

Visualizar o fluxo de desenvolvimento de um time ágil

5.1 Introdução

Trabalhar com o desenvolvimento de software é um grande desafio. Afinal, temos de lidar diariamente com incertezas, mudanças, pessoas, expectativas, dependências, restrições etc.

Mapear um fluxo que garanta o desenvolvimento de software em um ritmo sustentável, ou até mesmo que proteja as equipes das incessantes demandas geradas por diferentes facetas envolvidas na construção de um produto ou projeto, é um dos grandes desafios de Gestores, Agile Coaches, Scrum Masters e time.

Com as etapas do fluxo de desenvolvimento estruturadas, as demandas (como funcionalidades, *bugs* etc.) passam a ser construídas seguindo a lógica pensada para o processo. E é aí que passam a surgir gargalos, alto tempo de entrega e baixa sensação de progresso.

Se você faz parte de uma equipe que possui um fluxo de desenvolvimento, em algum momento do seu dia a dia, você se deparará com a seguinte questão: como eu posso dar visibilidade de como está a saúde do meu processo?

Independente do seu método de trabalho, seja Kanban, Scrum, XP etc., desenvolver uma visão que apresente a quantidade de itens entregues, a quantidade de itens em progresso e o montante de trabalho a ser feito deve ser um tipo de prática difundida na rotina de qualquer equipe.

Dessa forma, você estará promovendo uma cultura que privilegiará: cadência de entrega; melhoria baseada em dados; visibilidade do fluxo de trabalho; e, principalmente, uma constante inspeção e adaptação do processo. Em resumo, você estará buscando entregar valor mais cedo aos seus usuários e clientes.

Neste capítulo, teremos a oportunidade de discutir a construção do CFD (*Cumulative Flow Diagram*), bem como o benefício de sua análise e sugestões de métricas que você poderá extrair dele.

Se você já conhece o CFD, não deixe de conferir o estudo de caso que será relatado mais adiante. Faça uma reflexão de como o aprendizado gerado pelo projeto citado poderá ajudá-lo a melhorar o seu dia a dia, afinal, sábio é aquele que aprende com os erros e acertos

dos outros.

Ao final deste capítulo, espero que você entenda como o CFD pode se tornar uma importante ferramenta de monitoramento do processo de um time ágil.

5.2 O que é o CFD?

Primeiramente, é importante dizer que o CFD é um tipo de visualização que trata essencialmente de entradas e saídas. Como o próprio nome pode sugerir, o *Cumulative Flow Diagram* é uma excelente forma de compreender o fluxo de trabalho ao longo de um processo, afinal, o gráfico pode nos dar um panorama geral do que está acontecendo durante o desenvolvimento de um projeto ou produto.

O CFD é um instrumento de muito valor no processo de monitoramento em projetos ágeis, pois, usando-o, você poderá rapidamente analisar: quanto de trabalho foi realizado, quanto de trabalho está em progresso e quanto de trabalho ainda precisa ser feito.

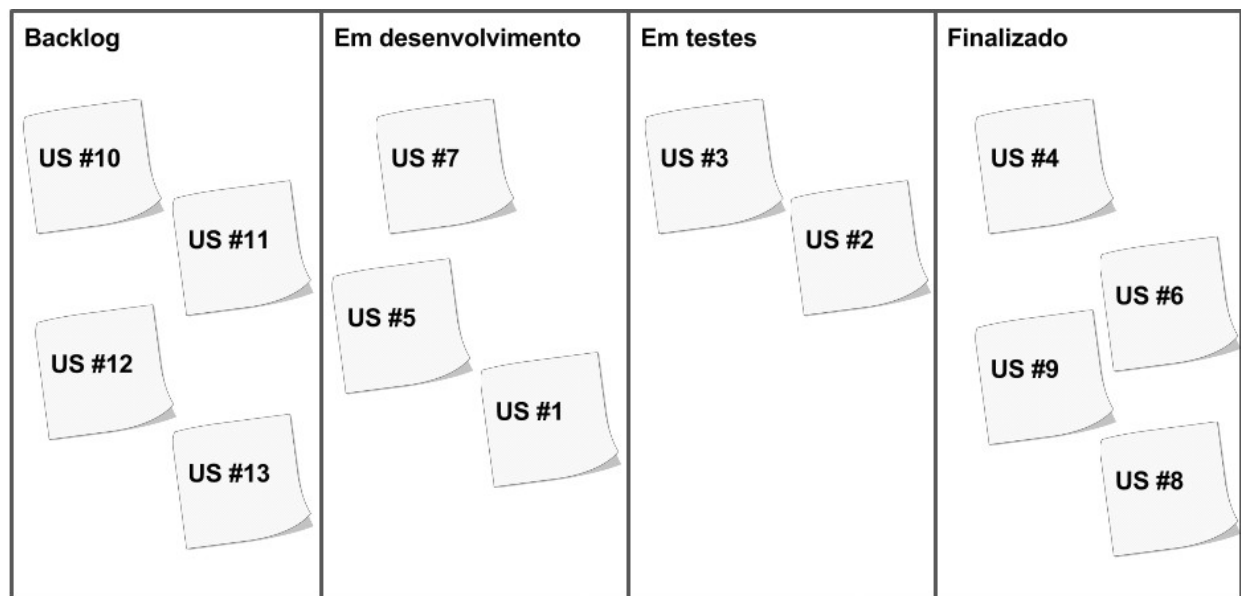
Visualizar o fluxo de um time a partir do CFD nos fornece *insights* quantitativos e qualitativos de potenciais ou reais problemas que possam ocorrer no contexto de trabalho do time. Afinal de contas, compreender a atual performance do processo é um dos primeiros passos para introduzir um sistema que tenha como fundação uma filosofia voltada à previsibilidade e melhoria contínua.

A estrutura do gráfico é muito simples. O eixo horizontal representa um período de tempo (semanas, *sprints* etc.) e o eixo vertical indica, de forma acumulada, o número de itens no processo (total de tarefas, total de histórias etc.). Cada área pintada no gráfico está relacionada a uma etapa do fluxo de trabalho (backlog, em progresso, finalizado etc.) e as curvas são basicamente o número de itens acumulados em tais etapas.

Vamos compreender como o que foi descrito anteriormente pode ser visto na prática. Imagine que o time XPTO possua o seguinte fluxo de trabalho:

- **Backlog:** primeira etapa do processo onde o time discute e detalha as funcionalidades priorizadas e que foram selecionadas para serem construídas.
- **Em desenvolvimento:** etapa na qual o time passa a construir as funcionalidades.
- **Em testes:** etapa na qual são realizadas validações e testes das funcionalidades construídas pelo time.
- **Finalizado:** última etapa do processo de desenvolvimento em que as funcionalidades são aprovadas e publicadas para os usuários finais.

A figura a seguir apresenta como as etapas descritas anteriormente estariam representadas em um quadro que expõe visualmente o trabalho do time.



Legenda:

US: User story = História do usuário.

Figura 5.1: Quadro mapeando o fluxo de trabalho do time XPTO

Mas, afinal, qual seria uma representação possível do CFD do time XPTO?

