



INSTITUTO DE GESTÃO E TECNOLOGIA
DA INFORMAÇÃO

Práticas para Operação do Produto

Carla Regina de Moraes Tonon Ezaki

2022

Práticas para Operação do Produto

Bootcamp: Product Owner

Carla Regina de Moraes Tonon Ezaki

© Copyright do Instituto de Gestão e Tecnologia da Informação.

Todos os direitos reservados.

Sumário

| | | |
|-------------|---|----|
| Capítulo 1. | Planning and Reviews | 5 |
| | Sprints | 6 |
| | Refinamento | 7 |
| | Estimativas | 8 |
| | Sprint Planning | 10 |
| | Sprint Review | 11 |
| Capítulo 2. | Daily, retrospectivas e demais cerimônias | 12 |
| | Daily | 12 |
| | Retrospectiva | 13 |
| | Fun retrospectives | 14 |
| Capítulo 3. | ATDD..... | 16 |
| Capítulo 4. | Kaizen..... | 20 |
| | Passos para implantar o Kaizen | 20 |
| | Utilizando o Kaizen no desenvolvimento de produtos digitais | 22 |
| Capítulo 5. | Acompanhamento de deploys..... | 23 |
| Capítulo 6. | Entrega de release | 24 |
| | Release Notes | 24 |
| | Comunicação..... | 24 |
| Capítulo 7. | Métricas | 25 |
| | Tipos de métricas..... | 25 |

| | |
|--|----|
| Algumas métricas utilizadas no mercado | 25 |
| O que considerar ao criar métricas..... | 27 |
| Capítulo 8. Captando Feedback dos clientes | 28 |
| Feedbacks indiretos | 28 |
| Utilizando o próprio produto para captar feedback..... | 28 |
| Perguntando diretamente ao cliente | 29 |
| Escuta ativa e comunicação não violenta | 31 |
| Referências..... | 33 |

Capítulo 1. Planning and Reviews

Uma parte importante da vida de um *product owner* é justamente as cerimônias do framework *scrum* dos quais faz parte e muitas vezes conduz.

O *scrum* foi criado em 1993 por Jeff Sutherland e Ken Schwaber, com o objetivo de trazer mais visibilidade de agilidade nas entregas de software, com entregas menores e validações mais rápidas. Esse framework se baseia em ciclos curtos com feedback rápido, interação constante entre os envolvidos com a intenção de entregar o máximo valor no menor prazo possível com alta qualidade.

O título do livro escrito pelo criador do framework é um tanto quanto audacioso: “A arte de fazer o dobro na metade do tempo”, esse foi justamente o que foi realizado em um dos cases mais conhecidos: o software *Sentinel* utilizado pelo FBI, que estava com 10 anos de realização sem chegar ao fim; ao utilizar *scrum* o software foi finalizado em 18 meses. Uau! E como se faz uma mágica dessas?

O grande pulo do gato aqui foi gestão a vista, comunicação e principalmente a definição do que gerava mais valor e fazer as entregas pequenas desses valores com rápidos feedbacks, para que na nova entrega pudessem ajustar o que foi necessário; além de todos estarem engajados juntos no sucesso da entrega, independente se são desenvolvedores, testadores, operação etc., o importante é a entrega de valor acontecer com agilidade e qualidade, da mesma maneira onde todos os jogadores de rugby se unem na jogada *scrum* para o bem do time:

Figura 1 – Jogada conhecida como Scrum no rugby.

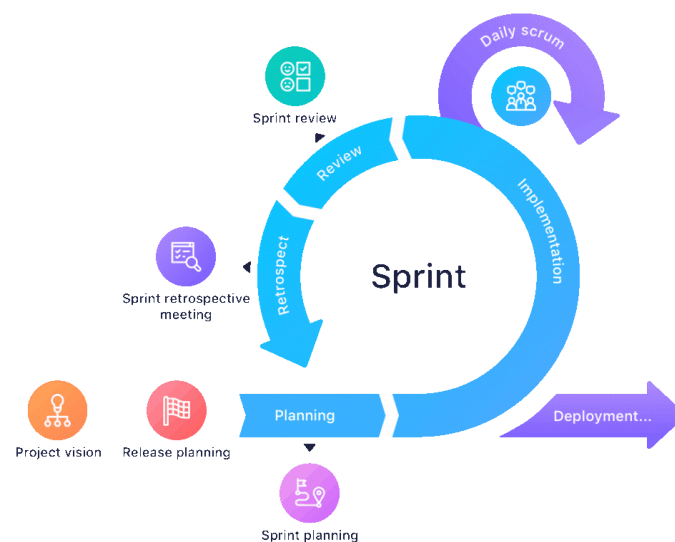


Fonte: [Scrum para iniciantes](#).

Sprints

Uma das grandes premissas do *scrum* é o ciclo curto entre a interação e a entrega, mas como isso é medido? No *scrum*, o tempo do ciclo de desenvolvimento até a entrega é chamado de Sprint, ele pode ter desde uma semana até um mês, não sendo recomendável ultrapassar um mês, e o mais usual é um ciclo de 15 dias.

