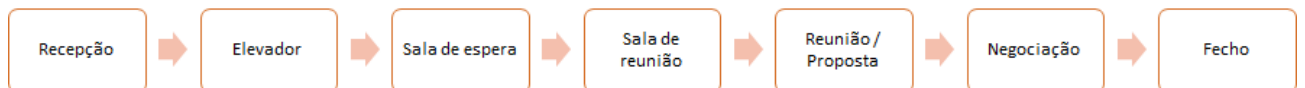


Calculando o prazo de um projeto ágil

Estimando o tamanho do projeto

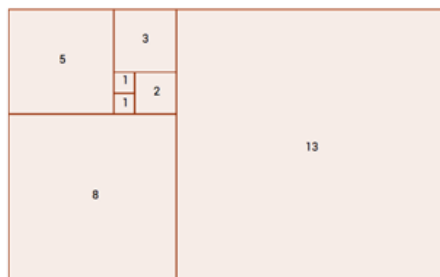
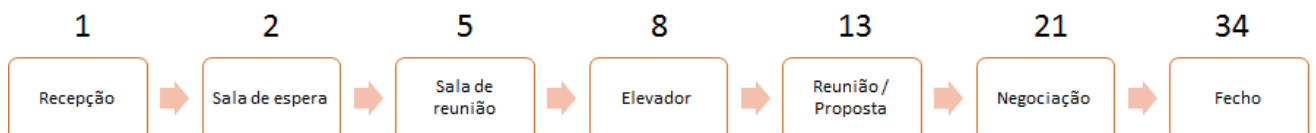
Uma das técnicas para cálculo dos *story points* é o *Planning Wall*. Nela, você deve primeiramente listar as *estórias de usuário*, geralmente numa parede usando post-its.



Em seguida, devem ordená-las por complexidade, ignorando-se o fator tempo (não se atenha se os itens que escolhi são realmente mais complexos, é apenas um modelo ilustrativo).



por último, deve usar uma escala não-linear para pontuar cada *estória*, como a sequência de Fibonacci, por exemplo: 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, ...



Com isso, você terá o tamanho do projeto: $1+2+5+8+13+21+34 = 84$. Este número será utilizado para medir o progresso.

- Quantidade total de story points?
- Quantidade de story points já realizados?
- Quantidade restante de story points?
- Tempo total do projeto?

Progresso x Velocidade

Mede-se o tamanho do projeto e não o prazo, pois, no Scrum, não importa quanto tempo passou e sim o quanto foi concluído.

Daqui nasce a diferença entre *progresso* e *velocidade*:

- Quanto mais histórias o time terminar, mais o projeto PROGREDIU. Ou seja, mais próximo está de terminar todas as histórias.
- Quanto mais rápido o time terminar, mais VELOCIDADE ele está alcançando.

Calculando a Velocidade do time e o Prazo do projeto

Então, digamos que foi definida uma duração de 2 semanas para cada sprint e que é provável que o time faça as 2 primeiras histórias na *sprint 1*, mas que, se sobrar tempo, podem começar a trabalhar na terceira história.

As duas primeiras histórias são, portanto, a meta da sprint e devem cumprir o objetivo traçado para a primeira *sprint*. A terceira *história* é desejável, se der tempo, o projeto avançará (terá progresso) mais rapidamente.

Usando o tempo como medida

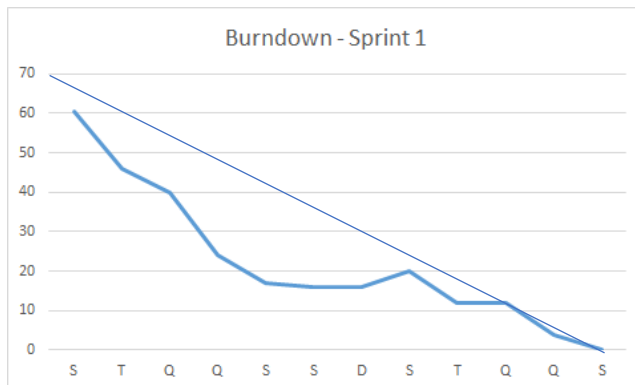
No planejamento da *sprint*, o time desdobra as atividades necessárias para finalizar cada *história*. O quadro (*sprint backlog*) fica mais ou menos assim.

