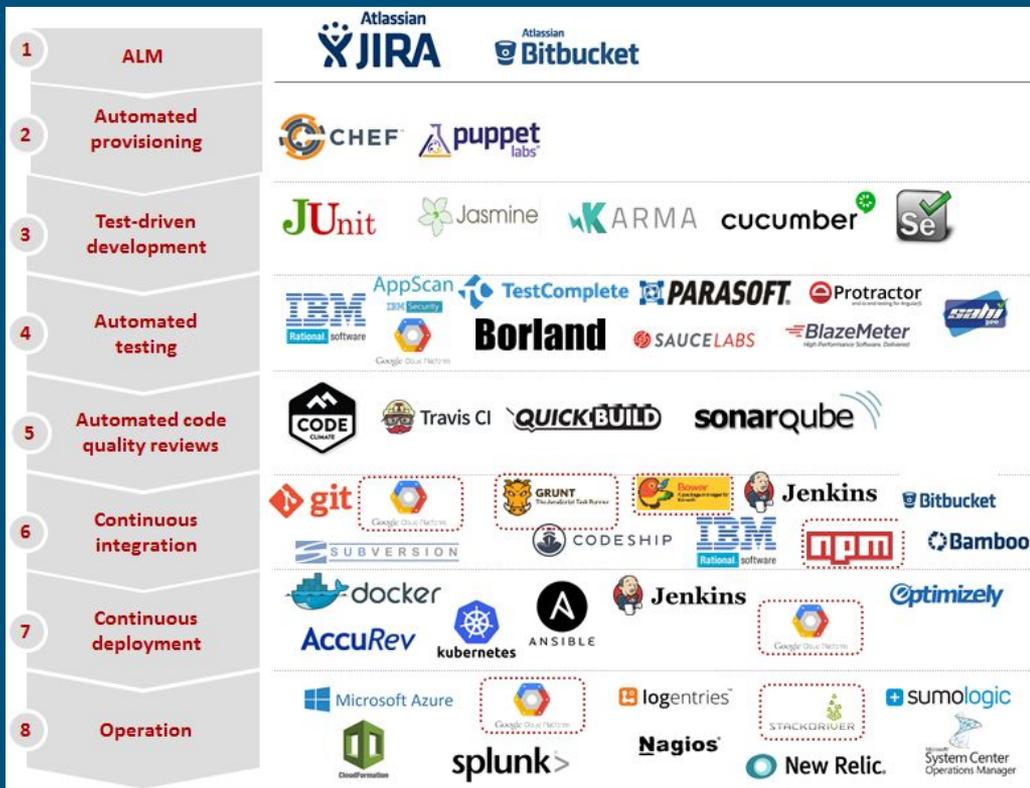
- Trata-se na sua essência de um processo de Gestão da Mudança!
- Importância especial têm a identificação das áreas piloto e as novas rotinas a estabelecer!
Ex :O caso EDP - Criação de 10 polos aplicativos com localização das equipas , só por si melhorou 60% a eficiência;
- A Gestão Financeira e dos RH têm de ser alteradas para refletirem os ciclos curtos das equipas operacionais.
- Figuras centrais do processo são os PRODUCT OWNER , os AGILE COACH e os SCRUM MASTER

RISCO PRINCIPAL - A adoção da metodologia Agile por líderes que não compreendem o essencial e que tentam, como sempre fizeram, escalar a partir de abordagens top-down, irá conduzir ao fracasso.

3. A importância das ferramentas suporte

Toda a prática deverá ser tão automatizada quanto possível.

As empresas que não automatizarem o seu IT, ficarão para trás quando tiverem de concorrer com startups mais ágeis.



4. O efeito do “tech debt”

“No core de muitos produtos tecnológico estão um conjunto de decisões históricas tomadas em sede de desenvolvimento, otimização e resolução de problemas, que frequentemente são desvalorizados em função de objectivos de curto prazo da empresa”

Ao contrário, numa startup a crescer rapidamente em termos de vendas, lucro e base de clientes, o desenvolvendo tem de ser rápido. O futuro pode até nunca existir!

