**Como estamos indo com a adoção de Scrum na Globo.com**

Muita gente têm me perguntado como estamos indo atualmente com a adoção do [Scrum](http://pt.wikipedia.org/wiki/Scrum) na [Globo.com](http://globo.com). Acho que já é hora de falar alguma coisa sobre isso. Esse não é um case oficial assinado pela Globo.com, são apenas alguns relatos do meu ponto de vista sobre o assunto.

Nossa história no início foi bem parecida com as histórias tradicionais. Sabe aquelas empresas que gastam uma semi-fortuna com uma consultoria para desenhar um processo de desenvolvimento de software? Assim estávamos nós no início de 2007. Poucos projetos eram entregues nas datas acordadas e muitos deles falhavam ou não satisfaziam as necessidades dos clientes. Contratar uma grande consultoria de software parecia ser uma boa tentativa de arrumar as coisas, mas o resultado do trabalho foi um documento que descrevia um processo com algumas centenas de páginas que devia ser seguido á risca, isso sem contar as dúzias de documentos padrão para todos os tipos de requisição e comunicação que se possa imaginar. Dezenas de páginas descreviam quem deveria falar com quem e quem entrega qual documento em qual momento. O processo foi feito para lidar com todas as complexidades, burocracias e exigências que nós mesmos criamos dentro da empresa.

Resumindo a história, isso não funcionou na Globo.com e acho que existem poucas chances de algo tão complexo assim funcionar em algum lugar. E por incrível que pareça, no fim das contas ninguém havia pensado em nada para resolver o problema principal: como entregar o software que nossos clientes querem no menor tempo e com a maior qualidade possível?

**Um pouco de história**

Ao contrário do que acontece em muitas empresas, as [metodologias ágeis](http://pt.wikipedia.org/wiki/Desenvolvimento_%C3%A1gil_de_software) na Globo.com vieram de baixo para cima. Tudo começou com um movimento tímido entre alguns desenvolvedores de aplicar [práticas de desenvolvimento ágil do XP](http://www.improveit.com.br/xp/praticas) ([Extreme Programming](http://www.improveit.com.br/xp)) nos projetos. A porta de entrada foi a oportunidade de melhorar a qualidade dos nossos sistemas, pois tinhamos uma incidência muito grande de bugs em produção e poucas aplicações eram testadas adequadamente.

Alguns desenvolvedores mais experientes começaram a desenvolver usando [desenvolvimento guiado por testes (TDD)](http://gc.blog.br/2007/05/09/test-driven-development-in-a-nutshell/), e com isso houve uma melhora nítida na qualidade do que era entregue. Aos poucos, enquanto isso acontecia, algumas seções de [programação em par](http://www.improveit.com.br/xp/praticas/programacao_par) aconteciam de vez em quando para difundir o conhecimento e ensinar a técnica a outros programadores, mas ainda de forma muito tímida e sutil (afinal às vezes é difícil convencer que dois programadores fazendo apenas um código é uma coisa boa).

Em seguida foi a vez de organizar as demandas de clientes e os ciclos de desenvolvimento. Lembro-me de ter colocado nessa época o mesmo software em produção *quatro vezes em duas semanas*, e depois disso o mesmo software foi trabalhado durante dois meses para só depois poder ir novamente para produção. Para piorar, nossos clientes nos traziam todo tipo de demanda que se pode imaginar, sempre em lotes e com prioridade máxima. Por exemplo, acontecia frequentemente do mesmo cliente demandar cinco coisas e todas elas com máxima urgência. Resumindo, os ciclos de planejamento, desenvolvimento e entrega eram totalmente irregulares e afetavam o desempenho de toda a equipe. Explicando todas as dificuldades que tinhamos para nos organizar nesse caos, conseguimos acordar que fariamos entregas constantemente de duas em duas semanas, e que dentro desse período não haveria nenhuma mudança no planejamento, que seria feito no início de cada período. Antes de iniciar cada ciclo de desenvolvimento, os clientes tinham a oportunidade de escolher quais seriam as prioridades da equipe de desenvolvimento nas duas semanas seguintes. Sem perceber eles haviam aceitado a idéia de [desenvolvimento iterativo](http://pt.wikipedia.org/wiki/Desenvolvimento_iterativo_e_incremental) e [jogo do planejamento](http://c2.com/cgi/wiki?PlanningGame), e nós teriamos alguma organização e paz para fazer o trabalho.