Horas por atividade		
Recepção	Buscar fornecedores	2
	Reunir com eles	16
	Escolher um fornecedor	4
	Fazer o contrato	8
	Assinar o contrato	4
	Iniciar setup	2
	Organizar local	8
Sala de espera	Selecionar móveis	4
	Escolher móveis	12
	Aprovar a compra	4
	Receber os móveis	2
	Organizar local	8
Sala de reunião	<Não planejado>	0
TOTAL DE HORAS		74

Aqui temos um primeiro *insight*: Duas semanas têm 40 horas por pessoa. Se consideramos 3 pessoas no projeto, já começou a sobrar tempo.

Simulando *sprint 1*

Então, duas semanas depois, chega-se ao seguinte *sprint burndown*:



Perguntas:

- Terá o time avançado na velocidade esperada?
- Houve dificuldades no meio do caminho?
- O que fez com que a primeira semana fosse mais rápida que a segunda?

Nesse projeto, já se conhecia um fornecedor de móveis, então as etapas de seleção de fornecedor e contrato foram mais rápidas. As aprovações da sala de espera (segunda história) demoraram mais que o previsto, pois o *Product Owner* não havia passado informações detalhadas do que gostaria de fazer. Por isso é importante o conceito de *definition of ready*, para que só se aceitem histórias que já tenham suas definições bem planejadas. Passar o trabalho do PO para o time reduz a produtividade.

Qual foi a velocidade do time?

3 story points (das duas histórias) por sprint. Sendo assim, o projeto terminará em:

- $84 \text{ sp} / (3 \text{ sp/sprint}) = 28 \text{ sprints}$
- Como cada sprint leva duas semanas => 56 semanas ou 14 meses.

Perguntas:

- Mas, será que a velocidade será constante nas demais sprints?
- Será que as atividades mais demoradas já não foram cumpridas?
- Será que a integração do time melhorará nas próximas rodadas?

Não se sabe.

Por isso, é importante acompanhar algumas *sprints* para se ter uma ideia mais concreta de qual será a velocidade média.

Simulando as *sprints* 2 e 3

Digamos que na segunda *sprint*, o time tenha concluído a *estória* “Sala de reunião” e na terceira tenha concluído a *estória* “elevador”. Como ficaria a velocidade média?

- Sprint 1 = 3 story points
- Sprint 2 = 5 story points
- Sprint 3 = 8 story points
- Média = $(3+5+8)/3 = 5,33$ sp/sprint

Perceba como é mais coerente considerar a variação de produtividade ao longo de várias *sprints*, por isso se considera sempre a média (ou média móvel, se preferir) para estimar o projeto.

Como ficaria o prazo do projeto considerando a perspectiva da velocidade média?

- Quantidade total de story points = 84
- Quantidade de story points já realizados = $(3+5+8) = 16$
- Quantidade restante de story points = $(84-16) = 68$
- Tempo total do projeto, após a revisão = $68 / 5,33 = 12$ sprints
- Em termos de prazo, ficaria em = $(12 \text{ sprints} * 2 \text{ semanas para cada}) / 4 \text{ semanas por mês} = 24 \text{ semanas ou } 6 \text{ meses}$.

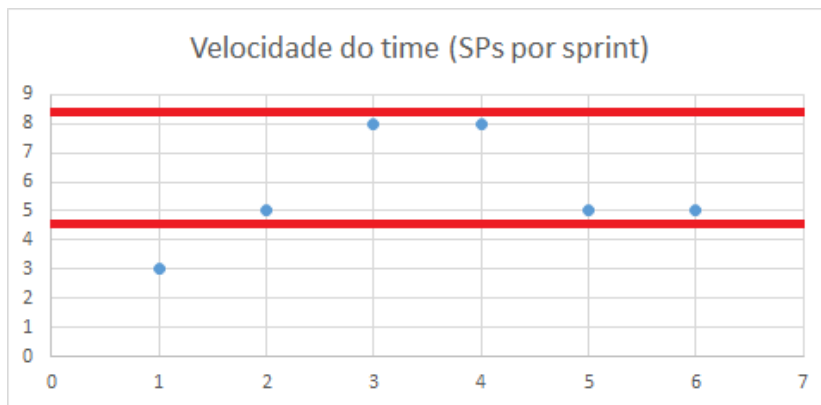
Pois é, num ambiente convencional, ao término da primeira *sprint* já se estariam fazendo reuniões de melhoria, o time teria levado muitas broncas e provavelmente alguém acabaria saindo da equipe, o que derrubaria ainda mais a produtividade.

