

# SCRUM

## Um Modelo Ágil para Gestão de Projectos de Software

Décio Ferreira<sup>1</sup>, Felipe Costa<sup>2</sup>, Filipe Alonso<sup>3</sup>, Pedro Alves<sup>4</sup>, and Tiago Nunes<sup>4</sup>

<sup>1</sup> ei03037@fe.up.pt

<sup>2</sup> ei03041@fe.up.pt

<sup>3</sup> ei03044@fe.up.pt

<sup>4</sup> ei03083@fe.up.pt

<sup>5</sup> ei03101@fe.up.pt

**Resumo** Scrum é um modelo ágil gestão de projectos de software. Baseia-se no desenvolvimento incremental das aplicações, centrado na equipa, permitindo um maior controlo do processo pelo facto de ter ciclos de iteração muito curtos. Este documento pretende apresentar a metodologia, desde o seu nascimento à própria implementação.

### 1 O que é Scrum?

Antes respondermos a esta questão, devemos primeiro introduzir um outro tema: o Desenvolvimento Ágil de Software.

Este caracteriza-se como sendo uma forma "não tradicional" de desenvolver software, em que (e segundo o Manifesto for Agil Software Development) a prioridade máxima se centra na satisfação do cliente através da entrega antecipada e contínua de software susceptível de avaliação.

Ken Schwaber, co-fundador da metodologia Scrum, envolvido na formação da Agile Alliance e curiosamente também envolvido na definição de modelos tradicionais como o modelo "Queda de água", define esta metodologia recordando um momento algo divertido:

*Quando me perguntaram em que consistia o meu trabalho, respondi que ajudo pessoas a desenvolver software em 30 dias. E o sujeito olhou para mim e disse "Então, não tenho de esperar 180 dias para ter aquilo que não quero?" Sim, exactamente, nós damos-lhe aquilo que não quer em 30 dias.*

Assentando os seus princípios fundamentalmente em boas práticas de gestão, o Scrum assume-se como uma metodologia extremamente ágil e flexível. Tem por objectivo definir um processo iterativo e incremental de desenvolvimento de produtos ou gestão de projectos. Produz um conjunto de funcionalidades potencialmente mais próximas do objectivo final no terminar de cada iteração (normalmente com uma duração de 30 dias). Centrado no trabalho em equipa,

melhora a comunicação e maximiza a cooperação, permitindo que cada um faça o seu melhor e se sinta bem com o que faz, o que mais tarde se reflecte num aumento de produtividade.

Scrum aplica-se a projectos tanto pequenos como grandes. Esforçando-se para libertar o processo de quaisquer barreiras, o seu principal objectivo é conseguir uma avaliação correcta do ambiente em evolução, adaptando-se constantemente ao caos de interesses e necessidades.

Englobando processos de engenharia, este método não requer nem fornece qualquer técnica ou método específico para a fase de desenvolvimento de software. Scrum apenas estabelece conjuntos de regras e práticas de gestão que devem ser adoptadas para garantir o sucesso de um projecto.

## 2 Como nasceu o conceito?

A metodologia Scrum, desenvolvida por Ken Schwaber e Jeff Sutherland e inspirada nas ideias seminais de desenvolvimento rápido e concorrente de produtos de Hirotaka Takeuchi e Ikujiro Nonaka, nasceu da necessidade de encontrar uma metodologia que abordasse o problema do desenvolvimento de software de uma forma não tradicional, em que tal como num jogo de Rugby, a equipa age como um todo para atingir os seus objectivos ("scrum" denomina, no Rugby, uma equipa aglomerada, em que toda a gente age em conjunto para transportar a bola pelo campo).

