

Count Down

Mensagens enviadas aos inscritos no Q-Day conference 2018

Quando faltavam 5 dias...



A conferência QDay 2018

A [conferência QDay2018](#) vai realizar-se já no próximo dia 20 de setembro, na Culturgest, em Lisboa.

Esta é, talvez, a mais tecnológica das conferências QDay, de entre todas as já anualmente organizadas pela Quidgest, desde 2009.

Na sua décima edição, o QDay 2018 tem por tema quatro domínios que se entrelaçam: Hyper Agile, Lean IT, DevOps e Machine Learning.

Todos prosseguem a demanda de um sonho de há muitos anos: que as tecnologias da informação cumpram todo o seu potencial e contribuam para estruturas produtivas mais eficazes e para um mundo mais desenvolvido e mais sustentável.

Estes quatro domínios têm, em comum,

- maior automação,
- ciclos de desenvolvimento mais curtos,
- contribuições mais frequentes do software para as operações
- e alinhamento com os objetivos e com o ritmo dos negócios.

Na realidade, ajustam-se bem à definição clássica de Qualidade: adequação ao uso e melhoria contínua.

Deste modo, mesmo orientada para conceitos que poderão ser considerados mais técnicos, a conferência QDay 2018 não deixa de ser, à semelhança das anteriores, uma ponte entre gestão e decisão, por um lado, e tecnologia e investigação, por outro. Uma ponte, aliás, obrigatoriamente atravessada nos dois sentidos.

No QDay 2018, queremos saber onde estão os líderes da Transformação Digital que conseguem ser os primeiros a colocar a tecnologia ao serviço da gestão, de forma ágil, inteligente e evolutiva, com qualidade, à velocidade que o mercado, a lei, o imprevisto e a regulação exigem. Na gestão pública? no setor financeiro? nas grandes empresas? na saúde? na indústria? nas tecnologias da informação?

Ao mesmo tempo, pretendemos dar a conhecer aquilo que nos define e nos torna diferentes. O que nos qualifica, a nós e aos clientes com quem co-inovamos. Não

somos apenas “mais uma” empresa na área das tecnologias de informação. Apresentamos um larguíssimo conjunto de soluções inovadoras¹, que vão evoluindo, constantemente, de modo a estarem alinhadas com as principais tendências da melhoria contínua.

Sabemos que não podemos - e não queremos – traçar este caminho sozinhos, por isso, procuramos projetos e parceiros que, como nós, consigam ver para além do óbvio, que saibam concretizar e tornar visível mesmo o que não é evidente.

Para além do continuado suporte da Caixa Geral de Depósitos² e da Culturgest, esta décima edição volta a contar com o inestimável apoio de associações e de parceiros da Quidgest³. Pela primeira vez, tivemos também o contributo de um conselho consultivo, composto por prestigiados líderes da inovação, da tecnologia e do empreendedorismo. A todos, o nosso muito obrigado.

A dimensão estratégica das tecnologias da informação nas organizações

Hyper Agile, Lean IT, DevOps e Machine Learning são tendências e movimentos que combatem as práticas irracionais, ainda predominantes na utilização de tecnologias de informação na gestão das organizações.

Com efeito, a relação dos gestores e dos decisores com o software continua a ser estranha e ilógica. O software de gestão empresarial é o último setor económico em que o produtor é mais importante do que o cliente.

Acusando os clientes mais exigentes e mais conhecedores de “excessiva” customização, os fornecedores de software tradicionais obrigam as empresas a adaptar-se aos seus sistemas em vez de, como seria natural, adaptarem os seus sistemas às empresas. Do mesmo modo, são os produtores tradicionais de software a definir os ritmos de evolução, em lugar de acompanharem a evolução e expansão das empresas, das tecnologias e dos mercados.

Num mundo global, em que tudo se acaba por saber, são cada vez mais as evidências de maus resultados de entidades que prescindem das suas práticas, maduras e aperfeiçoadas ao longo dos anos, para se sujeitarem a supostas “boas práticas” que lhes são vendidas por vendedores de software pouco escrupulosos.