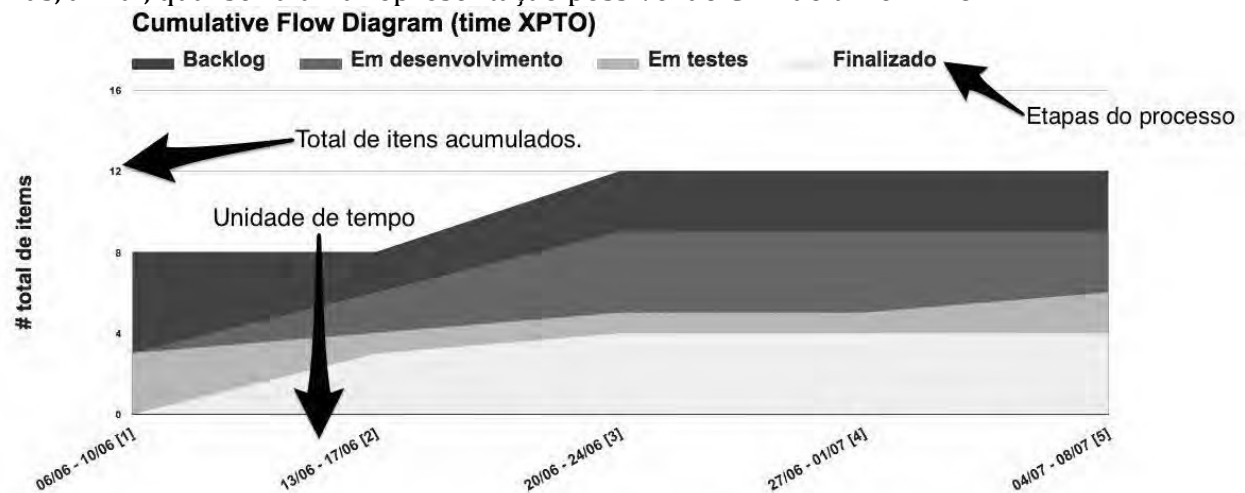


Figura 5.2: Exemplo de CFD a partir do processo do time XTPO

Conforme podemos observar no eixo horizontal da figura anterior, o time XPTO realiza um acompanhamento do seu processo de desenvolvimento semanalmente a partir de uma progressão do tempo da esquerda para a direita.

Um ponto importante de salientar é que a unidade de tempo e forma de ordenação definida para o CFD do seu time pode ser outra. Sendo assim, antes de fazer qualquer tipo de interpretação, compreenda em que unidade de tempo o gráfico está se comunicando e como ele está organizado.

As etapas do processo de desenvolvimento do time e o valor acumulado de itens que passaram por cada uma delas estão representadas no gráfico a partir das curvas "Backlog", "Em desenvolvimento", "Em teste" e "Finalizado". Ainda usando o exemplo de CFD, podemos dizer que, na quinta semana (04/07 até 08/07), o time XPTO possuía 12 itens que já haviam passado pela etapa de "Backlog", 9 itens que passaram pela etapa de "Em desenvolvimento", 6 que passaram pela etapa de "Em testes" e 4 que passaram pela etapa "Finalizado".

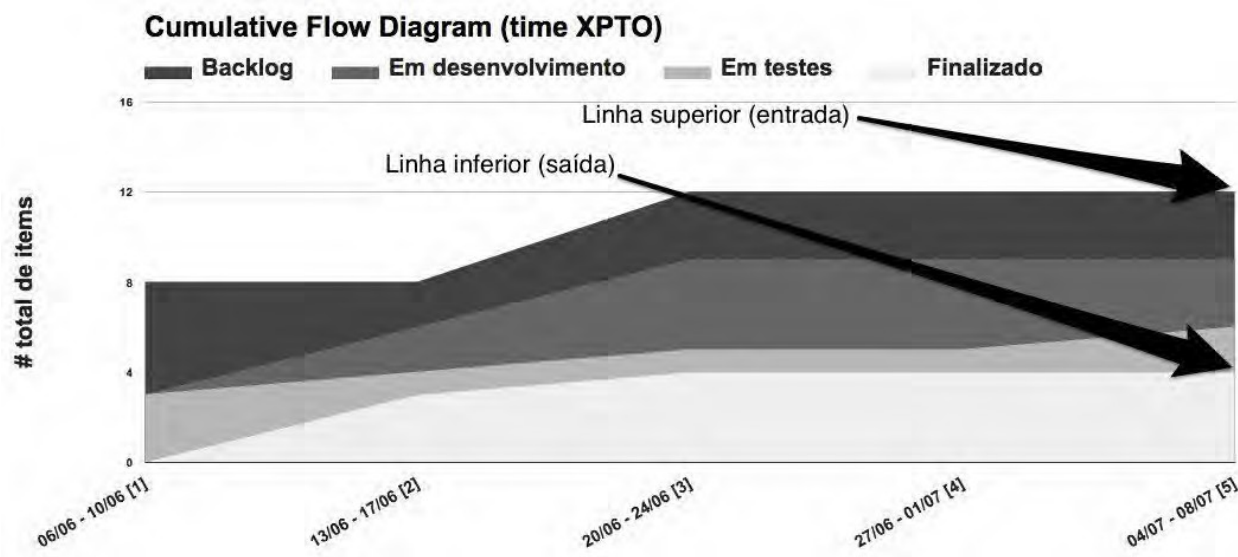


Figura 5.3: Linhas superior e inferior em um gráfico de CFD

Ainda sobre como ler o CFD, podemos analisá-lo a partir do conceito de entradas e saídas do processo (linhas superior e inferior, respectivamente).

Tendo como referência a quinta semana do time XPTO, ilustrada na figura anterior, podemos afirmar que 12 itens entraram para o processo de desenvolvimento (valor representado na linha superior, isto é, a primeira etapa do processo) e que 4 foram entregues (valor representado pela linha inferior, isto é, a última etapa do processo).

Para que a análise citada anteriormente faça sentido, é fundamental que o gráfico de CFD contenha apenas itens que passarão por todo o fluxo de trabalho. Isto significa dizer que, dado que o time assumiu o compromisso de desenvolver determinado item, este não poderá sair do processo, o que fará com que a linha superior nunca tenha o comportamento de queda.

A pergunta que você pode estar se fazendo neste momento é: “Afim, o backlog de um time não pode diminuir?”. A resposta é sim, porém, para que o CFD respeite a característica de ser uma análise **acumulativa**, o sistema de trabalho deve incluir apenas itens que já foram priorizados, analisados, refinados e estão prontos para entrar no fluxo.

Por esse motivo, torna-se relevante compreender o que está por trás da definição de “Backlog”. No capítulo no qual trataremos sobre a visualização do gráfico de Burnup, explicarei por que ela sim deve representar a variação do escopo, dado que, no contexto do CFD, estamos discutindo e analisando o processo.

A melhor maneira de você capturar os dados para um CFD é rastrear a data em que os itens que passam pelo processo do time entram em cada etapa. Coletar tais dados será também insumo para que você possa analisar outras métricas, como o *lead time* e o *throughput*.

Mapear as entradas e saídas de um processo a partir de uma perspectiva acumulativa ao longo do tempo é uma das melhores ferramentas disponíveis para que você visualize a saúde do processo do seu time.

5.3 Como analisar o CFD a partir de um caso real

Nesta seção, teremos a oportunidade de analisar, a partir de um caso real, como o CFD foi útil para que um time pudesse avaliar a saúde do processo ao longo do desenvolvimento de um produto.