Algumas datas importantes:

- 1993 - Ken Schwaber desenvolveu na ADM uma framework iterativa e incremental;
- 1993 - Metodologia Scrum implementada pela primeira vez por uma equipa liderada por Jeff Sutherland na *Easel Corporation*;
- 1994 - Sutherland definiu o Scrum em Easel;
- 1994 - Ken and Jeff refinam Scrum;
- 1996 - IDX usa Scrum em projectos com aproximadamente 600 pessoas;
- 1996 - Scrum é apresentado na OOPSLA (*international conference on Object Oriented Programming Systems, Languages and Applications*);
- 2000 - Práticas "eXtremme Programming" usadas conjuntamente com Scrum (XP @ Scrum);
- 2001 - Ken Schwaber e Mike Beedle editam o primeiro livro de Scrum;
- 2003 - Certificações Scrum;

## 3 Como se desenvolve software Scrum?

### 3.1 Vocabulário Específico

*Backlog* Lista de todas as funcionalidades a serem desenvolvidas durante o projecto completo, sendo bem definido e detalhado no início do trabalho, deve ser listado e ordenado por prioridade de execução.

*Sprint* Período não superior a 30 dias, onde o projecto (ou apenas algumas funcionalidades) é desenvolvido.

*Sprint Backlog* Trabalho a ser desenvolvido num Sprint de modo a criar um produto a apresentar ao cliente. Deve ser desenvolvido de forma incremental, relativa ao Backlog anterior (se existir).

*Scrum* Reunião diária onde são avaliados os progressos do projecto e as barreiras encontradas durante o desenvolvimento.

*Scrum Meeting Rules* Protocolo a seguir de modo a realizar uma reunião Scrum.

*Scrum Team* A equipa de desenvolvimento de um Sprint.

*Scrum Master* Elemento da equipa responsável pela gestão do projecto e liderar as Scrum Meetings, são normalmente engenheiros de software ou da área de sistemas. Apesar de ser gestor não tem propriamente autoridade sobre os demais membros da equipa. É incentivada a auto-gestão.

### 3.2 Regras do Scrum

De modo a produzir software correctamente, através do método ágil Scrum, é necessário respeitar algumas regras de execução, nomeadamente no que diz respeito ao Backlog, Sprint e Scrum Meeting.

Relativamente ao Backlog, deve ser debatido pela equipa todos os pontos que devem integrar a lista de funcionalidades da aplicação, sendo de responsabilidade de uma única pessoa (Scrum Master) a ordenação da lista por prioridade de execução.

Por sua vez, o Sprint deve ser realizado num período não superior a 30 dias e ter uma equipa de trabalho não superior a 9 elementos. Deve também ter um objectivo bem claro, baseado no Backlog. O Backlog não deve ser modificado durante a realização do Sprint, com excepção de novas funcionalidades que, segundo o Scrum Master, tenham influência fundamental no decorrer do projecto e que possam ser completadas dentro do período destinado ao Sprint. Se o Sprint

estiver a tomar um rumo não desejável, é possível dissolver o Sprint e começar um novo, baseando este num novo Sprint Backlog.

As Scrum meetings são um ponto de elevada importância no desenvolvimento do projecto. É nessas reuniões que o Scrum Master deve actualizar-se do decorrer do projecto e procurar identificar as principais barreiras ao desenvolvimento, podendo assim agir de uma forma eficaz na sua eliminação.

As reuniões durante um Sprint devem ser diárias, sempre a mesma hora e no mesmo local e não devem durar mais que 30 minutos.

Toda conversação é restringida às respostas dos elementos às perguntas colocadas pelo Scrum Master, sendo elas:

1. O que desenvolveu desde a última reunião?
2. Que dificuldades encontrou durante o seu trabalho?
3. O que planeia desenvolver até a próxima reunião?

Todos os elementos devem responder as perguntas e com base nas respostas o Scrum Master deve imediatamente tomar decisões (caso necessárias) por forma a remover todas as situações que impeçam o bom desenvolvimento do trabalho.

### 3.3 O Processo Scrum

Para iniciar o processo Scrum, a primeira coisa que é preciso definir é q equipa constituída pelas pessoas designadas para trabalhar e que irão compor a equipa Scrum. Esta equipa não deve ter mais de 6 a 9 membros. Se houver mais membros do que o possível gerir, separam-se várias equipas Scrum e cada equipa focar-se-á numa área específica do trabalho, envolvendo toda a equipa para trabalhar nesta área específica.