Figura 2 – Elementos de uma sprint.



Fonte: [Scrum: entenda o que esse framework ágil pode fazer pelo seu time!](#)

A contagem do tempo precisa ser fixa, ou seja, se for definido que será de 2 semanas, não pode ser mudado nem no meio da sprint e nem nas próximas sprints, conforme Sutherland (2014, p.79): “Você precisa ser consistente: estabelecer um ritmo de trabalho no qual as pessoas saibam o que pode ser feito em determinado período. Em geral a quantidade surpreende”.

A contagem da *sprint* inicia após a *planning* (que será verificada abaixo), onde é definido a meta e quais atividades serão necessárias para o atingimento dessa meta. Após essa definição não poderão ser acrescentadas nem retiradas tarefas (pelo menos não sem o consentimento do time e do *Product Owner*).

Refinamento

Antes de entrar na *planning meeting* é importante refinar as demandas priorizadas, essa etapa é muito importante pois revisamos os requisitos, critérios de aceitação, tamanho das histórias (é muito comum ter quebra de histórias em novas, cancelamento de histórias que não fazem mais sentido e até surgir novas histórias).

⚠ Atenção!

É muito comum equipes ágeis utilizarem o termo ***grooming*** para a etapa de refinamento, porém esse termo não deve ser utilizado, pois ele refere-se ao aliciamento de menores por pedófilos (cuidado com o ilegal grooming), ao invés de utilizar esse termo utilize ***Refinement***, se quiser continuar com termos em inglês, ou **refinamento** para o português.

Uma das técnicas mais utilizadas deriva do XP e é conhecida como reunião dos 3 amigos. Nela encontram-se o *product owner*, pelo menos um desenvolvedor e um testador, e neste momento serão levantados todos os requisitos, dúvidas e cenários possíveis das demandas priorizadas, também serão realizadas as estimativas para cada uma delas.

Estimativas

Bacana, já foi definido o backlog, priorizadas as histórias conforme o que gera maior valor ao cliente, já foi definido também o tamanho que durará a sprint, mas como saber o que cabe no sprint? Qual o tamanho daquela história? Tudo bem, as histórias seguem um padrão INVEST e, portanto, são pequenas, mas o quão pequenas elas são? Quais precisam de maior ou menor esforço do time de desenvolvimento, levam maior tempo para testar etc.?

O fato é que o ser humano é muito ruim para estimar prazos, nosso cérebro pode distorcer a percepção de tempo, onde cada indivíduo percebe o tempo de formas distintas, se estiver cansado é ainda pior. Em 2020, os neurocientistas Masamichi J. Hayashi e Richard B. Ivry publicaram um estudo no *Journal of Neuroscience*, onde puderam constatar que após vários estímulos (visuais e sonoros) havia menor atividade de neurônios sensíveis ao tempo. (HAYASHI, 2020; IVRY, 2020).

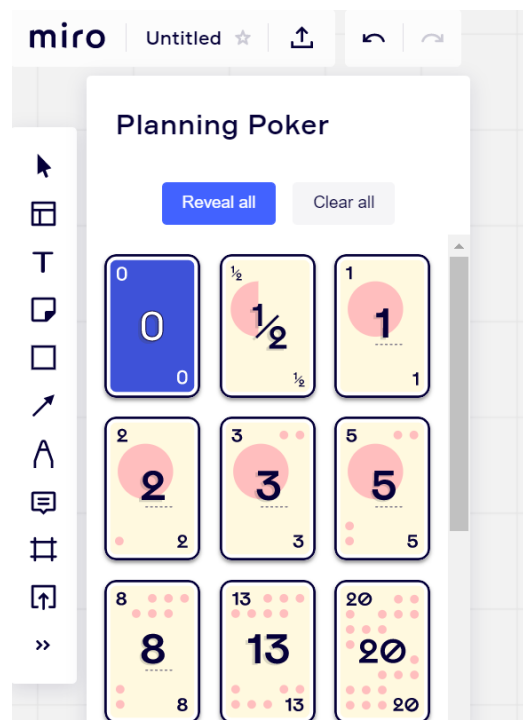
Bom, se humanos não são bons em definir prazos, como é possível realizar estimativas e fazer planejamentos confiáveis? Já que o cérebro pode distorcer o tempo, ele é muito bom em outra coisa, em comparações! Exemplo, se identificamos que um elefante é grande, um rato é pequeno, qual tamanho de um gato? Está mais para o tamanho do rato ou do elefante? Com base nas comparações de tamanho, foram criadas várias técnicas de estimativas de demandas:

- **Pontos por História:** para que possamos analisar com clareza a diferença entre as histórias analisadas, a métrica utilizada deverá ter um range grande entre si, então essa técnica utiliza a sequência de Fibonacci como régua. Sim, aquela mesmo que aprendemos na escola, que é considerada a matemática da natureza onde, a partir do seu terceiro termo, é calculado pela soma dos dois antecessores, a sequência de Fibonacci é infinita e começa dessa forma: 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, mas para o efeito dos *story*

points podemos considerar : $1/2$, 2, 3, 5,8,13. E como devem imaginar, utilizaremos essa escala para pontuar o tamanho das histórias, sendo $1/2$ a menor e 13 a maior.