Porém, estas situações persistem e multiplicam-se. Gestores e decisores mostram grande azáfama na procura da última *buzzword* sobre tecnologias de informação, mas revelam uma enorme inércia na altura de tomar decisões.

¹ Com mais de 600 soluções diferentes no seu portfólio, em 12 áreas de especialização, a Quidgest já é, provavelmente, o maior produtor europeu de *software*

² PME Líder e PME Excelência, a Quidgest foi também distinguida pela CGD em 2018 com o novo estatuto CAIXA TOP, pelo seu desempenho, dinamismo e competitividade.

³ [AERLIS](#), [AFIP](#), [AIP](#), [APDSI](#), [APGICO](#), [APOGEP](#), [APQ](#), [Beta-i](#), [Bright Partners](#), [Casa da América Latina](#), [Coimbra Business School ISCAC](#), [COTEC Portugal](#), [ISCTE](#), [Saphety](#), [Universidade Nova de Lisboa](#), [WERInteraction](#), para além do [Jornal Económico](#) como media partner

O foco do QDay nestes quatro domínios justifica-se porque, sendo inequivocamente este o futuro, não é neste futuro que a maior parte das organizações está a investir.

A conferência QDay é, por isso, um convite a repensar e reformular a estratégia das organizações, à luz dos avanços mais significativos das tecnologias da informação. Que não são apenas uma componente “como qualquer outra” das organizações. As tecnologias de informação são o fator decisivo de sucesso de um projeto, de um serviço, de uma empresa ou de um país.

Essencial, nesta estratégia, é o posicionamento dos agentes económicos e dos países. Se estes se posicionam como seguidores tardios ou consumidores passivos de tecnologia, têm seguramente resultados muitíssimo (e, por vezes, fatalmente) piores do que se se posicionam como *early-adopters* e inovadores.

Em todos os setores, os agentes económicos estão perante a ameaça de uma nova concorrência, que já nasce digital. E nascer digital significa, simultaneamente, ser mais ágil, decidir mais rapidamente, ter menos custos, ser mais eficiente, explorar mais facilmente novos mercados, em suma, ser mais competitivo em todos os pontos relevantes.

Escolha rapidamente, e bem, o seu posicionamento face à revolução tecnológica do nosso tempo. O tempo urge.

Quando faltavam 4 dias...



O Hyper Agile na conferência QDay 2018

O manifesto para o desenvolvimento ágil de software¹, em 2001, constituiu um marco no modo de produção de sistemas de informação.

Refletindo a crescente preocupação pelo mau serviço prestado pelo software aos seus clientes, algumas das vozes mais conscientes das tecnologias da informação formularam, em quatro linhas, proposições que invertiam tudo o que era anteriormente considerado irrefutável, e declararam preferir escolher

- Pessoas e interações em vez de processos e ferramentas
- Software funcional em vez de documentação detalhada
- Colaboração com o cliente em vez de negociação de contratos
- Resposta à mudança em vez de cumprimento de um plano.

Um conjunto de princípios adicionais elucidavam claramente as consequências desta perspectiva.

Quase duas décadas passadas, o balanço é misto. Há muitas boas experiências, mas o desenvolvimento ágil não é, ainda, generalizadamente aplicado. O processo iterativo ainda não é mais usado do que a alternativa, o clássico *waterfall*. “Já somos Ágil” significa, muitas vezes, ter um pequeno projeto Ágil, enquanto a maioria dos projetos segue a decomposição linear e sequencial do *waterfall*: requisitos > concepção > desenvolvimento > testes > instalação. É também comum, quando algo começa a correr menos bem num projeto, que as equipas (e os clientes) se refugiem em práticas *waterfall*. Se as promessas do Agile não se cumprem, a tendência é procurar abrigo nas velhas técnicas.