Quantos aos materiais que deverão ser fornecidos a cada equipa, certifique-se que tem sempre quadros de reunião disponíveis, marcadores para os quadros, postits, impressões do backlog do produto e cartões de referência diários para cada participante, para que todos tenham acesso a toda a informação e os meios para preparar e compreender cada parte do processo Scrum em qualquer altura.

A próxima coisa a fazer é apontar o Scrum Master, uma vez que é essa pessoa que conduz as Scrum Meetings, mede o progresso empiricamente, toma decisões e remove os obstáculos do caminho para não desacelerar ou mesmo parar o trabalho em pontos críticos. O Scrum Master fica encarregue de, como referido anteriormente, perguntar a todos os membros da equipa as três questões mencionadas.

É o Scrum Master que deve ser capaz de tomar decisões imediatas e resolver todos os impedimentos rapidamente, de modo a não estender o tempo da reunião. Compete-lhe identificar o backlog inicial, que é todo o trabalho proeminente para uma área do produto, tanto imediato e bem definido, como a longo prazo e indefinido.

Para identificar o backlog, a primeira coisa a fazer é listar todo o trabalho conhecido necessário fazer e agrupá-lo em incrementos que não devem ter duração superior a 30 dias. Se houver áreas de trabalho voláteis ou que não possam ser

completamente definidas para 30 dias, deve ser estabelecido um incremento para um tempo conhecido. Depois disto, é preciso listar todo o trabalho proeminente a fazer e definir prioridades para todos os elementos listados. Uma vez terminado, o backlog deve ser assinado pelos membros das equipas e a partir daí, só o backlog criado deverá ser cumprido durante este Sprint para cada área.

A partir deste ponto, para conduzir o trabalho colaborantemente e verificar o progresso dentro de cada equipa, obriga a uma auto-organização e disciplina, começar a distribuição de trabalhos e naturalmente, conduzir as reuniões Scrum diariamente. É vital para que o processo funcione cumprir com os trabalhos rigorosamente com base nos pontos restantes do Sprint Backlog.

Para isso, é preciso estabelecer e conduzir as reuniões diárias Scrum onde as equipas se encontram e se actualizam sobre o que se vai fazendo. Isto fornece um foco diário no trabalho em desenvolvimento. Antes de mais, certifique-se que as reuniões se realizam sempre à mesma hora e no mesmo local, evitando gastos na procura diária de um lugar, assim como evita as equipas terem que diariamente saber onde e quando será a reunião desse dia. Não esquecer que cada reunião não deve ultrapassar os 30 minutos. Durante este tempo o Scrum Master cumpre o seu papel em colocar as referidas questões e em resolver todas as decisões necessárias. Qualquer questão que não responda às três questões colocadas deverá ser adiada para posteriores reuniões.

No fim de cada Sprint, deve ser feita uma reunião para revisão e demonstração do Sprint. Para conduzir estas reuniões deve ser eleito um porta-voz para a guiar e conduzir de forma a obter algumas questões resolvidas e registar a retrospectiva do grupo do Sprint:

1. Qual o valor acrescentado neste incremento (Demonstração)?
2. O que foi completado do nosso Sprint Backlog?
3. Qual o feedback por parte do Cliente do produto?
4. O que se aconteceu de relevante no grupo durante o Sprint?
5. Como é que cada um se sentiu?
6. O que podemos concluir disso?
7. O que pode ser aplicado para melhorar o próximo processo Sprint?

Podem ser consideradas também alguns linhas guia facilitadoras como fornecer um blocos de notas a cada equipa para capturar o seu objective das reuniões de sprint backlog, revisão e demonstração e demais informações que suportem a sua demonstração, capturando os resultados em câmara digital para rever e confrontar se necessário, explicar perfeitamente as regras para que o processo corra melhor como que a equipa deve trabalhar em conjunto, toda a gente tem que trabalhar no Sprint e cada equipa deve demonstrar algo no fim de cada Sprint, não permitir apresentações em powerpoint nas reuniões uma vez que é uma perda de tempo tanto na sua concepção como na explicação em reuniões de tempo reduzido, focar que a equipa deve completar o seu planeamento do Sprint com um Sprint Backlog num quadro de reunião e que não há papeis predefinidos na equipa, uma vez que o objective é que sigam regras de auto-organização.