- **Planning Poker:** para pontuar cada história baseado na escala de Fibonacci, uma técnica lúdica, simples e objetiva é o *planning poker*. Nessa técnica cada integrante do time recebe um conjunto de cartas, contendo os valores da sequência de Fibonacci, e após um refinamento do item, cada um joga a carta que acredita ser correspondente ao item e após todos discutem para chegar num consenso a respeito do item. Essa técnica pode ser utilizada em times remotos também e algumas ferramentas como o miro já possuem *templates* para o *planning poker*:

Figura 3 – Planning poker no Miro®.





Sprint Planning

Após a priorização, refinamento e estimativa, já é possível planejar o que será entregue na próxima sprint. Esse planejamento é feito em uma reunião que tem um *timebox* máximo de 8 horas para sprints de 30 dias, onde serão definidas a meta da sprint e quais atividades pretende-se entregar. Maravilha, mas como saber quantas histórias cabem no sprint?

É muito importante que algumas métricas ágeis sejam acompanhadas para conseguir saber qual a velocidade e a capacidade de entrega do time, e desta forma poder calcular a previsibilidade das entregas. Algumas métricas mais usadas são:

- **Lead Time:** nessa métrica conseguimos verificar qual a velocidade do time, pois ele mede a quantidade de tempo de um item desde a sua concepção até a sua entrega.
- **Throughput:** no *Throughput* é avaliado a capacidade do time, pois mede a quantidade de vazão das entregas, ou seja, quantos itens foram entregues naquela medida de tempo.

Com esses dados em mãos é possível estimar capacidade e velocidade, e assim saber o que cabe na *sprint* para então planejar o que será entregue nessa sprint. Após a finalização da *sprint planning* é dado o start da sprint onde os itens escolhidos começam a ser trabalhados.

Sprint Review

Ao final da sprint é necessário validar o que foi entregue, se está de acordo com o que foi planejado e se a meta da sprint foi atendida.

Esta é uma reunião com o time de desenvolvimento, o *product owner* e pode haver também o envolvimento dos *stakeholders* importantes para o assunto discutido. Nessa reunião são apresentadas ao time as demandas realizadas, e eles verificarão se os critérios acordados foram devidamente cumpridos, dando o aceite dessas atividades.

Atenção!

Não há problema algum em termos alguma apresentação em slides para contextualizar algum ponto, porém não deve ser o ponto principal e não pode ser o único! O objetivo dessa reunião é de vermos software funcionando e, na prática, avaliar o que foi realizado.

A reunião de review é muito importante para acompanhamento do que foi realizado, para a transparência da equipe e para o feedback das demandas, pois dessa forma podemos reajustar o que poderá melhorar em termos de produto nas próximas sprints.

Capítulo 2. Daily, retrospectivas e demais cerimônias

No capítulo anterior vimos as cerimônias do *scrum*, que devem ser conduzidas pelo *product owner* e facilitadas pelo *scrum master*, porém existem mais cerimônias do *scrum* que, mesmo estando no *scrum guide*, não necessitam obrigatoriamente da presença do *product owner*, este como parte do time *scrum* deve sim participar. Vamos conhecê-las

Daily

Como o nome diz, essa é uma reunião diária onde o time responde a três perguntas:

- O que fiz ontem?
- O que farei hoje?
- Tenho algum impedimento?

Essa reunião tem um *timebox* de 15 min e deve ser bem simples e objetiva, caso alguém tenha alguma dúvida ou que precise de ajuda, deverá ser realizada após a *daily* pelos envolvidos.

O *scrum guide* não determina a presença do *product owner* nessa reunião, trazendo o mesmo como um opcional, porém é importante sim a presença do PO por vários motivos:

- O *product owner* é parte do time e deve estar presente e ciente do que acontece na sua equipe.
- Dúvidas de negócio podem surgir quando o time está em desenvolvimento, ou mesmo o desenvolvedor pode ter compreendido algo de forma errônea e ao

descrever o que está desenvolvendo já podemos ajustar o foco, e não apenas na review.

- Fortalece o sentimento de time, onde todos estão focados no objetivo em comum.

Mas em tempos remotos, como manter o *timebox* e o fato de ser uma reunião em pé? Essa questão é bem particular de cada time e de cada necessidade, e em muitos casos não é um problema grave estarmos todos sentados (afinal muitos estão em casa) e ultrapassar um pouco o *timebox*, pois no trabalho remoto não há tanta interação quanto no trabalho presencial e pode ser o momento de compartilhamento entre o time, mas não pode ser demais para não atrapalhar o andamento das atividades, portanto o time ideal é muito subjetivo e o time como um todo deverá medir e se adaptar.

Retrospectiva

A retrospectiva é aquele momento em que muitos não gostam de participar. De fato, a forma como é conduzida é muito importante, pois pode se tornar uma reunião muito pesada, com críticas e “caça às bruxas”.