Em meados de Julho de 2007 a empresa decidiu bancar um curso de Scrum com o [Boris Gloger](http://www.linkedin.com/in/borisgloger) para alguns membros da nossa equipe e o resultado foi ótimo! Na semana seguinte mesmo já começou a mudança. Todos voltaram com um novo gás e dispostos a mudar a empresa. Me lembro de nessa época alguém ter falado a seguinte frase: “Agora eu acredito em desenvolvimento de software”.

Desse momento em diante passamos a aplicar Scrum no nosso time de desenvolvimento. O problema é que estávamos no meio de um projeto feito da forma “tradicional” (leia-se [waterfall](http://en.wikipedia.org/wiki/Waterfall_model)) e toda a fase de análise e especificação já tinha sido concluída. Tudo indicava que seria melhor esperar uma oportunidade melhor para começarmos a adoção, só que nós sabiamos que tinha que ser “naquela hora ou nunca” e então começamos a desenvolver nosso principal projeto usando práticas do Scrum.

Quase ninguém na equipe havia trabalhado com desenvolvimento ágil, então achamos que seria mais fácil não falar de Scrum, XP ou qualquer nome diferente. Simplesmente começamos a praticar e durante o desenvolvimento o time foi introduzido às práticas e à forma de trabalhar. As regras são muito simples e por isso foi muita fácil de adotá-las. Ao longo dos cinco Sprints o time foi amadurecendo, entendendo como trabalhar e também foram nascendo algumas adaptações necessárias ao funcionamento do projeto dentro da estrutura da empresa. Por exemplo, tivemos que resolver como seriam criadas as histórias e como isso seria integrado com o nosso sistema de tracking, como integrar com a equipe de QA, como colocar os projetos em produção e diversas outras questões. O [Phillip falou bastante sobre essa fase introdutória no seu blog](http://blog.fragmental.com.br/2007/08/15/introduzindo-agilidade-num-ambiente/) no ano passado.

Hoje, o [Globo Vídeos 4.2](http://video.globo.com), que é o tal projeto, está em produção. Para os padrões da empresa na época, ele foi um projeto surpreendente do ponto de vista de qualidade. Na versão anterior do Globo Vídeos (4.0) foi necessária uma janela de mais de 24 horas para colocá-lo no ar e ela aconteceu aos trancos e barrancos. Além disso a semana seguinte foi infernal, com muitos bugs para serem corrigidos e várias mudanças arriscadas durante o dia para corrigi-los. Já a versão 4.2 foi colocada em produção em pouco mais de uma hora e na primeira semana nem ouviamos falar do Globo Vídeos. Nem parecia que um dos sites de maior audiência da empresa havia sido quase totalmente substituído por um totalmente novo e com grandes mudanças arquiteturais!

Ao mesmo tempo que acontecia o Globo Vìdeos, o site de inscrições para o [Big Brother Brasil 8](http://bbb.globo.com) foi desenvolvido de cabo a rabo usando Scrum, da forma estrita, como está nos livros. O projeto simplesmente não teria sido feito considerando-se a complexidade e o tempo disponíveis. No final, ele foi um sucesso e além de ter sido entregue no prazo houve uma percepção de muita qualidade por todos – mais uma prova viva de que o Scrum era uma possível resposta para os nossos problemas. O [Danilo inclusive fez uma apresentação sobre esse projeto](http://blog.bardusco.com/2008/05/26/scrum-cmmi/) na semana passada num evento do [C.E.S.A.R.](http://www.cesar.org.br/) em Recife.

Esses dois projetos serviram de aprendizado e base para a estruturação de todos os projetos seguintes na empresa. O sucesso deles foi o grande responsável para ganharmos carta branca e começarmos a implementação de Scrum em massa na Globo.com.

**Como estamos hoje**

Após um ano de metodologias ágeis a empresa já está bem mais madura nas práticas e já temos mais de 10 times usando Scrum.

No fim do ano passado, após um treinamento para pessoas de todas as áreas da empresa, começamos todos a usar Scrum da forma estrita (obviamente houve uma curva de aprendizado até todos estarem mais seguros e praticando melhor). Hoje, alguns meses depois, já é possível perceber diferenças entre alguns times, que acabaram se adaptando de forma diferente por terem problemas e questões diferentes.