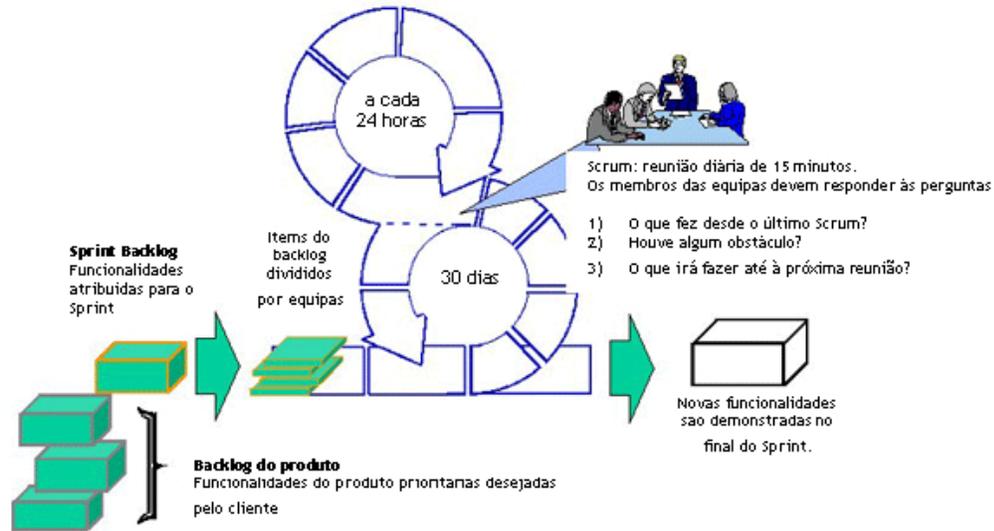


Figura 1. Descrição do processo scrum

## 4 Aplicações Práticas

Diz-nos Ken Schwaber (um dos criadores da metodologia) que algumas respostas comuns à pergunta "Porque decidiram implementar Scrum?" são "Não temos nada a perder" ou "Não pode ficar pior do que o que já está". Como já foi visto, os métodos tradicionais não lidam muito bem com o risco associado ao desenvolvimento do projecto, pelo que é comum haver organizações a procurar uma "salvação" quando se apercebem que estão no caminho do fracasso. Scrum consegue avaliar o ambiente regularmente e adaptar-se a ele, minimizando os riscos e produzindo a funcionalidade possível em determinado momento.

No entanto, apenas um terço das organizações que tentam implementar Scrum como modelo de desenvolvimento conseguem ter sucesso. Na verdade, as estruturas de desenvolvimento tradicionais impedem, de certo modo, uma transição fácil para métodos ágeis, devido aos hábitos que incutem. Por exemplo, enquanto que tradicionalmente existe um Gestor de Projecto que comanda e controla as equipas de desenvolvimento, Scrum baseia-se na capacidade de auto-gestão das equipas. Dá-se até outra designação ao Gestor de Projecto (Scrum Master), para de certo modo frisar o facto de este não ter autoridade nas equipas. Outro exemplo é o facto de os intervenientes no desenvolvimento terem, em Scrum, funções variadas (por forma a maximizar a produtividade da equipa), em vez de possuírem funções definidas e limitadoras. É exigido um esforço bastante grande no sentido de ultrapassar estes hábitos e conseguir uma viragem razoavelmente brusca em termos de metodologias. Como diz Schwaber, Scrum não é milagroso; é, sim, muito e árduo trabalho.