Essa cerimonia se conduzida de forma leve e com o objetivo claro de melhoria é de suma importância para que as ações de todos tenham sempre a melhoria contínua como objetivo.

Basicamente, a reunião é conduzida pelo *scrum master* e todos do time (sim todos, o próprio *scrum master*, o *product owner*, desenvolvedores, testadores, suporte, documentadores, designer) enfim, todos que participaram da sprint, levantam os pontos positivos, negativos e qual os planos de ação para a próxima sprint. É importante destacar o princípio básico das retrospectivas escrito por Norm Kerth (2013, tradução nossa):

“Independentemente do que descobrirmos, devemos entender e realmente acreditar que todos fizeram o melhor trabalho que puderam, dado o que sabiam na época, suas habilidades e aptidões, os recursos disponíveis e a situação em questão.”

Partindo dessa premissa a parte mais importante e que deverá ser gasto mais energia são nos planos de ação, o que faremos para melhorar os pontos positivos, para que vire uma reunião de soluções e não uma reunião de levantamento de problemas.

Fun retrospectives

Para que a reunião seja lúdica e com o objetivo de atender a premissa de ser uma reunião de soluções para as problemáticas e para que o time reconheça seus esforços e realizações nas sprints, existem técnicas onde podemos pontuar essas coisas de formas mais divertidas. Algumas técnicas são:

- **Retrospectiva da Escola de Samba:** nessa prática o time representa a escola de samba e o objetivo é ganhar nota 10 em todos os quesitos do desfile, então é bacana trazer uma imagem de um carro alegórico ou mesmo de um desfile que tenha sentido para o time. Depois de definir o samba, o enredo do time é pontuado: os pontos fortes do desfile (sprint), o que se atravessou no samba (pontos negativos) e como podemos manter a harmonia nota 10. Essa dinâmica é bacana para pontuar momentos de espírito de equipe, onde todos estão em harmonia com o time e o objetivo da sprint.
- **Retrospectiva das Cinco Tias:** nessa dinâmica o time pontua tudo o que foi bacana na TIA CALMA, o que foi uau na TIA LEGRA, o que não foi legal na TIA FLIGE, o que acordamos a não mais fazer na TIA FASTA e os planos de ação na TIA NIMA. Essa técnica é bem lúdica, deverão para isso ser desenhadas

imagens de “tias”, com as devidas expressões e colaborar para o time celebrar as conquistas e reafirmar os acordos.

- **Retrospectiva: Carro de corrida:** nessa dinâmica, o time desenha o carro de corrida que deverá representar o próprio time e podem incluir as características do time nele, e aí vão definindo: as pedras que são os obstáculos e/ou problemas que o time enfrentou na sprint; os declives são as coisas que ajudaram o time a aumentar a velocidade; a linha de chegada é o valor entregue pelo time na sprint e a garagem representa os planos de ação que o time irá fazer nas próximas sprints.

Capítulo 3. ATDD

O Desenvolvimento Orientado a Testes de Aceitação (*Acceptance Test-Driven Development* – ATDD), é uma técnica colaborativa (pode ser feita na reunião dos três amigos vista anteriormente) onde são levantados todos os requisitos e critérios de aceitação das histórias.

O ATDD possui quatro etapas:

- Debater (Discuss): essa etapa é muito importante e todos os riscos devem ser mensurados, de negócio, técnicos, performance etc. Também são levantados os exemplos para formação de cenários de teste.
- Refinar: após os requisitos estarem devidamente levantados é a fase de refinar as histórias e escrevê-las no formato *Gherkin*. Esse formato possui a seguinte sintaxe:

Funcionalidade: <título da funcionalidade>

Texto Adicional: <Informações relevantes de negócio>

Cenário 1: <Uma determinada situação do negócio>

Dado que <pré-condição>

Quando <uma ação é realizada>

E <outra ação>

Então <um resultado é realizado>

Cenário 2: <Outra determinada situação do negócio>

Por exemplo:

Funcionalidade: Visualização de extrato bancário

Quando: estiver no meu aplicativo bancário

Como: titular da conta

Eu quero: poder consultar os movimentos financeiros da minha conta

Cenário 1: Visualizar extrato bancário

Dado que eu esteja na tela inicial

Quando eu acesso o menu Saldos e Extratos

Então escolho a opção extratos

E eu preencho as datas de início e fim do período a ser consultado

E eu seleciono visualizar

Então o sistema exibe todas as operações financeiras relacionadas ao período informado.