O CFD foi usado como forma de comunicar o progresso do trabalho para os principais interessados no resultado do projeto, bem como para identificar gargalos no processo e, principalmente, para gerenciar estoques nas etapas do fluxo de desenvolvimento.

Antes do início do projeto, foram realizados *workshops* educativos com o objetivo de ensinar o time a interpretar o fluxo de trabalho a partir do CFD.

Caso: desenvolvimento da nova versão de uma plataforma de B2B

O caso que será descrito se passou em uma equipe que trabalhou na recriação de uma aplicação B2B (*business to business*), responsável por conectar fornecedores de materiais esportivos e lojas de varejo.

A aplicação possuía 10 anos de existência e contava, em seu leque de usuários, com mais de 5.000 lojas compradoras e 800 fornecedores. Por mês, a plataforma transacionava mais de 10 milhões de reais e a empresa detentora da tecnologia era a principal líder deste tipo de solução no Brasil.

Em decorrência da chegada de novos concorrentes, alguns usuários começaram a migrar para as novas soluções, pois consideravam que a plataforma atual da empresa tinha baixo poder de inovação. As atualizações eram demoradas e as funcionalidades lançadas tinham pouco valor para quem utilizava a solução no dia a dia.

Um dos maiores desafios técnicos do projeto foi garantir a compatibilidade entre a nova e a velha tecnologia. Afinal de contas, o patrocinador colocou como restrição que a plataforma antiga deveria coexistir com a nova, já que ele não gostaria de gerar uma mudança drástica no dia a dia dos usuários.

O time teria 5 meses para realizar o primeiro grande lançamento da nova versão do produto, pois a área comercial da empresa gostaria de aproveitar uma feira do meio esportivo para promover a solução junto aos seus parceiros e usuários. A equipe que trabalharia no desenvolvimento era composta por: 4 desenvolvedores; 1 Agile Coach; 1 testador; e 1 gestora de produto.

O time decidiu trabalhar com o método Kanban e incorporou para o seu processo de desenvolvimento: cerimônias de planejamento, nas quais eram priorizadas e discutidas as funcionalidades que entrariam no fluxo de desenvolvimento; retrospectivas, onde eram discutidas melhorias de processo, principalmente, a partir das métricas coletadas; e refinamento, em que eram avaliadas as incertezas, complexidades, dependências e critérios de aceite das funcionalidades que seriam trazidas para o fluxo de desenvolvimento.

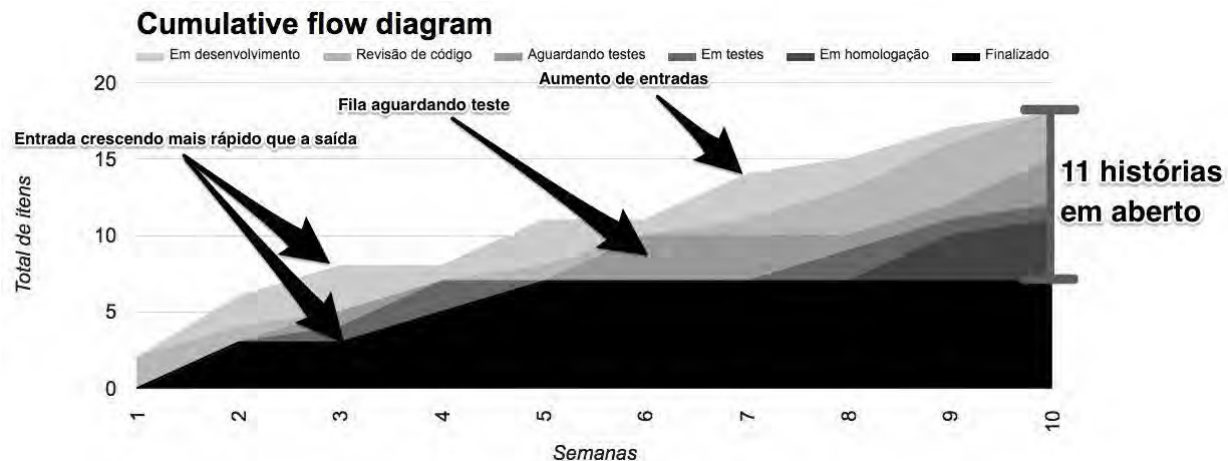
O fluxo de trabalho do time estava dividido nas seguintes etapas:

- **Em desenvolvimento:** etapa na qual a demanda que estava refinada e priorizada entrava para o fluxo de desenvolvimento do time. Neste momento, o time assumia, junto a gestora de produto, a responsabilidade de entregar a funcionalidade que acabara de entrar no processo.
- **Revisão de código:** etapa em que os desenvolvedores revisavam o código que havia sido produzido, a fim de garantir qualidade naquilo que estava sendo construído a partir da perspectiva arquitetural e técnica.
- **Aguardando testes:** etapa onde os desenvolvedores liberavam as funcionalidades criadas para a validação do testador que estava no time.
- **Em testes:** etapa na qual eram testadas as funcionalidades. Eram realizados testes exploratórios, de performance e de integração.

- **Em homologação:** etapa em que a gestora de produto validava se as funcionalidades desenvolvidas atingiam o objetivo de negócio que havia sido definido.
- **Finalizado:** etapa onde as funcionalidades eram publicadas no ambiente que seria futuramente o de produção.

O início: poucas entregas

As 10 primeiras semanas do projeto foram de muito aprendizado para todos os membros



do time.

Figura 5.4: Estudo de caso: dez primeiras semanas do projeto

Nas três primeiras semanas de trabalho, a taxa de entrada de demandas no fluxo era maior do que a taxa de saída. Na média, o time entregava uma história por semana e 3 novas histórias entravam para o fluxo de desenvolvimento.

Neste momento, os desenvolvedores, o testador e a gestora de produto entraram em um consenso de que nenhum novo item entraria para o fluxo de desenvolvimento na semana seguinte (quarta semana). Com tal decisão, o time foi capaz de reduzir o número de histórias em aberto. Ainda sobre o processo, nenhum outro tipo de gargalo era perceptível naquele momento e o time parecia começar a entender qual era o seu limite ideal de entradas e saídas.

Na quinta semana do projeto, o time recebeu a notícia de que o testador entraria de férias por duas semanas e só retornaria aos trabalhos na sétima semana. Mas, afinal, o que o CFD nos diz sobre o que ocorreu com o projeto?

- Ninguém do time de desenvolvimento se prontificou a testar as funcionalidades que estavam sendo criadas, portanto, um estoque de 3 funcionalidades se formou e perdurou até a volta do testador aos trabalhos.
- Na sétima semana, em vez de auxiliar o testador na validação das funcionalidades que haviam sido desenvolvidas durante a sua ausência, os desenvolvedores decidiram trazer novas funcionalidades para o fluxo.
- Da oitava a décima semana, mesmo com os desenvolvedores e o testador fazendo uma força tarefa para testar as funcionalidades, por conta de compromissos organizacionais, a gestora do produto ficou indisponível, o que gerou estoque na etapa de homologação.
- Na décima semana, o time se viu com um total de 11 histórias em aberto e com apenas 7 funcionalidades em produção. Tal cenário gerou uma certa desconfiança por parte do patrocinador do projeto, pois o time atingia a metade do prazo e os números mostravam uma projeção não muito animadora quanto ao que seria entregue até o evento.