Numa empresa de sistemas legados, não cuidar de seu “tech debt” é comprometer a operação futura, aumentar a complexidade e sem qualquer surpresa, os custos!

É neste contexto que o DevOps ganha uma importância crucial. Complementando o Agile, a fiabilidade do DevOps assegura um baixo défice tecnológico.

As empresas nativamente digitais com sucesso, consideram o tech. debt uma questão central. O *Zero Downtime* já é perseguido por organizações como a Spotify.

4. Formas de minimizar o tech debt

São regras insubstituíveis:

- Integrar na mesma equipa o Dev e o Ops, facilitando a colaboração entre quem desenvolve e testa e quem opera os sistemas;
- Enfatizar as boas práticas de programação; uso de ferramenta de ALM - Application Lifecycle Management;
- Automatizar todas as tarefas que seja possível e em particular as fases que acrescentem menos valor;
- Testar ao mesmo tempo que se faz o desenvolvimento da forma mais automática possível;
- Não acelerar a fase de requisitos de todos os desenvolvimentos que sejam muito interativos com outras aplicações; é o ponto de falha mais frequente;
- Fazer o desacoplamento dos sistemas digitais, dos sistemas legados (API's; Data Lake, etc)

5. Casos

Spotify

Bosch

USAA

SaaB

ING

Vattenfall

Entre muitos outros que facilmente se encontram hoje na internet

5. Recomendações

- Nos Produtos nativamente digitais usar sempre uma *framework* AGILE - SCRUM.
- Nos sistemas legados, usar as práticas e as rotinas que nos aproximem do AGILE. Ter objetivos claros sobre onde todos estarão no médio prazo, sob pena de facilitar rupturas culturais e criar fronteiras que reduzem os efeitos do AGILE;
- No AGILE, ser flexível na definição dos métodos, ajustando-os à empresa, e inflexível na sua operacionalização e qualificação dos Recursos Humanos.
- Nas áreas de TI, associar sempre a operacionalização das metodologias AGILE à operacionalização de uma prática de DevOps, como forma de mitigar o Tech. Debt.
- A liderança é sempre das áreas funcionais (Negócios), mas o IT é o guardião do Método, da Arquitectura e da Segurança. Não há oposição, há especialização e complementaridade.
- A equipa que desenvolve também testa, opera e faz evoluir a aplicação;
- Com subcontratação, terá de existir ajustamento do modelo de sourcing ao AGILE

6. Referencias

- HBR - Agile at Scale, Junho 2018 by Darrell K. Rigby, Jeff Sutherland, Andy Noble
- MIT Sloan Man. Review - Technical Debt might be hindering your digital transformation, June 2018
- SearchSoftwareQuality.techtarget.com
 - Comparing development methodologies, 2018 by Christopher Ward.
 - Five ways to reduce technical debt, 2018 by Jan Stafford
 - Comparing Methodologies: Agile vs. DevOps vs. Lean startup - e guide
- Gartner
 - Become an Agile Superhero, July 2018 by Bill Holz, Kirk Knoernschild
 - Life after scrum: development methodologies for DevOps
 - Solution path for achieving continuous delivery with Agile and DevOps, Nov 2017, Bill Holz
 - Market guide for Agile and DevOps Services, Oct 2017
- Computerweekly.com
 - Is DevOps the best approach to software development?, Rob Bamforth
 - Adapting to the rise of the holistic application - Jun 2018, Adrian Bridgwater
- How to enable quality at speed in 5 steps, Parasoft, February 2018 by Mark Lambert
- Speed vs Quality- which is preferred? - SmartBear, October 2017 by Alex McPeak
- The Agile Cultural Shift: Why Agile isn't always agile, CGI 2018