Cenário 2: Data incorreta

Dado que eu esteja na tela inicial

Quando eu acesso o menu Saldos e Extratos

Então escolho a opção extratos

E eu preencho as datas de início e fim do período a ser consultado de forma inválida

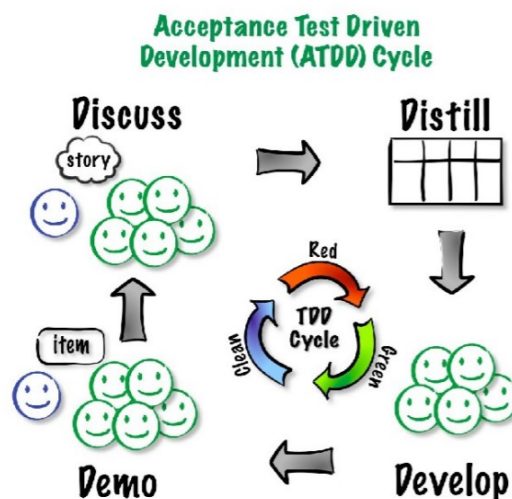
E eu seleciono visualizar

Então o sistema exibe uma mensagem de texto informando que as datas de consulta são inválidas

- Implementar: nesse ponto desenvolvedores e analistas de QA já podem iniciar o desenvolvimento que terá como o TDD, onde é realizado primeiro o teste automatizado, para só então o desenvolvimento do software de fato.
- Apresentar: por fim os resultados são apresentados após todos os testes serem finalizados e executados com sucesso, porém a reunião de review ainda deverá acontecer para demonstração do que foi realizado e pode ser apresentado, inclusive, o resultado dos testes automatizados se for pertinente para a reunião.

James Shore criou um modelo do ciclo de ATDD, ficando desta forma:

Figura 4 – Ciclo de vida do ATDD.



Fonte: <https://tasafo.org/wp-content/uploads/2010/01/dirigindo-o-desenvolvimento-com-testes-atdd-e-tdd.pdf>.

Os testes são incrementais, ou seja, a cada nova história são incluídos novos testes, e sempre que são executados (nos *deploys* por exemplo), todos os testes são executados, os novos e os antigos; isso garante que novos códigos não impactarão nos antigos, diminuindo a chance de bugs.

Capítulo 4. Kaizen

O Kaizen surgiu no Japão após a Segunda Guerra Mundial, com o objetivo de reconstruir o país. Kaizen em kanji (改善) significa mudança para melhor ou como conhecemos no Brasil, melhoria contínua.

O kaizen possui alguns princípios básicos:

- Aprenda com a prática, aja ao invés de ficar só na teoria ou num planejamento sem fim.
- Elimine desperdícios, produza o necessário sem excessos.
- Engaje a todos em todas as tarefas.
- Não são necessários altos investimentos.
- Deve ser agnóstico.
- A comunicação deverá ser fluida e transparente.
- Priorização de itens mais importantes.
- Como o nome diz, a melhoria deverá ser constante.
- Pessoas sempre em primeiro lugar.

Perceberam a semelhança com os princípios do manifesto ágil? Vamos ver como essa filosofia pode auxiliar na construção de produtos digitais.

Passos para implantar o Kaizen

O Kaizen divide-se em três fases:

1. **Preparação:** nesta fase são identificados a oportunidade da mudança, as pessoas envolvidas, ferramentas utilizadas e o papel de cada um.
2. **Evento:** fase da mão na massa propriamente dita, aqui todos os envolvidos se dedicarão 100% nas suas atividades.

3. **Follow up:** fase em que as mudanças criadas serão incorporadas na rotina.

Já os passos são 7 e andam por toda as 3 fases já descritas, como é explicado no infográfico:

Figura 5 – Ciclos do Kaizen.



Fonte: <https://caetreinamentos.com.br/blog/melhoria-continua/o-que-e-kaizen/>.

Utilizando o Kaizen no desenvolvimento de produtos digitais

O Kaizen pode ser usado como uma prática para melhoria contínua de inúmeros problemas, que muitas vezes são negligenciados no dia a dia do desenvolvimento de produtos digitais como por exemplo:

- Nos planos de ação levantados em retrospectivas.
- Nas melhorias levantadas nos feedbacks em reviews.
- Para corrigir débitos técnicos.
- Para novas ideias.

Todo o trabalho acaba se beneficiando com a filosofia simples e direta do Kaizen, o começo pode ser lento, porém a melhoria é consistente e contínua.

Capítulo 5. Acompanhamento de deploys

Bom, a sprint finalizou, as demandas foram aprovadas então acaba aqui o trabalho do *product owner*, certo? Negativo! Um trabalho entregue é aquele que está em produção, ou seja, que está na mão do cliente.

Mas espere aí, vou ter que virar *devops*? Mas não sei nada sobre criar *branches* e *deploys*, e agora? Calma! A parte técnica continua com os desenvolvedores e com os *devops*, porém o acompanhamento desses *deploys* é muito importante principalmente se tivermos demandas com teste A/B por exemplo.

OK, mas vamos voltar um pouquinho, o que é um *deploy* e por que eu preciso me preocupar com ele? O *deploy* em inglês significa implantar e em produtos digitais significa que o que foi feito deverá ser colocado no servidor, para que o cliente tenha acesso a ele, e é aqui que entra a função do *product owner*. Após aprovar o que foi produzido na *review*, o *product owner* deverá definir o que será incluído no *deploy* para ser entregue para o cliente, se serão para todos os clientes ou não (uma customização em especial ou realização de testes A/B) etc.

É importante também garantir que os testes automatizados sejam executados e que todos os serviços estejam ativos e com os índices saudáveis (existem métricas para isso que são implantados geralmente pelo time de *devops*).