O que podemos aprender deste início do projeto?

1. Identifique gargalos o mais cedo possível. Visualize os estoques que podem ser formados ao longo do processo, isto é, crie etapas intermediárias entre as etapas de trabalho. No caso, só ficou explícita a existência de um gargalo na fase de testes, pois o estoque da etapa "Aguardando testes" aumentou ao longo das semanas.
2. Estimule o time a olhar a vazão do processo. No caso, sabendo da ausência do testador, o time (gerente de produto, desenvolvedores e Agile Coach) deveria ter escalado uma pessoa que ficaria responsável pelos testes até o retorno do especialista. Dessa forma, o processo estaria mais estabilizado, isto é, sem gargalos e com uma relação mais equilibrada entre as entradas e saídas.
3. Monitore e controle a taxa de entrada e de saída do processo. Caso você perceba que a taxa de saída do processo é menor do que a taxa de entrada, cuidado, o time estará trabalhando acima da sua capacidade de entrega, o que gerará acúmulos de itens em aberto. Ambientes que lidam com esse tipo de cenário geralmente têm problemas de qualidade, possuem alta rotatividade por conta de estresse e não conseguem garantir uma cadência de entrega (o que gera baixa previsibilidade).
4. Quanto maior o número de demandas em aberto, mais difícil será estabilizar o processo. Crie uma cultura de entregas constantes e incrementais. Os 11 itens que estavam em aberto na décima semana compunham um ativo de funcionalidades que não representava valor para o patrocinador do projeto.

Momento chave: estabilizando o processo

No final da décima semana do projeto, o time realizou uma grande retrospectiva com o objetivo de aumentar as entregas, diminuir os gargalos do processo e aumentar a qualidade do produto que estava sendo desenvolvido. O CFD ilustrado a seguir demonstra o caminho que o time percorreu entre as semanas 10 e 20.

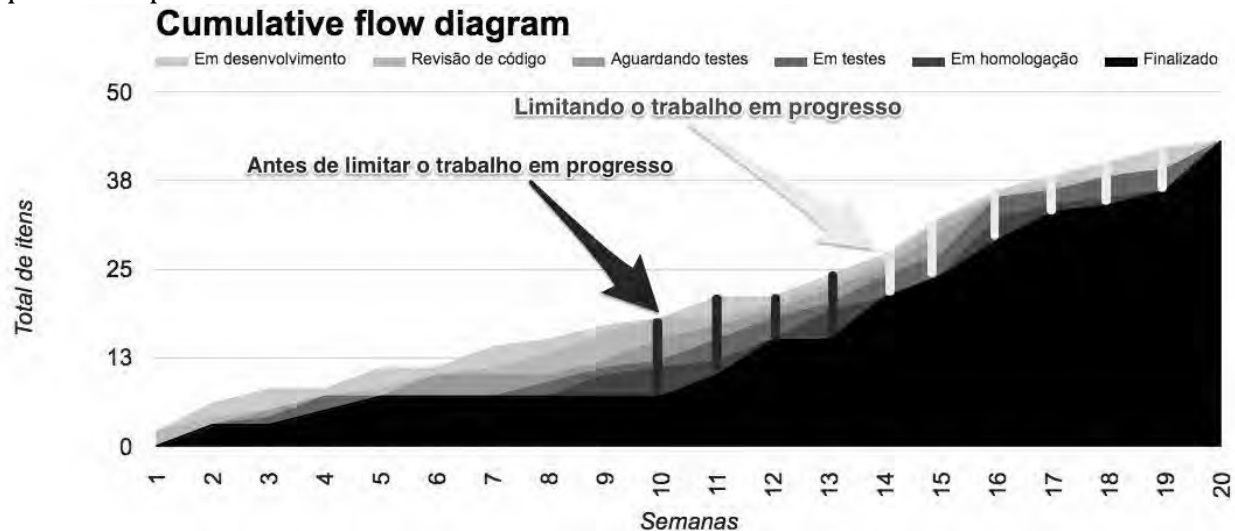


Figura 5.5: Estudo de caso: análise das dez últimas semanas do projeto

Entre as semanas 11 e 12, o time concentrou todos os esforços disponíveis para conseguir colocar em produção o maior número de funcionalidades possível. Para isso, desenvolvedores, testador e gestora de produto focaram em reduzir os gargalos localizados nas etapas de testes e aprovação (representadas por aguardando testes, em testes e em homologação).

A partir da semana 14, o time decidiu definir um limite para o total de itens que estariam em aberto no processo de desenvolvimento. O principal objetivo de tal decisão foi buscar garantir uma cadência na entrega e, principalmente, evitar gargalos nas etapas intermediárias do fluxo de desenvolvimento.

Sabendo que o prazo de entrega se aproximava, após a semana 15, a gestora de produto precisou tomar a decisão de controlar as funcionalidades que entravam no fluxo de desenvolvimento do time. Naquele momento, todos sabiam que os esforços deveriam estar alocados para a finalização da versão que seria apresentada no evento programado para acontecer na semana 20.

Três semanas antes da realização do evento, os desenvolvedores decidiram remover do processo a etapa de revisão de código, pois eles passaram a se organizar em duplas que

mudavam ao longo do dia. Dessa forma, todos participavam ativamente da construção e revisão do que estava sendo produzido.

Por fim, uma força tarefa foi realizada na semana 19 para que as 6 funcionalidades que estavam em aberto pudessem ser finalizadas.

O que podemos aprender com a segunda fase do projeto?

1. Controle o número de demandas que se encontram em aberto no processo. Monitore e limite a quantidade de itens que passam pelo fluxo. Essa é uma dica valiosa para aqueles que, de alguma forma, desejam trazer previsibilidade para o processo de desenvolvimento.
2. Fique atento caso alguma faixa desapareça (etapa do processo sem nenhum tipo de trabalho sendo feito). Caso alguma etapa do processo deixe de fazer sentido, talvez seja o momento de removê-la. No entanto, caso o time passe a pular etapas importantes, como por exemplo aquelas relacionadas à qualidade, cuidado, pois provavelmente as entregas produzidas podem estar repletas de defeitos e gerarão retrabalho no futuro.
3. Utilize o CFD como forma de discutir melhorias no processo junto do time. Dados e visualizações são ótimas maneiras de remover o subjetivismo que pode existir nas discussões sobre como melhorar o processo a partir do que passou.

5.4 Métricas que podem ser extraídas de um CFD

Agora que tivemos uma visão geral sobre a estrutura do CFD, seus atributos e como utilizá-lo a partir da sua visualização, vamos dar nomes e definir algumas métricas que podem ser extraídas a partir das informações do fluxo que foram coletadas, e que às vezes não ficam tão visíveis graficamente.

Work in progress (WIP)

Dado que a linha superior de um CFD representa o total de itens que entraram no processo de desenvolvimento do time, e a linha inferior representa o acumulado de itens que saíram do sistema, podemos concluir que a diferença vertical entre as linhas, em qualquer intervalo de tempo, representará o total de trabalho em progresso (WIP) do time.

Ao longo dos anos, já me deparei com análises que consideravam o tamanho do WIP a partir da diferença dos valores acumulados de etapas intermediárias do processo. Tendo como exemplo o processo do time XPTO, é como se disséssemos que o WIP fosse a diferença entre o acumulado de itens das etapas “Em desenvolvimento” e “Finalizado”.