No entanto estas estão claramente ultrapassadas, devido à rápida mudança de paradigmas. Nos dias de hoje, chegamos a ver casos em que o software antes de ser terminado já se encontra obsoleto e já se sabe que não irá responder às necessidades do cliente. A realidade e as necessidades alteram-se hoje em dia mais rapidamente do que o desenvolvimento do software, quando este é realizado pela metodologia *waterfall*.

Por outro lado, o desenvolvimento ágil transformou-se na metodologia Agile e, de algum modo burocratizou-se. Perde-se frequentemente a noção do objetivo, passando

¹ <http://agilemanifesto.org/>

o meio a ser identificado como o objetivo final². Alistair Cockburn, um dos subscritores do manifesto, queixava-se, no ScrumDay realizado em Lisboa³ em 2017, deste efeito. A certificação (de dois dias) de *scrum masters* e a identificação de *product-owners* não garante, de forma alguma, o sucesso dos projetos.

O sucesso do software não é apenas uma questão de metodologia. Seguir uma metodologia Agile, como o recente caso Lidl demonstra⁴, pode também conduzir a um desastre. E pode até ser mais difícil e demorado (e, conseqüentemente, mais caro) declarar um projeto ágil como falhado. É essencial que essa agilidade seja já uma característica técnica do software utilizado para desenvolvimento, o que, na Lidl, não era evidentemente o caso.

O software inspira outras áreas da gestão e devemos seguir com toda a atenção o projeto do BBVA de alterar a sua organização, a sua estratégia e (mais difícil ainda) a sua cultura com o objetivo “More than 30,000 people working “agile” in 2018”⁵.

A contribuição da Quidgest

Quase duas décadas depois, há que exigir mais ao Agile.

O desenvolvimento de software não se tornará efetivamente mais ágil (mais rápido) enquanto a escrita de código for realizada por humanos a uma taxa de escrita de 2 caracteres por segundo e, ainda assim, sujeita a naturais erros humanos, por vezes bastante difíceis de detetar, mesmo com a prática de *peer-reviewing*.

Também não se torna mais ágil ou mais rápido se se aplica a uma configuração de um *software standard* tradicional. Estas configurações são manietadas pela componente *standard* (uma *black box* intocável), que tem uma agilidade semelhante ao betão. Neste caso, falar de conciliar Agile e um ERP *standard* tradicional, por exemplo, é apenas enganar o cliente⁶.

Os ganhos obtidos por apenas se proceder a uma reorganização das equipas e das tarefas (isto é, o Agile não suportado por automação da geração de código) são limitados e estão confinados a projetos de pequena dimensão e pouco complexos, de curta duração e desenvolvidos por equipas relativamente diminutas.

A realidade é que o Agile é limitado⁷, e de modo a suprir essas fragilidades e aplicando as lições aprendidas onde foi não tão bem-sucedido como se desejaria, surgiu o Scaled

² Uma posição interessante <https://www.lambdacambridge.com/blog/2018-05-how-scrum-destroyed-agile>

³ Em 2018, vai organizar-se o Experience Agile <https://www.experienceagile.org/>

⁴ <https://www.consultancy.uk/news/18243/lidl-cancels-sap-introduction-having-sunk-500-million-into-it> notando-se que a metodologia proposta se pretendia ágil: <https://www.kps.com/en.kps-transformation.html/article-2203-unternehmen-der-unterschied>

⁵ <https://www.bbva.com/en/bbvas-journey-becoming-agile-organization/>

⁶ <http://www.scmfocus.com/sap/2018/08/kps-continues-to-keep-promote-hana-for-retail-for-lidl-after-failure/> A este respeito, aconselhamos vivamente a opinião fundamentada e corajosa de [Shaun Snapp](#).

⁷ Por exemplo, as limitações da aplicação do Scrum estavam bem patentes no primeiro artigo sobre o tema, “[The New New Product Development Game](#)”, de Hirotaka Takeuchi e de Ikujiro Nonaka, Harvard Business Review, January 1986, muito antes de o termo ser aproveitado pela produção ágil de software.