Além desses pontos, fica a cargo do *product owner* a comunicação com o cliente: se a aplicação é *on premisses* (a instalação é a cargo do cliente) deverá ser informado quando o pacote estará disponível, caso seja SaaS (a instalação é em um servidor remoto e a manutenção do produto fica a cargo do fornecedor), a comunicação é ainda mais emergencial, pois deve ser informado ao cliente a data e horário que a atualização do sistema dele será realizado, e a previsão de quanto tempo os serviços poderão ficar indisponíveis.

Capítulo 6. Entrega de release

As releases são basicamente os lançamentos das versões do produto e é muito importante, como visto no capítulo anterior, fazer o planejamento dessas releases, o que será entregue, para quem, como e com qual frequência.

A cadência das entregas é um assunto muito importante, pois além da organização do time também é necessário a organização do cliente para que possam atualizar o ambiente (em caso de *on-premises* por exemplo), e mesmo no *saas*, os clientes precisam se programar para caso haja alguma indisponibilidade, mudança de comportamentos do sistema, de layout etc.

Release Notes

O release notes é basicamente uma lista de itens que fazem parte daquela versão do produto e deve incluir tanto novas funcionalidades quanto correções do sistema. Essa lista de itens deve ser concisa, porém deve expressar tudo o que foi incluído e modificado, para que o cliente possa realizar a conferência dos itens.

Um fato que deve ser observado é que o release notes deve ser publicado ou enviado ao cliente antes do *deploy*, para que seja consultado antes da atualização e qualquer possível problema possa ser verificado, e se for o caso atrasar o *deploy*. Um exemplo disso é por exemplo um novo *layout* de nota fiscal, sendo incluído antes de um prazo final para a entrega da nota fiscal para a receita federal.

Comunicação

Após o release notes estar publicado e o *deploy* finalizado, é hora de comunicar ao cliente as novidades que este pode passar a utilizar, um grande aliado nesse quesito é o time de marketing, além do time de suporte.

Capítulo 7. Métricas

Entrega da release feita com sucesso, mas como saber se a release foi de fato um sucesso? Uma forma de acompanhar se a entrega foi com valor é medindo, porém o que medir? E como? Números apenas não dizem muitas coisas e corremos o risco de estar inundados por números, mas sem informações relevantes, afinal a métrica tem um propósito de trazer informações para tomadas de decisão.

Tipos de métricas

- Temos quatro tipos de métricas que precisam ser consideradas:
 - **Métricas chaves de adequação (KPI):** os indicadores a serem acompanhados pelo negócio devem ser os mesmos que o cliente escolheu como seus critérios de adequação. Cuidado com o impulso de medir tudo! Escolha o que faz sentido medir.
 - **Métricas de Saúde:** precisam de acompanhamento e devem gerar alertas a partir do momento em que saem de uma faixa “saudável”.
 - **Direcionadores de Melhoria:** uma vez que um alerta é gerado por algo ter saído da faixa “saudável”, é preciso estabelecer um objetivo de melhoria do status.
 - **Métricas de Vaidade:** indicadores que se mostram positivos, mas que não contribuem de fato com o que define o sucesso do negócio.

Algumas métricas utilizadas no mercado

- **Customer Satisfaction Score (CSAT):** é uma pesquisa direta, geralmente realizada após uma entrega de release, onde é questionado o nível de

satisfação do cliente para aquela entrega. Vai de muito insatisfeito para muito satisfeito.

- **Cycle time:** mede a velocidade do time, onde verificamos o tempo que uma demanda demora para ficar pronta desde o início do desenvolvimento até a entrega.
- **Lead time:** o Lead time é parecido com o cycle time, porém mede a demanda desde a concepção da ideia até a sua entrega.
- **Release Frequency:** é a frequência com que as releases são liberadas para o cliente.
- **Technical Debt:** é a quantidade de débitos técnicos que foram gerados, isso impacta na qualidade e na manutenibilidade do software.
- **NPS:** refere-se a Net Promoter Score e mede o quanto o cliente está satisfeito com o produto, indicaria ele sendo um promotor ou se seria um detrator (além de não indicar não aconselharia).
- **Churning:** esse é bem claro, é a quantidade de clientes que cancelam o produto ou deixam de usá-lo. Pode ser cancelamento total ou parcial (por exemplo, o cliente contratou 100 licenças e passa a usar apenas 50, gerando uma diminuição de receita).
- **Métricas de usabilidade quantitativas:** Nessa podemos medir o tempo gasto em uma tarefa feita pelo cliente, a taxa de sucesso nessa tarefa, a facilidade de utilização, performance de resposta, engajamento do cliente no produto, o quanto textos, menus, ícones são intuitivos ao cliente;

- **Métricas de usabilidade qualitativas:** aqui pode ser medido o sentimento do cliente ao usar o produto.