Sobre a duração dos [Sprints](http://www.mountaingoatsoftware.com/scrum), estamos trabalhando com duas semanas porque achamos que quatro semanas é muito tempo e poderiamos acabar nos tornando lentos na resposta às mudanças. Além do mais já vinhamos usando iterações de duas semanas desde que começamos a adotar desenvolvimento iterativo e continuar com este tamanho de ciclo seria mais fácil para nós.

Em virtude disso, como temos a metade do tempo de desenvolvimento recomendado pelo Scrum, decidimos que só precisamos da metade do tempo de planejamento. A recomendação é que o [Sprint Planning](http://www.mountaingoatsoftware.com/sprint_planning_meeting) tenha 8 horas para um Sprint de 4 semanas, e como nosso Sprint é de 2 semanas decidimos fazer um planning de 4 horas. As outras reuniões ([Sprint Review](http://www.mountaingoatsoftware.com/sprint_review_meeting), [Retrospective](http://www.improveit.com.br/scrum/sprint_retrospective), [Daily Scrum](http://www.mountaingoatsoftware.com/daily_scrum), etc) permaneceram com a mesma duração, lembrando que essa duração não é obrigatória mas sim um [limite para não ficarmos discutindo as coisas indefinidamente](http://blog.improveit.com.br/articles/2007/01/04/time-box).

Nossos Sprint Plannings têm melhorado constantemente. No início sempre estouravamos o tempo da reunião discutindo um monte de coisas que não eram necessárias mas agora já estamos mais focados e a reunião tem funcionado bem melhor. As estimativas com [planning poker](http://www.planningpoker.com/) também evoluiram um bocado e em várias ocasiões todos os desenvolvedores colocam o mesmo número sem combinar, o que indica que estamos evoluindo na sensação de esforço necessário para desenvolver as coisas.

A equipe, que antes era só de desenvolvedores, agora tem também um designer, um arquiteto de informação, um especialista em programação client-side (JavaScript/CSS/HTML), uma pessoa da equipe de testes e homologação e uma pessoa focada em negócios e [ROI](http://en.wikipedia.org/wiki/Return_on_investment) (que é o [Product Owner](http://www.mountaingoatsoftware.com/product_owner)). Todas essas pessoas sentam próximas umas das outras, e isso efetivamente aumentou muito a produtividade delas e o ritmo do projeto. Aos poucos todo o conhecimento específico está sendo difundido entre os membros da equipe. Infelizmente nem todas as equipes estão completas dessa forma por diferentes motivos (espaço físico, distância entre os prédios da empresa, etc), mas estamos trabalhando duro para resolver isso. Inclusive temos uma reunião chamada de [Scrum Of Scrums](http://www.scrumalliance.org/articles/46-advice-on-conducting-the-scrum-of-scrums-meeting), onde todos os Scrum Masters se reunem para se ajudarem a resolver esses tipos de problemas, que muitas vezes são comuns a várias equipes.

Já que o Scrum não fala nada sobre práticas de desenvolvimento de software, acabamos adotando muitas práticas do [XP](http://www.improveit.com.br/xp). Isso fica por conta de cada equipe e cada uma faz o que julga mais adequado para entregar o melhor software possível, mas muitas equipes usam [desenvolvimento guiado por testes](http://www.improveit.com.br/xp/praticas/tdd), [integração contínua](http://www.improveit.com.br/xp/praticas/integracao), [programação em par](http://www.improveit.com.br/xp/praticas/programacao_par), [metáforas](http://www.improveit.com.br/xp/praticas/metafora) e várias outras práticas de Extreme Programming com muito sucesso. De fato [a integração entre Scrum e XP funciona muito bem](http://gc.blog.br/2008/03/31/xp-complementa-o-scrum/).

Definimos o nosso [conceito de “done” (finalizado)](http://www.agile-software-development.com/2007/07/definition-of-done-10-point-checklist.html), que é a parte mais divergente entre as equipes. Na equipe de desenvolvimento de vídeos (minha equipe), uma história ou funcionalidade está pronta quando foi desenvolvida (incluindo [testes unitários](http://pt.wikipedia.org/wiki/Teste_unit%C3%A1rio)), integrada à base de código, testada novamente (fazemos [testes de aceitação com critérios definidos no Sprint Planning ao criar as histórias](http://gcirne.wordpress.com/2008/04/21/criterios-de-aceitacao/)) e homologada. Como nosso time tem uma pessoa da área de [QA](http://en.wikipedia.org/wiki/Quality_Assurance), podemos incluir a homologação da aplicação dentro do Sprint (nem todos os times fazem dessa forma).