Como acompanhar a velocidade do time?

A velocidade do time é medida por quantos *story points* aquela equipe conseguiu fazer num período fixo. Os *story points*, no entanto, são medidas comparativas entre *estórias de usuário*, que usam escalas não-lineares.

Deste modo, ao comparar uma *estória de usuário* com outra, o time é obrigado a escolher, por exemplo, entre 8 e 13, de modo que uma *estória* que valha 8 *story points* pode acabar levar mais tempo que outra com o mesmo valor. Isso serve para que o time “perca a noção do tempo exato” para fazer uma *estória* e acabe focando-se nos resultados. Esse é um dos grandes segredos do acompanhamento ágil, foco nos resultados.

Para acompanhar a velocidade, deve-se usar um [gráfico de controle](#), como o exemplo a seguir.



As linhas em vermelho mostram os limites superior e inferior, apenas como alerta para que o *Scrum Master* avalie se houve um potencial desvio. Basicamente, o gráfico de velocidade serve para dar uma noção do prazo do projeto, mas sem certezas, afinal, o Scrum foca em resultados não em estimativas.

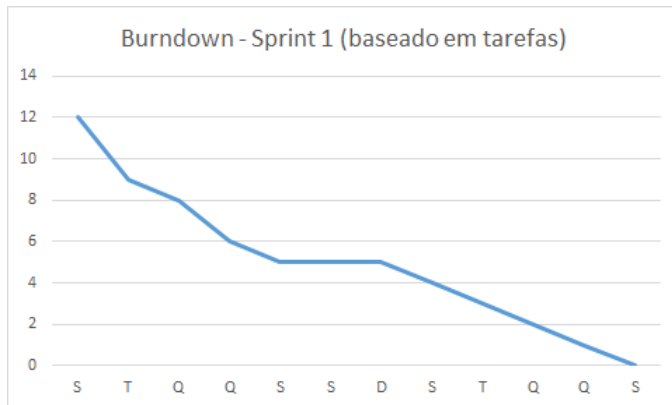
Usando a quantidade de tarefas como medida

Uma outra forma de controlar o andamento da *sprint* é por **quantidade de atividades**. Ela é mais rápida e mantém as pessoas focadas no que deve ser feito e não em quanto tempo levará para fazer.

		Qtde de atividades
Recepção	Buscar fornecedores	1
	Reunir com eles	1
	Escolher um fornecedor	1
	Fazer o contrato	1
	Assinar o contrato	1
	Iniciar o setup	1
Sala de espera	Organizar local	1
	Selecionar móveis	1
	Escolher móveis	1
	Aprovar a compra	1
	Receber os móveis	1
	Organizar local	1
Sala de reunião	<Não planejado>	0
TOTAL DE ATIVIDADES		12

O *burndown* tem o mesmo formato do gráfico baseado em horas, mas, retrata apenas atividades concluídas.

Se lembrarmos da diferença entre progresso e velocidade, notaremos que “terminar uma atividade não significa finalizar o trabalho (*a estória*), afinal, a atividade pode não ter dado certo.

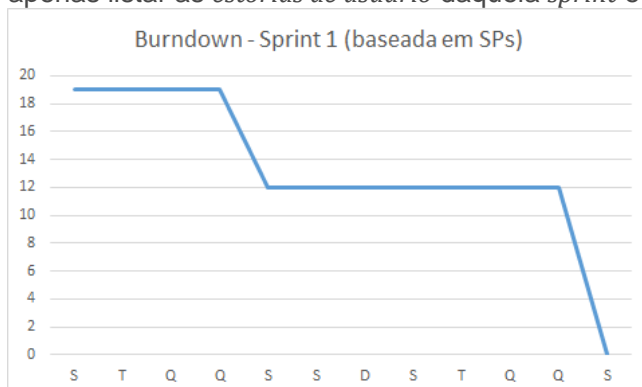


Talvez lhe incomode não saber as horas que o time gastou com cada tarefa, mas isso é realmente mais importante que o resultado?