O que considerar ao criar métricas

Quando iniciamos a contar com as métricas e dados para a tomada de decisão, podemos cair em algumas armadilhas de querer medir tudo, analisar todos os dados sem refinar, criar métricas de vaidade etc. Essas dicas podem facilitar a criação e análise dos dados coletados:

1. **Priorize os dados críticos:** indicadores com valores ruins são minas de ouro para avaliação e priorização de demandas e mudanças. Comece melhorando os itens com piores valores.
2. **Crie planos de ação:** para dados ruins crie planos de ação práticos e esquematizados para resolver a questão.
3. **Revise os indicadores de tempos em tempos:** um indicador pode não ser mais válido após um tempo de uso e deverá ser alterado ou mesmo inutilizado.
4. **Não tente medir tudo:** priorize os indicadores mais importantes.
5. **Caça as bruxas:** cuidado em utilizar indicadores para a caça as bruxas, a ideia não é encontrar culpados e sim identificar pontos de melhoria no processo e/ou produto.

Capítulo 8. Captando Feedback dos clientes

Saber o que o cliente pensa é muito importante, mas é igualmente importante saber como captar esses feedbacks que podem ser diretos ou indiretos e até mesmo dentro do próprio produto.

Feedbacks indiretos

Um setor da empresa que mais recebe feedbacks de clientes é o suporte técnico, o *customer care* e a equipe comercial. Os clientes que reclamam de problemas possuem dúvidas de utilização e devem ser observados com atenção, e planejado em conjunto com os times de suporte e *customer care* ações para diminuir a incidência de insatisfação dos clientes e da quantidade de chamados, podendo ser:

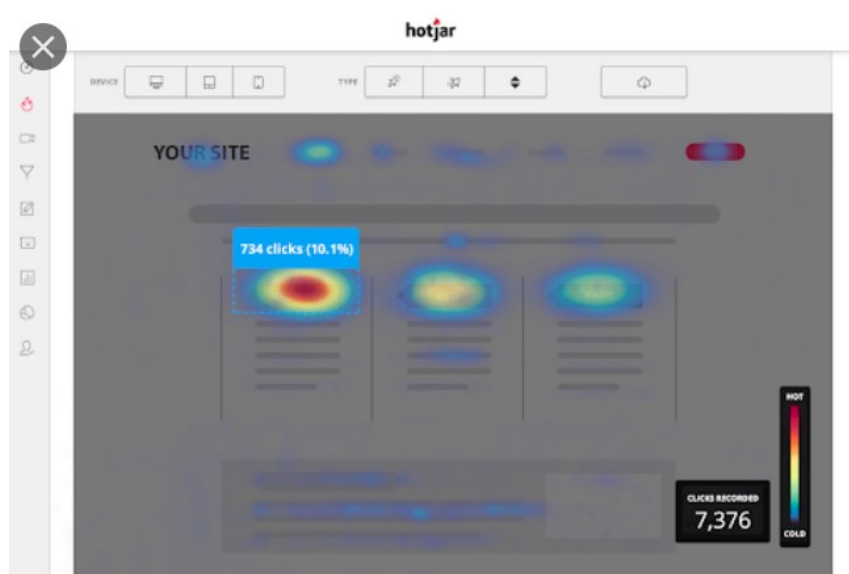
- Uma força tarefa para diminuir o estoque de manutenções e correções quando essas possam ser muito grandes e principalmente de erros graves e críticos.
- Melhoria na documentação e treinamentos dos sistemas para os times de atendimento, parceiros, consultores aos próprios clientes.
- Melhoria na usabilidade de funcionalidades que mais geram chamados de dúvidas.
- Análise de itens solicitados pelos clientes aos canais de vendas, afinal se foram muito pedidos é alguma função que os clientes necessitam e a falta destes podem ser motivos para o não fechamento de uma venda e a perda do cliente.
- Avaliação dos motivos de *churn* pelos clientes e sendo de ordem financeira, pensar em planos de ação para, em conjunto com equipes de vendas e canais, avaliar a possibilidade de novos pacotes mais básicos por exemplo.

Utilizando o próprio produto para captar feedback

Através de métricas é possível avaliar se o cliente está utilizando e como ele está utilizando o produto. Por exemplo, o cliente pode estar acessando uma determinada tela por um fluxo muito complexo e que este poderia estar usando um atalho, ou esteja pedindo uma funcionalidade já existente que ele não sabe como usar.

Existem algumas ferramentas que incluídas no produto geram um mapa de calor e que podem ser analisados para saber como o cliente acessa e utiliza certas funcionalidades com maior ou menor frequência etc. Assim poderemos planejar ações para gerar mais conhecimento do cliente sobre funcionalidade pouco usadas, ou até mesmo melhorias em usabilidade por exemplo. Um exemplo dessa funcionalidade é o hotjar:

Figura 6 – Mapa de calor gerado pelo hotjar.



Fonte: <https://www.hotjar.com/>.