Agile ⁸(agora na versão SAFe 4.5). Esta metodologia incorpora um conjunto de preocupações relacionadas com níveis mais elevados de complexidade e com carteiras simultâneas de projetos, ao mesmo tempo que integra os conceitos de Lean IT.

A Quidgest é uma orgulhosa defensora do Agile e do Scaled Agile, mas tem contribuído para elevar estas metodologias a outro patamar de eficácia.

As contribuições decisivas da Quidgest para o Hyper Agile são

- a automação: ao saltar de 2 para 1 000 000 de caracteres escritos por segundo, e sem erros;
- a modelação: o que significa um controlo a um nível mais elevado de abstração, uma representação de alto nível e uma explicitação da arquitetura do sistema de informação;
- a eficiência das equipas: com as competências integradas no Genio as equipas são mais pequenas, há um muito elevado reaproveitamento de padrões e muito menos trabalho direto para cada projeto;
- a redução dos ciclos de produção de software: para apenas algumas horas;
- a comunicação e a interação com o negócio: é fácil comunicar e refletir ao nível do modelo, tarefa que é impossível ao nível do código, uma vez que as linguagens de programação (quaisquer que sejam) não são a forma adequada para conceber soluções de software ou para falar com CEO ou utilizadores-chave.

⁸ <https://www.scaledagileframework.com/>

quando faltavam 3 dias...



O Lean IT na conferência QDay 2018

Fruto da muito maior procura do que oferta, o desenvolvimento de software de gestão empresarial beneficia de um estatuto particular. Não existiu pressão para a mudança que fosse suficiente para impor, neste setor, uma cultura da Qualidade Total, como se impôs, há mais de 50 anos, na produção industrial.

O *Scrum* é um conceito proveniente de ambientes industriais *Lean*¹. Porém, a maior parte dos conceitos que provaram a sua eficácia na produção industrial (nomeadamente, no sistema Toyota de produção) não transitaram, nem foram acolhidos pela indústria do software.

E, claramente, deveriam ter sido considerados.

Aqueles que a produção de software deveria adotar, e não adotou, são demasiados para estas notas, mas deixamos aqui alguns conceitos para inspiração.

Gemba ou ir ao local. A mais grave de todas as faltas da qualidade, na produção tradicional de software, é o processo de verificação de erros. Na produção de software tradicional, a deteção de erros (bugs) é feita por testers² e no final do desenvolvimento. O que viola grosseiramente o terceiro dos 14 Pontos para a Gestão da Qualidade Total de Edward Deming³. É certo que controlo de qualidade só no final é melhor do que nenhum controlo da qualidade⁴. Mas a dependência de inspetores é fortemente desaconselhada. Os erros são detetados fora do contexto em que foram produzidos e, não se estando no local, é muito mais difícil perceber a sua causa e atuar sobre ela. É acrescentado trabalho (e equipamentos) a fases posteriores da produção, quando se podia ter detetado o problema logo quando ele foi introduzido no software. A responsabilidade é diluída e não se cria a cultura de a Qualidade ser uma tarefa de todos os que participam na produção de software.

¹ [The New New Product Development Game](#), Hirotaka Takeuchi e Ikujiro Nonaka, Harvard Business Review, January 1986

² Existe uma dinâmica Associação Portuguesa de Testes de Software <https://www.pstqb.pt/>. Do mesmo modo, há, nas comunidades [meetup](#), inúmeros grupos focados em testes de software, espalhados por todo o mundo. E muitas ofertas de emprego como Test Engineers, mesmo em empresas consideradas Agile <https://jobs.jobvite.com/farfetch/job/oCSF3fwW>

³ <http://asq.org/learn-about-quality/total-quality-management/overview/deming-points.html>

⁴ Joel Spolsky, uma voz reconhecida no desenvolvimento de software, representa bem a linha de raciocínio que prevaleceu, durante anos neste setor: <https://www.joelonsoftware.com/2000/04/30/top-five-wrong-reasons-you-dont-have-testers/>