Caso queiramos analisar o sistema de trabalho como um todo, tal leitura estaria incorreta, pois estaríamos ignorando a primeira etapa do processo (Backlog) que demandou esforço do time para que as demandas pudessem ser definidas e priorizadas.

Voltemos ao exemplo de CFD do time XPTO para entendermos como tais diferenças geram ruído na análise. Imagine que, na terceira semana de trabalho do time, você precisasse compartilhar o total de itens em WIP.

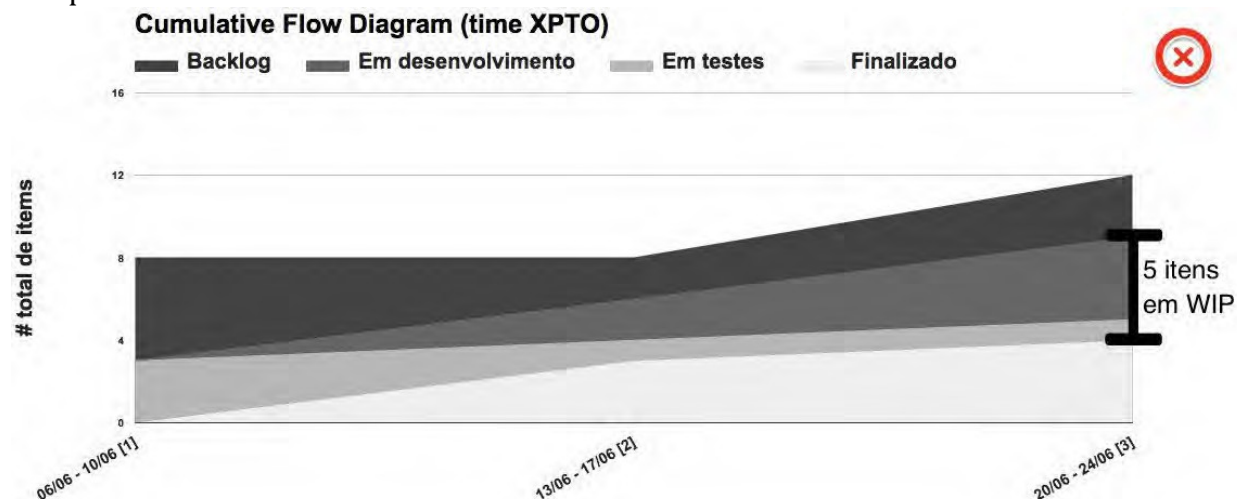


Figura 5.6: Considerando o WIP a partir de etapas intermediárias

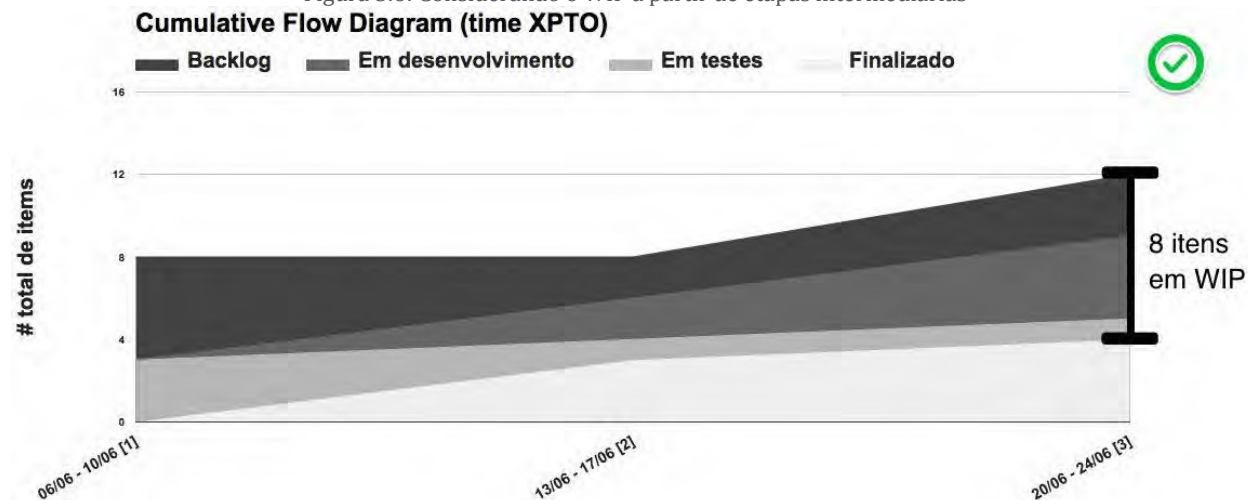


Figura 5.7: Análise do WIP de todo o processo do time XPTO

Analisando as figuras anteriores, observamos uma diferença quando extraímos a métrica de WIP a partir de etapas intermediárias, e quando calculamos a métrica a partir de uma

perspectiva de todo sistema. Na primeira figura, observa-se que o total de WIP no sistema de trabalho do time XPTO é 5 e não 8, afinal, o WIP está sendo medido a partir de etapas intermediárias.

Se você está usando tal abordagem, cuidado, pois é como se estivesse ignorando, no exemplo, o trabalho que foi despendido de análise e refinamento dos 3 itens que representam a diferença entre as etapas de “Backlog” e “Em desenvolvimento”.

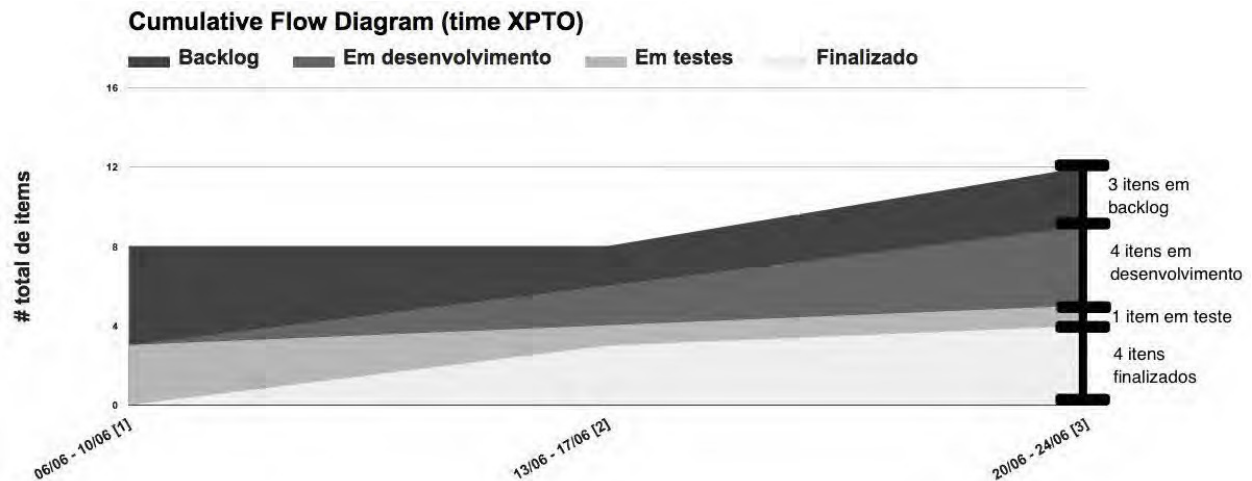


Figura 5.8: Análise do WIP por etapa do processo de desenvolvimento

Ainda sobre WIP, é possível extrair do CFD uma visão de trabalho em progresso em cada uma das etapas do processo do time. A figura anterior ilustra que, na terceira semana, o time XPTO tinha 3 itens sendo tratados na etapa de Backlog, 4 sendo trabalhados em desenvolvimento, 1 sendo testado, e 4 que haviam sido finalizados e publicados em produção.