Ultrapassadas estas dificuldades (algo bem possível, empresas como a Philips, Cisco Systems, Siemens, Nokia, Microsoft e Sun Microsystems têm já experiência no uso deste método), importa a perseverança. Sendo que a cada Sprint se procura implementar um dado conjunto de requisitos, o normal para uma organização que começa a usar Scrum é falhar grande parte dos requisitos na primeira iteração. Razões várias determinam este facto: insegurança, não tão bom conhecimento da equipa, dificuldades tecnológicas... Após a segunda iteração, é normal observar já uma maior aproximação ao proposto e eventualmente, por já dominar o processo, a equipa consegue uma exactidão notável.

Segundo alguns estudos, durante os primeiros meses de implementação de Scrum as organizações conseguem cumprir 4/5 requisitos com um custo de 70,000 Euros, com uma média de 100 defeitos na produção. Um ano depois, conseguem já cumprir o dobro de requisitos pelo mesmo custo e número de defeitos. Ao segundo ano cumprem já 13/14 requisitos pelo mesmo preço mas já com uma média de 5 defeitos na produção. Estes números são indicadores da evolução do processo de implementação da nova metodologia. A preocupação inicial consiste em conseguir a relação desejada entre a equipa e o cliente e o entrosamento da própria equipa. Estando este processo cimentado tenta-se melhorar os aspectos de engenharia inerentes ao desenvolvimento, reduzindo o erro na produção.

É já reconhecida a capacidade dos processos ágeis de aumentar a produtividade. De facto, em termos de Pontos de Função (usados para "medir" uma aplicação em termos de funcionalidades) e enquanto que o normal, na indústria, é de 2 Pontos de Função por pessoa, por mês, um processo ágil (bastante optimizado) consegue alcançar os 77 Pontos de Função por pessoa, por mês. Apesar deste aumento brutal, a qualidade de vida dos trabalhadores não é de todo afectada pela negativa. Pelo contrário, o carácter colaborativo da filosofia de Scrum, em que se abolem completamente os cubículos de trabalho e as "ordens de cima" e se fomenta a resolução de problemas pela via da comunicação entre os elementos, faz com que se note um aumento significativo na moral das equipas. As pessoas chegam ao trabalho com grandes expectativas para o dia e geram um ambiente vibrante. Chega até a ser dito que é possível identificar o método de trabalho de uma organização simplesmente avaliando o ambiente no seu interior: se o nível de ruído for baixo, não está decerto a usar uma metodologia ágil.

Não é pouco comum implementar outras metodologias em paralelo com Scrum. Por exemplo, é frequente utilizar o Extreme Programming para colmatar algumas falhas possíveis, nomeadamente de qualidade, devidas à velocidade a que é desenvolvido o software (XP fornece algum controlo já que uma funcionalidade só é considerada como pronta quando aprovada em testes de unidade e aceitação). Desta forma é possível uma melhoria ainda mais eficaz nos processos de desenvolvimento de uma organização.

## 5 Vantagens vs. Desvantagens

O Scrum apresenta vantagens e desvantagens, tal como qualquer outra metodologia ou processo de fabrico de software. Pode ser implementado em conjunto com

outros métodos para tentar preencher algumas lacunas existentes no Scrum, sem obviamente se conseguir alcançar a perfeição.

O desenvolvimento de software nem sempre se baseia na perfeição. Muitas vezes é prioritário agradar ao cliente em diversos aspectos, como na rapidez de entrega e cumprimento de prazos e no acompanhamento deste na evolução do projecto (o que implica que haja mudanças repentinas e inesperadas na sua evolução). Para além disso, não devemos ignorar o facto de o cliente ser (geralmente) apenas um utilizador e desejar apenas que o programa funcione para o que ele quer, não lhe importando o modo como o faz acontecer.

Como referido anteriormente, devido à própria natureza de Scrum, a produtividade irá decerto aumentar. Tal facto não deixa de depender, porém, da dedicação da gestão do projecto e do rigor na implementação deste método. Assim sendo, os "Scrum Masters" devem conhecer o processo profundamente e praticá-lo o mais rigorosamente possível, de forma a obter a maior produtividade possível.