Temos na empresa uma equipe que é especializada no ambiente de produção e por isso colocar os sistemas no ar fica fora do nosso Sprint. Quando nosso Sprint termina, entregamos um pacote fechado, testado, homologado e pronto para ser colocado em funcionamento, e então agendamos uma data e hora para que a subida seja feita acompanhada por um ou mais desenvolvedores do time. Todas as alterações de arquitetura e ambiente de deployment, quando necessárias, são feitas dentro do Sprint, somente as mudanças que envolvem risco de indisponibilidade nos sistemas que são feitas numa data que agendamos após o termino do Sprint, geralmente no meio da madrugada (as famosas “janelas”). Alguns times, por terem menos criticidade no seu ambiente de deployment, consideram que um Sprint está concluído quando o software está no ar, e o seu último dia do Sprint sempre é uma subida para produção. Como já havia falado anteriormente, cada time se adaptou para funcionar da melhor forma possível de acordo com suas peculiaridades.

E por último, nossas retrospectivas têm sido uma base forte para adaptações no processo e na forma de trabalhar. Para mim foi um aprendizado enorme quando o [Boris Gloger veio na Globo.com](http://gc.blog.br/2008/02/01/sprint-review-e-retrospective-com-boris-gloger/) e acompanhou uma retrospectiva inteira do nosso time – ele deu excelentes toques importantissimos. A última retrospectiva em especial, que aconteceu após uma grande entrega de um projeto interno, mostrou nitidamente a evolução do time e como estamos efetivamente conseguindo aos poucos achar e resolver todos os problemas. Estamos evoluindo devagar mas constantemente e já temos várias equipes com um bom nível de maturidade e evolução na empresa.

**Concluindo…**

Não só o Scrum como todos as metodologias ágeis dependem muito das pessoas. Na Globo.com não foi diferente e as pessoas certas fizeram toda a diferença. Também existe o outro lado da moeda: algumas pessoas simplesmente não se adaptam a essa forma de trabalhar. Desde o início da adoção nós já perdemos muitos desenvolvedores e acredito que ainda perderemos muito mais. Por conta disso [nossos processos seletivos se tornaram mais exigentes e demorados](http://gc.blog.br/2008/01/21/sobre-entrevistas-parte-2/), porque agora não só precisamos de pessoas que sejam ótimas tecnicamente mas também que tenham um perfil adequado para trabalhar no tipo de ambiente que criamos dentro da empresa.

Além disso é essencial que haja apoio da gerência e “carta branca” para que as equipes de desenvolvimento tenham a autonomia necessária para levar o projeto da forma correta. Como estes “processos” são muito diferentes dos tradicionais, algumas empresas acabam fazendo modificações antes do tempo por puro medo ou falta de conhecimento, que acabam atrapalhando ou até mesmo arruinando a adoção de uma metodologia ágil. Ainda em relação aos times de desenvolvimento, é essencial que os líderes de equipe (ou [Scrum Masters](http://www.improveit.com.br/scrum/scrum_master) no caso do Scrum) sejam muito bem capacitados e que conheçam profundamente as práticas/regras/princípios, não só para que tenham capacidade de argumentação com a empresa mas também para que não façam adaptações que violem os [princípios básicos das metodologias ágeis](http://agilemanifesto.org/).

Por fim, acho que ainda estamos muito longe do ideal e vejo muitas oportunidades de melhoria na nossa forma de trabalhar, mas o mais importante é que agora acreditamos que estamos no caminho certo.

Tags: [Boris Gloger](http://gc.blog.br/tag/boris-gloger/), [Desenvolvimento Ágil](http://gc.blog.br/tag/desenvolvimento-agil/), [Globo Vídeos](http://gc.blog.br/tag/globo-videos/), [Globo.com](http://gc.blog.br/tag/globocom/), [Scrum](http://gc.blog.br/tag/scrum/), [Scrum Master](http://gc.blog.br/tag/scrum-master/), [Sprint](http://gc.blog.br/tag/sprint/), [Sprint Retrospective](http://gc.blog.br/tag/sprint-retrospective/), [Sprint Review](http://gc.blog.br/tag/sprint-review/), [TDD](http://gc.blog.br/tag/tdd/), [XP](http://gc.blog.br/tag/xp/)

Origem:<http://gc.blog.br/2008/05/27/como-estamos-indo-com-a-adocao-de-scrum-na-globocom/>