Analisar o WIP a partir do CFD torna-se algo importante para qualquer pessoa que trabalha com desenvolvimento de software. Desenvolver tal visualização gera a possibilidade do time compreender a quantidade de itens que estão sendo trabalhados neste momento (como número de histórias em andamento na semana ou no *sprint*), *lead time*.

Além disso, também traz uma análise temporal de possíveis gargalos que possam ser formados no fluxo de trabalho como um todo (estoque de WIP só tem aumentado nas últimas semanas ou *sprints*, por exemplo), ou em determinadas etapas do processo — como o número de histórias em teste só tem aumentado, o que tem refletido na não entrega de novas funcionalidades aos usuários.

Em resumo, ter em mãos tal ferramenta auxiliará na compreensão, definição e comunicação dos limites do sistema de trabalho, além da visualização de estoques.

Lead time médio

Além da possibilidade de analisarmos o tamanho do estoque de itens que estão em progresso (WIP) no fluxo de desenvolvimento, o CFD permite calcular, de forma aproximada, o tempo médio necessário para que um item saia do processo (*lead time*).

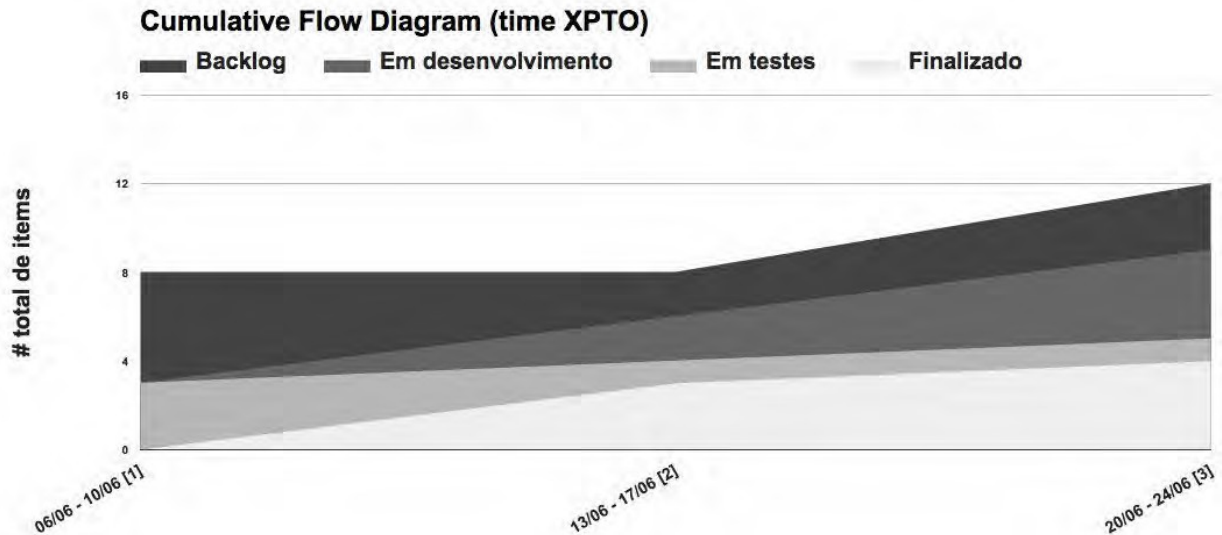


Figura 5.9: CFD do time XPTO na semana 3 do projeto

Para compreender melhor como extrair o *lead time* médio, voltemos ao CFD do time XPTO.

Conforme ilustrado na figura anterior, imagine que a partir da terceira semana do projeto o time já está mais confiante de que o fluxo de trabalho tenha atingido o nível de maturidade desejado por todos.

Sendo assim, para dar visibilidade aos envolvidos (clientes, gestores de produtos etc.) de quantos dias estão sendo necessários para que sejam feitas as entregas, você acaba de receber a tarefa de extrair e mostrar ao time qual é o seu *lead time* médio. Para isso, você precisará realizar os seguintes passos a partir da visualização do CFD — note que o exemplo utilizará dias como unidade de tempo do *lead time*, no entanto, os passos serão os mesmos para qualquer outra unidade, como semanas, meses etc.

1. Defina um ponto na linha inferior do gráfico que corresponderá ao período de tempo que será analisado, e desenhe uma linha vertical. No caso que estamos analisando, tal período de tempo representa a semana 3, que corresponde aos dias 20/06 a 24/06.

2. A partir do ponto definido, desenhe uma linha horizontal da direita para a esquerda até que a linha inferior (etapa de saída do processo) toque a linha superior (etapa de entrada no processo) do CFD. No caso do time XPTO, temos o encontro entre as linhas na semana 1.
3. Dada a intersecção entre as linhas, o *lead time* médio será a diferença entre as datas das linhas superior e inferior. Como o CFD analisado utiliza no eixo horizontal o intervalo de tempo em semanas e desejamos extrair o *lead time* em dias, usaremos os últimos dias das semanas 1 e 3 para a realização do cálculo. Subtraindo 24/06 por 10/06, temos 14 dias.
4. Como último passo, adicione 1 ao resultado da subtração. No caso do time XPTO, temos que o *lead time* médio será de 15 dias. Mas, por que adicionar 01 ao resultado? Eu sempre aconselho a adição de uma "unidade de tempo" (no exemplo que estamos tratando, seriam dias), dado que o tempo mais curto que um item pode demorar para ser concluído é uma unidade.

Imagine que um determinado item de trabalho comece e termine no mesmo dia (por exemplo, 24/06), qual será o seu *lead time*? Se fôssemos apenas subtrair 24/06 de 24/06, teríamos um *lead time* de 0 dias. Particularmente não concordo com esse tipo de visão, afinal, se considerássemos 0, estaríamos passando a ideia de que nenhum esforço foi necessário para a entrega deste item.

O passo a passo descrito anteriormente pode ser resumido na seguinte fórmula:

$$\text{leadtime médio} = (\text{data linha superior} - \text{data linha inferior}) + 1$$

Figura

5.10:

Fórmula para o cálculo do lead time médio

Voltando à análise do nosso exemplo, a partir de agora, o time poderá inferir que os novos itens que entrarão no fluxo de desenvolvimento levarão, aproximadamente, 15 dias para serem entregues. Tal dado pode ser útil em situações nas quais o time precise estimar datas de entregas para novas demandas.