Kaizen ou melhoria contínua. A fragilidade da produção de software tradicional, dependente de programadores e das suas habilidades específicas e, até, humores, reflete-se na forma como o software se vai construindo. O software tradicional não resulta de uma linha de produção que se vai afinando. Cada programador escreve uma parte do código, junta-o ao de outros programadores e dá a sua tarefa por concluída. Uma vez acabado o software, entrega-se. Não se continua a melhorar. Só será melhorado, se posteriormente se encontrar algum defeito. Se cumprir os mínimos (funcionar) não há nenhum esforço para o tornar mais eficiente, mais robusto ou mais adequado à utilização.

Jidoka ou automação inteligente. A produção de software tradicional não é automatizada. Mas, na Quidgest, a máquina (o computador) deteta automaticamente eventuais erros e a intervenção humana tem força para parar o processo de produção, reparar o problema e encontrar a causa, de forma a eliminar a hipótese de ocorrências posteriores. A produção não é retomada enquanto este ciclo não for executado.

Poka yoke, isto é, “à prova de erro” ou “prevenindo de erros inadvertidos”. É um conceito bem ilustrado e compreendido pela forma de uma bateria, de um cartão de memória ou de um microchip, forma essa que impede que estes objetos sejam colocados incorretamente.

De notar que, tal como as restantes dimensões debatidas na conferência Q-Day 2018, o Lean IT aplica-se a organizações de produção de software autónomas e descentralizadas e a redes de equipas colaborativas. Requer uma cultura própria, partilhada por todos os que participam na produção de valor, o que conduz novamente ao *Gemba*.

A contribuição da Quidgest

Na Quidgest, o software resulta de uma linha de produção automatizada, o Genio. A produção de software adota, assim e muito mais facilmente, a lógica de uma indústria. A identificação de padrões permite a utilização de máquinas em tarefas antes apenas desempenhadas por humanos. A sua reutilização cria uma massa crítica de casos e de contextos de aplicação que convidam a um aperfeiçoamento continuado.

O Genio, como uma linha de produção industrial automatizada, tem uma componente de modelação / conceção e uma componente de criação de código.

O modelo é um espaço adicional, que abre oportunidades excelentes para a implementação da Qualidade Total. Partindo dos conceitos *Jidoka*, *Poka-yoke* e *Kaizen*, a Quidgest integrou no Genio formas inovadoras de garantir a Qualidade não apenas dos produtos que fabrica, mas do processo de produção.

O já anteriormente referido Scaled Agile / SAFe ⁵ é uma corrente que agrega Agile e Lean IT e é, provavelmente, o movimento a que devemos estar mais atentos. Junte-se-

⁵ <https://www.scaledagileframework.com/>

Ihe a modelação e geração automática de software e temos o ambiente de qualidade existente na Quidgest, que consideramos ideal.

Mas a Quidgest é, ainda, uma exceção. Não conhecemos nenhuma outra empresa de software, presente no mercado global (isto é, excluindo investigações de âmbito mais reduzido ao nível académico), em que o modelo defina inequivocamente a totalidade do sistema de informação. E, por isso, em que seja possível extrair qualquer sistema de informação (nomeadamente os mais complexos ERP) de um processo industrializado.

quando faltavam 2 dias...



O DevOps na conferência QDay 2018

A fusão do Desenvolvimento (Dev) com as Operações (Ops), o DevOps, é o seguimento natural do Agile.

O primeiro princípio do manifesto para o desenvolvimento ágil de software¹ já deixava claro que *“our highest priority is to satisfy the customer through early and continuous delivery of valuable software”*: a maior prioridade é a entrega rápida e contínua de software valioso para o negócio.