O andamento do projeto é acompanhado na *daily meeting* e o tempo que se investe estimando horas é desperdiçado em relação ao tempo de execução.

Usando story points como medida

Para os mais radicais, nem mesmo estimar a quantidade de tarefas é justificável. Preferem apenas listar as *estórias de usuário* daquela *sprint* e seguir para a execução.

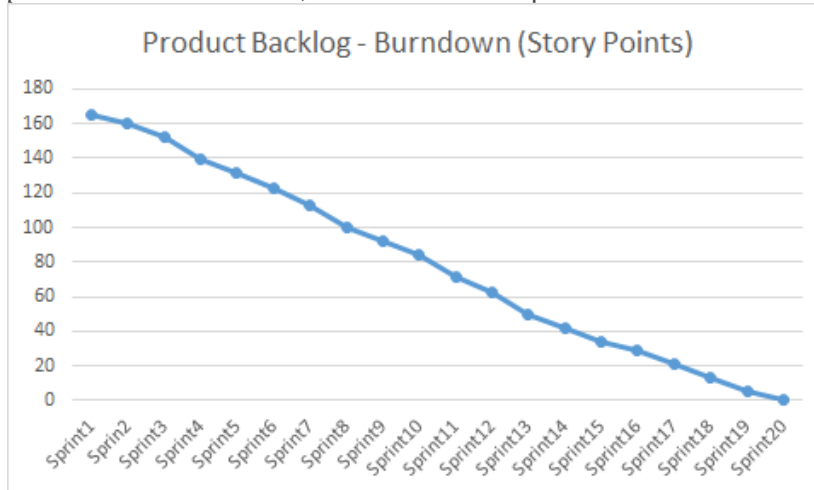


Quando se têm um grande número de *estórias*, o gráfico faz mais sentido. Mas, de qualquer forma, só haverá progresso quando uma *estória* for 100% concluída, ou seja, quando atender a *definition of done*.

Acompanhando o Progresso do projeto

Agora voltando ao *product backlog*: Como o Product Owner deve acompanhar o progresso do projeto?

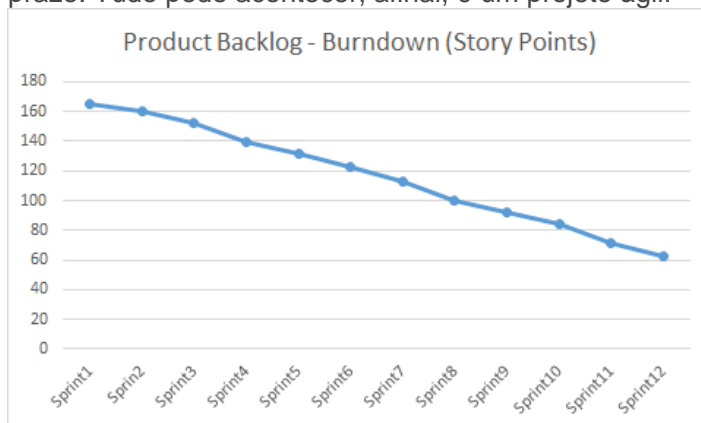
Ao final de cada sprint, deve atualizar o gráfico de *burndown*, mostrando quantos *story points* foram finalizados, conforme o exemplo.



Replanejamento a cada sprint

Seria sensacional se os projetos seguissem uma linha reta, mas na prática, o PO (Product Owner) decidirá, a cada planejamento de sprint (*sprint planning*), quais histórias permanecem, quais serão simplificadas e quais serão adicionadas.

Em termos práticos, a *estória* "Negociação" poderia ser "repriorizada" para entrar na quarta sprint. Ela também poderia ser "simplificada", reduzindo sua complexidade para caber no prazo. Tudo pode acontecer, afinal, é um projeto ágil!



O mais comum é que o *product burndown* finalize com algumas *estórias* não implementadas, seja porque o prazo terminou ou porque não houve interesse em seguir com elas. Mas, não se preocupe, na segunda-feira tudo que foi produzido irá ao ar e não haverá surpresas no último dia.