Ao nível da inovação, o Scrum fornece uma re-engenharia contínua, rápida e do tipo "bottom-up", permitindo mudanças repentinas e em massa ao projecto, sem grandes prejuízos em termos de tempo e custos. Tal é possível a partir do momento em que o projecto é repartido num conjunto de pequenos problemas, "pedaços" de um problema maior, muito mais fáceis de gerir do que o projecto seria na sua forma original. Normalmente, em projectos complexos, a interface entre os diversos módulos é demasiado complexa devido ao elevado grau de isolamento dos elementos envolvidos. No caso do Scrum, pelo facto de colocar as equipas a trabalhar juntas e a comunicar entre si, a integração de todo o trabalho torna-se bastante mais fácil, já que toda a gente sabe o que toda a gente está a fazer.

Outra vantagem é o curto intervalo de tempo entre as apresentações do trabalho produzido ao cliente. O cliente pode deste modo acompanhar de perto a evolução do projecto e obter/fornecer feedback frequente acerca das funcionalidades do produto. Desenvolve-se desde cedo uma relação com o cliente, constrói-se confiança e o conhecimento cresce. Como é obvio, também a comunicação entre os elementos da equipa melhora substancialmente e todos os sucessos ao longo do projecto são partilhados, o que contribui para que seja criada uma cultura em que toda a gente envolvida espera de facto que o projecto tenha sucesso, tendo isso enorme carácter motivador para a equipa. Isto é algo de grande importância nas organizações actuais. Este método pode no entanto ser aplicado a qualquer outro processo, já que é simplesmente um conjunto de valores, princípios e práticas.

No entanto, como qualquer método, também o Scrum apresenta algumas lacunas difíceis e, devido à sua própria natureza, talvez mesmo impossíveis de ultrapassar.

O Scrum é um método que exige uma gestão "on-hand". Isto implica que o gestor tem que estar constantemente disposto a efectuar alterações e modificações, de forma a providenciar assistência e ajudar as equipas a ter sucesso, removendo as barreiras que surgirem. Tal exige uma monitorização constante.

As equipas de desenvolvimento têm ainda que operar adaptativamente num ambiente complexo, usando processos imprecisos. Isto requer que a gerência confira autoridade de decisão às equipas de desenvolvimento. Os gestores devem deixar as equipas tomar as suas próprias decisões, permitindo até que estas falhem ao fazê-lo, se necessário. Tudo isto, embora permita uma maior evolução e desenvoltura em projectos futuros (se com os erros se aprender algo), causa certamente prejuízos não só em tempo como também em dinheiro. Permitir prejuízos a curto prazo, porém, pode reflectir-se em benefícios a longo prazo.

Além de todos estes factores, há ainda a ter em conta que o Scrum é um método novo e diferente e que as pessoas, na sua generalidade, são resistentes à mudança. Isto pode gerar um desconforto inicial com a prática deste método, podendo ser necessária uma fase de adaptação e, na pior das hipóteses, a possibilidade de alguns elementos não se conseguirem adaptar ao ritmo ou metodologia do Scrum. Sendo que é necessário mexer com toda uma cultura organizacional, a sua implementação deve ser analisada cuidadosamente. Após a adaptação ao método, outro problema comum é a possibilidade de alguns membros ou mesmo sub-gestores não se sentirem confortáveis com as responsabilidades que o Scrum exige, ou a equipa de gestores não gostarem da falta de visibilidade que passam a ter.

### 5.1 Comparação com outros modelos de processo

	<b>“Waterfall”</b>	<b>Espiral</b>	<b>Iterativo</b>	<b>SCRUM</b>
Processos definidos	Necessários	Necessários	Necessários	Planeamento e Terminação apenas
Produto Final	Determinado no planeamento	Determinado no planeamento	Determinado ao longo do projecto	Determinado ao longo do projecto
Custo do projecto	Determinado no planeamento	Em parte variável	Determinado ao longo do projecto	Determinado ao longo do projecto
Data final	Determinada no planeamento	Em parte variável	Determinada ao longo do projecto	Determinada ao longo do projecto
Adaptação ao ambiente	Apenas no planeamento	Principalmente no planeamento	No final de cada iteração	<b>Ao longo do projecto</b>
Flexibilidade e criatividade da equipa	Limitada, promove a reutilização	Limitada, promove a reutilização	Limitada, promove a reutilização	<b>Ilimitada durante as iterações</b>
Transferência de conhecimento	Formação prévia	Formação prévia	Formação prévia	<b>Trabalho de equipa durante o projecto</b>
Probabilidade de sucesso	Baixa	Média/baixa	Média	<b>Alta</b>