Perguntando diretamente ao cliente

A forma mais clara de obter feedback é pedindo para o cliente! Envio de pesquisas de satisfação ou mesmo entrevistas com este nos dão muitos insights, porém é necessário ter os devidos cuidados:

- **Utilize uma metodologia clara para identificar os dados:** se utilizar um método para cada cliente, não conseguirá realizar os devidos mapeamentos de melhorias e planos de ação.
- **Não faça perguntas amplas demais:** tente direcionar para o que precisa verificar, se fizer uma pergunta muito ampla poderá ter respostas vagas e não vai conseguir os insights necessários.
- **Não faça muitas pesquisas em pouco tempo com os mesmos clientes:** Se encher o cliente com muitas pesquisas, estes poderão se aborrecer, e com os mesmos clientes sempre poderá ter pesquisas viciadas, ou seja, terá sempre a mesma visão.
- **Utilize a comunicação não-violenta:** quando o cliente estiver muito insatisfeito ele muitas vezes poderá se exaltar, uma postura acolhedora com escuta ativa e comunicação não-violenta poderá facilitar a conversa, pois o cliente não está insatisfeito de forma pessoal, mas sim com a situação em que está! O nosso papel é ouvi-lo e resolver o problema do cliente da melhor maneira possível mesmo que este se exalte.
- **Pesquisa por escrito:** quando enviar uma pesquisa por e-mail por exemplo, deixe claro o objetivo da pesquisa e qual o ganho que terá com ela.
- **De resposta posterior ao feedback:** muitas vezes o cliente nos dá um feedback na intenção de que uma ação seja tomada, seja por uma insatisfação ou melhoria que este deseje, é importante repassar um status do que está sendo ou será feito para atender ao desejo do cliente, e quando alguma coisa não será feita com o motivo para ele.

Escuta ativa e comunicação não violenta

Muitas vezes a forma como falamos e ouvimos é mais importante que a mensagem em si. Todos gostamos de ser ouvidos de verdade, que o nosso interlocutor demonstre interesse na nossa fala, respeito e empatia, e como trabalhamos o tempo todo com pessoas, essas características são fundamentais ao trabalho do *product owner* seja com o cliente, sua equipe, *stakeholders* e outras equipes.

A escuta ativa é basicamente ouvir de fato o que o outro está compartilhando, tendo um verdadeiro interesse no assunto, o que não significa ouvir calado, mas estando totalmente envolvido com a conversa, participando de fato dela. E como podemos desenvolver a escuta ativa? Vejamos algumas dicas:

- **Não se distraia:** ao conversar com alguém, preste atenção a pessoa e a conversa, sem olhar o celular, relógio etc.
- **Faça perguntas:** pergunte sobre o assunto da conversa, tenha curiosidade verdadeira no assunto.
- **Mantenha a mente calma:** use de técnicas de ioga, meditação, atividade física para que a sua mente esteja calma e “limpa”, sem ansiedade ou outros assuntos na sua cabeça, que caso esteja cheia e não permita a sua dedicação total a conversa, é melhor remarcar.
- **Tenha empatia:** ouça sem julgamentos e tente calçar os sapatos do outro. E se fosse com você, como estaria se sentindo?

A comunicação não violenta tem por base o ouvir e ser ouvido, com gentileza e empatia para que a comunicação flua de forma facilitada e o entendimento seja correto pelas partes, em especial quando há conflitos. Essa se baseia em:

- **Observar:** a comunicação não se dá apenas com palavras, a linguagem corporal, expressões podem falar mais do que as palavras propriamente ditas, devemos ouvir, prestar atenção com clareza e sem pré-julgamentos.
- **Sentimentos:** devemos entender o sentimento envolvido nessa conversa, o que estamos e o que a pessoa está sentindo, estando abertos e empáticos, a comunicação será mais fluida e verdadeira.
- **Identificar as necessidades:** a partir dos sentimentos podemos verificar quais as necessidades do interlocutor e assim poderemos ter clareza de como podemos saná-las, se pudermos.
- **Nada de ordens:** quando necessitar, peça as coisas. Não há necessidade de impor ordens, afinal não somos maiores que ninguém, não é?

Com essas ferramentas devidamente exercitadas, as comunicações serão cada vez mais fluidas e assertivas, e poderemos obter insights fantásticos até das situações mais difíceis.

Referências

HAYASHI, Masamichi J; IVRY, Richard B. Duration Selectivity in Right Parietal Cortex Reflects the Subjective Experience of Time. *The Journal of Neuroscience*, Berkley, n. 30, sept. 2020, p. 7749–775. Disponível em: <https://www.jneurosci.org/content/jneuro/40/40/7749.full.pdf>. Acesso em: 11 mar. 2022.

KERTH, Norman L. *Project Retrospectives: A Handbook for Team Review*. UK: Dorset House, 2001.

MUNIZ, Antonio; et al. *Jornada ágil de qualidade: aplique práticas no início do ciclo para potencializar a implantação contínua de software com qualidade*. Rio de Janeiro: Brasport, 2019.

MUNIZ, Antonio; et al. *Jornada ágil do produto: unindo práticas e frameworks para capacitar donos de produto (product owners) e gerentes de produto (product managers) com fluxo de valor entregue ao cliente*. Rio de Janeiro: Brasport, 2020.

SUTHERLAND, Jeff. *Scrum: a arte de fazer o dobro do trabalho na metade do tempo*. São Paulo: Leya. 2014.