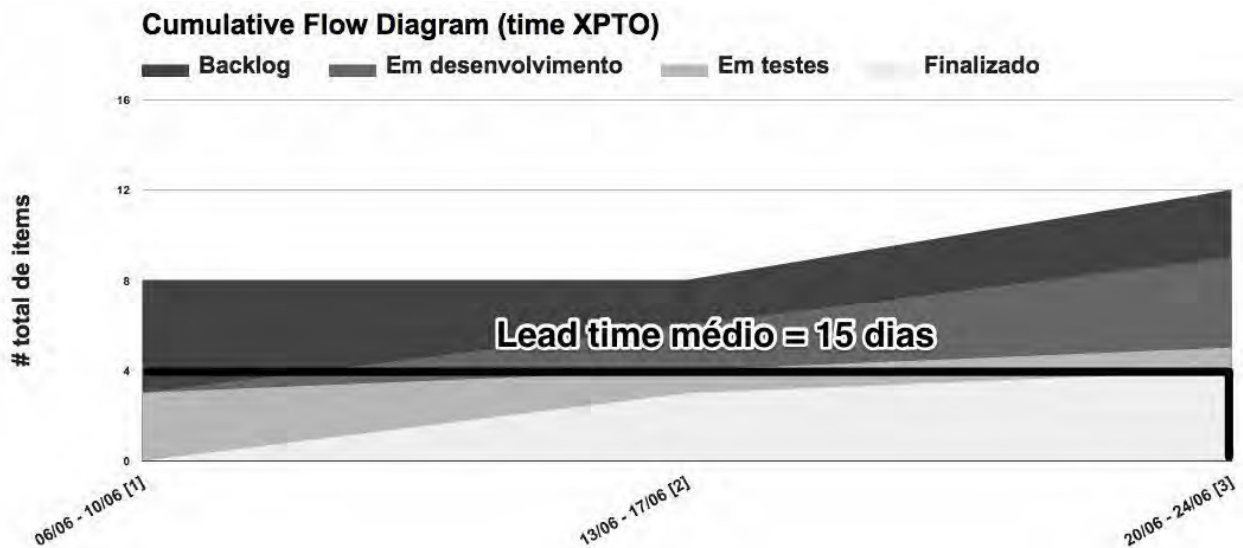


Figura 5.11: Exemplo de cálculo de lead time médio

É importante salientar que a métrica de *lead time* médio não deve ser considerada como uma meta, um comportamento padrão ou uma medida absoluta. Afinal de contas, pela natureza incerta que é desenvolver software, esse tipo de informação deve ser usado apenas como indicativo de tendência.

Throughput médio

Outra métrica que pode ser extraída da visualização do CFD é o *throughput* médio do processo, isto é, a taxa de vazão média de itens do fluxo de trabalho de determinado time.

Em algum momento de um projeto de desenvolvimento de software, o time precisará expor aos *stakeholders* qual tem sido, em média, a quantidade de funcionalidades entregues por um determinado período de tempo para que possam ser feitas projeções de quando o projeto será finalizado.

Para ilustrar como calcular o *throughput* médio, voltemos ao exemplo do time XPTO. Imagine que o time acaba de finalizar a quarta semana de trabalho no projeto e o gerente de produtos deseja saber quantos itens estão sendo entregues por semana, para se ter uma ideia da vazão do time. Como poderíamos calcular o *throughput* médio?

1. Determine um período de tempo. No exemplo, determinamos que vamos calcular o *throughput* médio das quatro primeiras semanas do projeto.

2. Descubra a "ascensão" da linha inferior, isto é, quantos itens saíram do sistema de trabalho naquele período de tempo. No caso do time XPTO, temos que 4 histórias foram entregues até a semana 4.
3. Por fim, divida a quantidade de itens que saíram do processo pelo período analisado. No caso do time XPTO temos que, em média, é entregue uma história por semana (resultado da divisão de 4 histórias por 4 semanas).

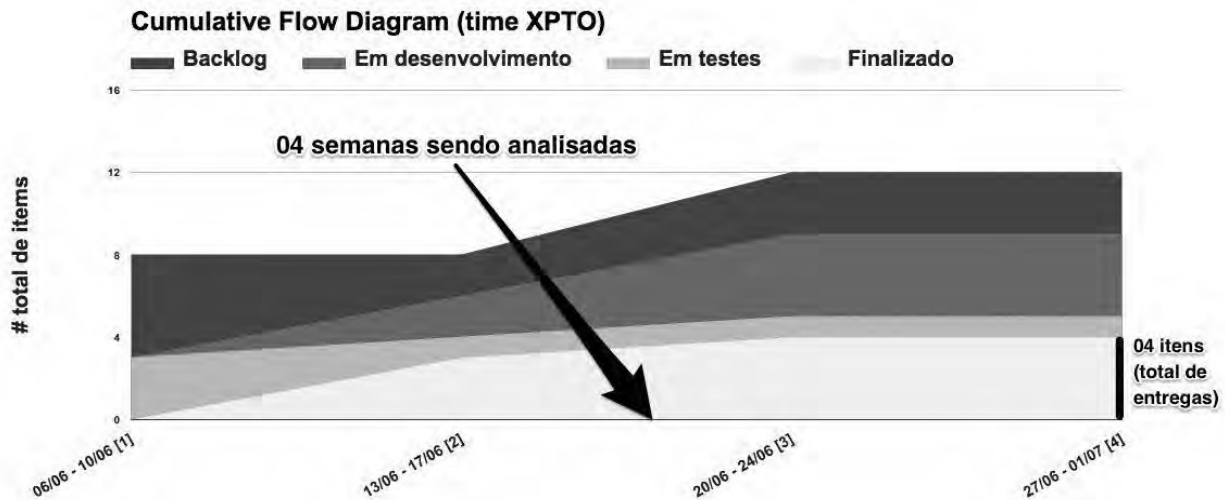


Figura 5.12: Exemplo de cálculo do throughput médio

O *throughput* médio nos dá a possibilidade de saber se o time está conseguindo garantir uma cadência de entregas ao longo do tempo.

É válido ressaltar que, como qualquer métrica, tal informação será útil e fará sentido dentro um contexto e para uma determinada unidade de análise (no caso, o time). Portanto, muito cuidado ao comparar o *throughput* médio de diferentes times.

Uma dica que costumo compartilhar sobre a análise do *throughput* é a de calcular a média a partir de um período de tempo mais recente.

Mudanças que aconteceram no fluxo de trabalho do time podem ter melhorado a taxa de vazão. E caso você esteja considerando momentos nos quais a vazão foi baixa (como produção por problemas de ambiente ou indefinições), a métrica média será influenciada por tal comportamento que, no entanto, dado o novo contexto, já não tem mais sentido.

No fundo, a ideia é remover da medida qualquer tipo de ruído que aconteceu no passado e que pode estar prejudicando uma visão mais realista de qual tem sido a cadência de entrega do time.

Análogo ao *throughput*, é possível extrair do CFD a taxa média de entrada de itens no processo de desenvolvimento. Os passos necessários para o cálculo desta métrica são os mesmos do *throughput*, exceto pelo fato de que queremos avaliar o número de entradas e não mais saídas no fluxo de trabalho.