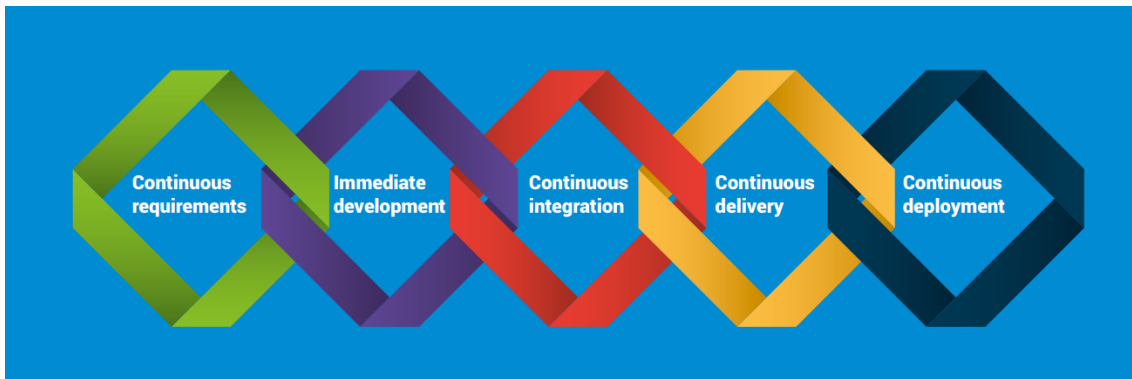
Esta parece uma proposição neutra e universalmente válida, até nos apercebermos de quão distante está das práticas correntes das organizações.

A relação entre as tecnologias de informação (o IT) e as operações (o negócio) nunca foi pacífica. Representado por departamentos de sistemas de informação internos ou por fornecedores externos, o IT tradicional (pré DevOps) tem assumido posições pouco flexíveis, demorando muito tempo a dar resposta às aspirações das áreas que utilizam as suas soluções. E as áreas operacionais tendem a não transferir o seu conhecimento de processos e práticas para o IT, sendo certo que, a maior parte das vezes, o IT tradicional não aprecia e reage negativamente a estas sugestões, que considera “excessiva” customização².

No fluxo contínuo de criação de valor para o cliente, é necessária uma muito melhor colaboração entre as múltiplas funções envolvidas. Isto é, é necessária uma reengenharia de processos que inclua também o IT.

O DevOps advoga a constituição de equipas conjuntas e diretas, solidariamente responsáveis, sem intervenção de terceiros.

Os terceiros habitualmente convocados (o Quality Assurance³, uma coordenação, uma fiscalização de projeto) para criar uma ponte entre Dev e Ops, na prática destroem a nova relação de confiança que a cultura⁴ DevOps pretende instituir entre o desenvolvimento e as operações. E destroem, inevitavelmente, porque a sua única razão de existência é o pressuposto de essa confiança estar ausente⁵.



Na cultura DevOps, a organização do desenvolvimento de software já não é por projeto, mas como um serviço contínuo que

- 1) deteta (e até antecipa) novas necessidades do cliente e modela os novos requisitos,
- 2) converte estes requisitos, rápida e imediatamente, em código,
- 3) integra, testa e garante a coerência de todas as componentes e do trabalho desenvolvido em paralelo por várias equipas,
- 4) entrega ou disponibiliza a nova versão ao cliente e
- 5) (a pedido ou após validação) coloca a nova versão em utilização produtiva pelo negócio / operações (Ops).

A comunidade DevOps, bastante forte, organiza todos os anos o [All Day DevOps](#), um evento que, seguramente, vale a pena acompanhar. E, enquanto não chega o de 2018, podemos assistir às 100 sessões de 2017⁶.

O relatório da Puppet sobre o estado atual do DevOps ([2018 State of DevOps Report](#))⁷ é também uma referência útil.

A contribuição da Quidgest

O Genio é um fator decisivo na garantia de automatização de todo este processo.

Sem o Genio, o DevOps sofre de uma de três limitações

- apenas começa na integração contínua e não inclui a fase de exploração de novos requisitos,
- não tem forma de modelar esses novos requisitos
- entre a exploração / modelação e a integração contínua exige um esforço considerável de programação manual, sendo um facto que a programação manual, muitíssimo lenta, estrangula e deteriora a fluidez do processo.