**Figura 2.** Tabela de comparação com outras metodologias do sector

Na tabela apresentada podemos verificar que, relativamente a outras metodologias do sector, o Scrum é a única que, entre outros factores, permite uma flexibilidade e criatividade ilimitadas, ao longo das iterações (reuniões e sucessivas

inovações no projecto a cada dia), o que realmente caracteriza este método. A transferência de conhecimento, como já havia sido dito, é feita ao longo do projecto em trabalho conjunto e partilha de objectivos alcançados. Conseguindo uma óptima avaliação do risco, Scrum consegue uma alta probabilidade de sucesso.

É possível fazer uma analogia do futuro das metodologias ágeis com as metodologias para desenvolvimento de software orientadas a objecto. Da mesma forma que várias metodologias de desenvolvimento orientadas a objecto surgiram e competiram entre si, surgiram também várias metodologias ágeis. Surgiu a necessidade de padronização que resultou na notação UML, actualmente um padrão na indústria de software. Da mesma forma, é possível ver várias metodologias ágeis em processo de fusão no futuro, principalmente em torno do eXtreme Programming, a mais aceite. Existe, por exemplo, um movimento para o uso conjunto do XP com Scrum (XP@Scrum). A XP seria usada para a fase de desenvolvimento e a Scrum para o planeamento e gestão do projecto. A integração das duas metodologias seria relativamente simples, uma vez que compartilham algumas características (como a necessidade da presença do cliente, pequenas releases e o encorajamento de mudanças necessárias para atender os requisitos reais dos stakeholders).

Concluindo, Scrum vê o desenvolvimento de software de uma forma empírica, muito como um bando de aves migratórias: elas avaliam o estado do tempo e decidem que está demasiado frio para continuarem onde estão; decididas a partir, analisam uma série de factores para determinar a direcção a seguir; chegando a um sítio suficientemente bom para permanecerem, ficarão por lá, mesmo que tenham percorrido apenas metade do caminho que inicialmente julgaram que iriam percorrer. Se o cliente viver durante mais um mês, ele vai mudar de ideias. Se um projecto durar mais do que um mês, os requisitos irão mudar e o desenvolvimento terá de se adaptar a essas mudanças se de facto quiser te-las em conta. É totalmente infrutífero perder tempo, esforço e dinheiro a planear exaustivamente um projecto no seu início se não podemos esperar que o mundo pare enquanto trabalhamos nele. Não é também de todo útil colocar pessoas, supostamente parte integrante de uma equipa, a trabalhar em cubículos e a fornecer dados apenas quando lhe são pedidos (o que pode nem chegar a acontecer). O desenvolvimento de software não se assemelha de nenhuma maneira ao fabrico de automóveis. O software nunca está completamente definido na sua fase embrionária, e não existe uma linha de produção em que as pessoas mais à frente nessa linha não precisam de se preocupar com o que outras antes de si terão feito. Porquê tratar o desenvolvimento de software como um processo definido, então? Assim sendo, Scrum resume-se a duas palavras: bom senso.

## Referências

- [Novembro 2005] <http://www.controlchaos.com>
- [Novembro 2005] <http://www.jeffsutherland.com>
- [Novembro 2005] <http://www.presidentekennedy.br>
- [Novembro 2005] <http://www.scrumalliance.org>

[Novembro 2005] <http://www.scrum-master.com>  
[Novembro 2005] [http://hubert.blogs.com/dahdidahdi\\_dadadihda/](http://hubert.blogs.com/dahdidahdi_dadadihda/)