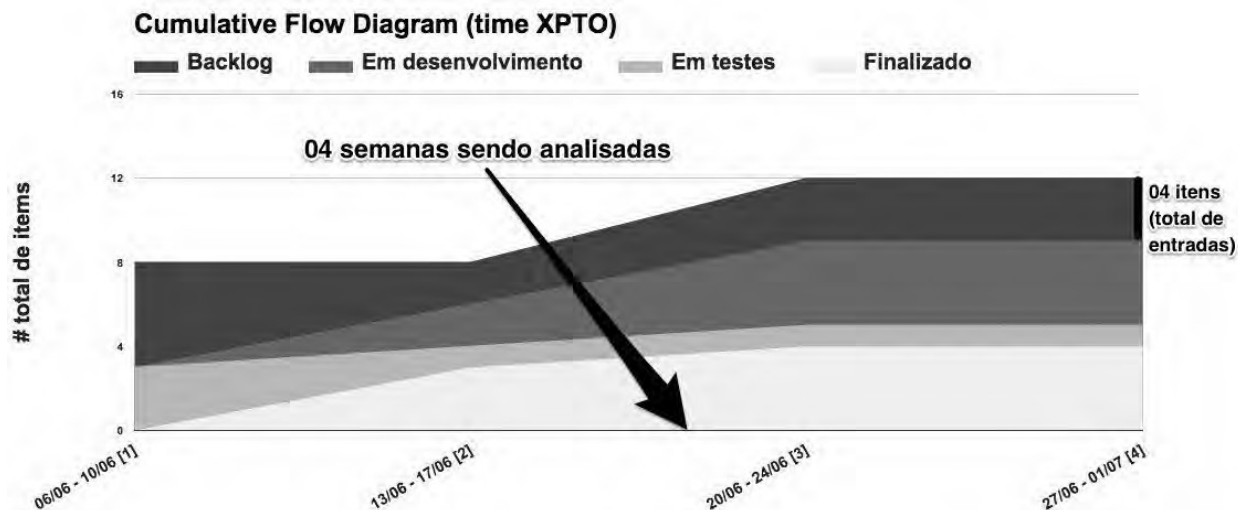
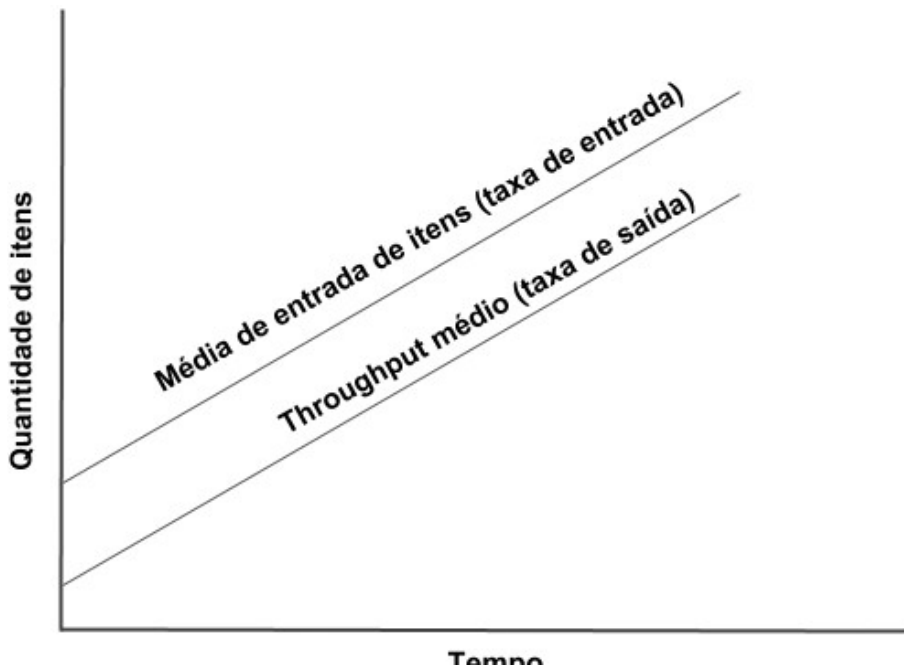


Figura 5.13: Exemplo de cálculo do taxa de entrada média

Utilizando como exemplo o time XPTO, conforme ilustrado na figura anterior, é possível dizer que a partir da primeira semana, 1 novo item entrou para o fluxo de desenvolvimento por semana, dado que a diferença entre o total de itens que passaram pela etapa de entrada do fluxo (Backlog) foi 4 (8 na primeira semana e 12 na última semana) e o total de semanas analisadas foi 4.

Analisar a taxa de entrada e o *throughput* médio juntos pode ser útil para compreender se um time tem conseguido manter uma constância de entrega (saída) a partir do que tem sido definido como necessidade (entrada).

Figura
Relação



5.14:
entre

Caso o time tenha um comportamento parecido com o exibido nessa figura, ele terá dado o primeiro e, provavelmente, o mais importante passo para equilibrar o que entra e o que sai do processo de desenvolvimento. Buscar esse tipo de comportamento é algo fundamental para que haja um bom alinhamento de expectativas entre quem solicita uma funcionalidade de software e quem a constrói.

A pergunta que você pode estar se fazendo neste momento é: como eu consigo estimular o time a ter o tipo de comportamento ilustrado no último gráfico?

A melhor maneira de equilibrar entradas e saídas no desenvolvimento de projetos de software é garantindo uma quantidade constante de trabalho em progresso (WIP). Entenda como WIP constante um limite global de todo o sistema — o time pode ter, no máximo, 3 histórias em progresso, por exemplo —, ou uma determinação de quantos itens poderão ser trabalhados em cada etapa do fluxo — o time pode ter, no máximo, 2 histórias na etapa de teste).

Limitar o sistema tem se mostrado uma forma de garantir que os times passem a trabalhar em um ambiente com baixa variabilidade. Isso auxilia na previsibilidade das projeções de entrega.

5.5 Recapitulando

A geração da visualização do CFD pode ser feita manualmente (dará um pouco mais de trabalho, porém sua atualização pode ser realizada em cerimônias como o *daily*), por meio de planilhas eletrônicas ou até mesmo a partir de *softwares* especializados em gestão de projetos. Independente da forma como o time gerará o gráfico, o importante é dar visibilidade da evolução do fluxo para todos que estiverem envolvidos com o processo de desenvolvimento.

Difícilmente, você verá CFDs que sigam todas as dicas que compartilhei neste capítulo. Muito provavelmente você se deparará com visualizações bem diferentes das citadas nos exemplos, no entanto, não tenha medo.

Agora que você conhece os elementos do gráfico e as respectivas métricas (WIP, *lead time* médio, *throughput* médio e taxa de entrada), você saberá extrair ótimas ideias e, principalmente, saberá o que analisar.

Lembre-se: o gráfico só indicará problemas no processo. Todo o trabalho de análise e melhoria caberá a você e ao seu o time.

Uma forma de extrair informações úteis do CFD é através de perguntas, como por exemplo:

- O que está acontecendo com o nosso fluxo?
- Nossas métricas estão mostrando um bom ou mau caminho?
- Se estamos caminhando bem, como podemos manter?
- Se estamos com problemas, que tipo de mudança precisamos fazer para que o fluxo melhore?

O CFD garantirá responder as perguntas certas de forma antecipada e sugerirá as ações necessárias para aumentar a previsibilidade do processo.

Caso tenha interesse de saber mais sobre o assunto, não deixe de ler o excelente artigo escrito por Pawel Brodzinski (2013), ou o livro *Actionable Agile Metrics for Predictability* do Daniel Vacanti (2015).

5.6 Referências

BRODZINSKI, P. *Cumulative Flow Diagram*. Jul. 2013. Disponível em: <http://brodzinski.com/2013/07/cumulative-flow-diagram.html>.

VACANTI, D. *Actionable Agile Metrics for Predictability: an introduction*. Daniel S. Vacanti, Inc., 2015.