O fluxo do DevOps pode ser mais lento ou pode ser mais rápido. Mas a rapidez de todo o processo integrado é, naturalmente, o objetivo a atingir e a vantagem competitiva a alcançar. De pouco vale dizer-se que se adota esta perspetiva se os tempos de disponibilização de versões melhoradas pelo desenvolvimento (Dev) às operações (Ops) permanecem muito longos.

O Genio da Quidgest contribui decisivamente para o sucesso desta perspetiva, ao interligar todo o processo:

- 1) modela os requisitos,
- 2) programa de forma automática e extremamente rápida o código,
- 3) integra o trabalho de várias equipas (através do GenioSVN),
- 4) prepara, testa e entrega o sistema (através de ferramentas open-source como Jenkins, a que a Quidgest pode recorrer, ao contrário de outras plataformas, por usar exclusivamente linguagens de programação padrão),
- 5) assegura a evolução automática da estrutura de dados e documenta, informa e prepara os utilizadores finais, de modo a garantir a utilização produtiva da nova versão da solução.

O DevOps estabelece a existência de *architecturally significant requirements*. Na Quidgest, esses requisitos foram transformados em padrões, constituindo o conjunto de padrões relacionados com a engenharia do software.

O processo de integração requer uma ferramenta de controlo de versões (na Quidgest, o GenioSVN), com todas as funcionalidades associadas a repositórios, *checkouts*, *baselines*, *updates*, *commits*, gestão de conflitos, lista de mudanças, *branches* ou *forks*, revisão pelos pares e responsabilização (*blame*).

O Genio suporta um outro aspeto muito importante, que é a evolução em paralelo de duas camadas do modelo: o modelo de negócio (estrutura de dados, regras de negócio, interfaces e processos) e a tecnologia (arquiteturas, linguagens de programação, bibliotecas, padrões). Estas duas camadas garantem a continuidade e a integração de todo o processo.

Num processo de integração contínua, a evolução é também contínua. Tal garante 1) vidas úteis das soluções muito mais prolongadas e 2) inexistência de disrupções pela mudança de versões do software.

Os nossos clientes e parceiros na co-inovação, nomeadamente os que este ano premiamos na [conferência QDay](#), são um bom exemplo de aplicação prática de DevOps a grandes sistemas de gestão.

¹ <http://agilemanifesto.org/principles.html>

² A revista [QuidNews nº 10](#) ilustrava, com uma pequena história “O aleijadinho e o alfaiate”, a páginas 25, o que a Quidgest pensa deste conceito de excessiva customização.

³ O diagrama que, a esta data, ainda figura na página wiki correspondente ao DevOps <https://en.wikipedia.org/wiki/DevOps#/media/File:Devops.svg> ainda inclui o QA na equação, o que não nos parece fazer qualquer sentido

⁴ O DevOps é uma cultura

⁵ Uma posição muito próxima é defendida por Asaf Yigal <https://devops.com/devops-killed-qa/>

⁶ Disponíveis em <https://www.alldaydevops.com/2017-all-day-devops-video-recordings> Por exemplo “[It's Not Continuous Delivery If You Can't Deploy Right Now](#)”, por Ken Mugrage

QDay Countdown no LinkedIn

QDay -5 Countdown

<https://www.linkedin.com/pulse/countdown-qday-5-jo%C3%A3o-paulo-carvalho/>

QDay -4 O Hyper Agile na conferência QDay 2018

<https://www.linkedin.com/pulse/o-hyper-agile-na-confer%C3%Aancia-qday-2018-jo%C3%A3o-paulo-carvalho/>

QDay -3 O Lean IT na conferência QDay 2018

<https://www.linkedin.com/pulse/o-lean-na-confer%C3%Aancia-qday-2018-jo%C3%A3o-paulo-carvalho/>

QDay -2 O DevOps na conferência QDay 2018

<https://www.linkedin.com/pulse/o-devops-na-confer%C3%Aancia-qday-2018-jo%C3%A3o-paulo-carvalho/>