SOMOS TODOS T.I.

SCRUM

Método Ágil

CRIADO POR: ANDRÉ OLÍMPIO

Produzido e publicado por:

SOMOS TODOS T.I.

Por: André Olímpio

Copyright (C) 2025
Todos os direitos estão reservados

somostodosti.com.br

Este material pode ser compartilhado desde que ele não seja modificado, mantendo os direitos e créditos intactos.
Não é permitida a distribuição para obtenção de cadastros por outro que não a Somos Todos T.I.
Se este material estiver disponível para download por outro site ou serviço que não esteja vinculado à Somos Todos T.I. por gentileza, denuncie através do .e-mail: contato@somostodosti.com.br ou pelas redes sociais.

Quem sou eu?

Olá, eu me chamo André Olímpio, sou nascido na cidade de São José dos Campos, interior do estado de São Paulo. Sou professor de Tecnologia da Informação desde 2005, já ministrei aulas para ensino médio técnico, graduação e pós-graduação.
Sou bacharel em Ciência da Computação, pós graduado em Engenharia de Sistemas de Software, pós graduado em Administração de Banco de Dados, pós graduado em Docência do Ensino Superior, pós graduado em Metodologia do Ensino à Distância, pós graduado em Gestão de Projetos e também pós graduado em Gestão Empresarial.

Também sou um fã incondicional de cultura nerd, futebol e música. Além disso, sou casado e pai de dois filhos lindos.
Após bons anos de experiência em sala de aula, decidi que deveria dar uma guinada na minha vida e investir em uma nova empreitada, o empreendedorismo digital. Foi assim que tive a ideia de expandir os meus horizontes para fora da sala de aula e investir na Internet. Assim, criei o site e o canal no YouTube "Somos Todos T.I.", além de também disponibilizar o conteúdo de minhas aulas em uma plataforma de aprendizado à distância, a EADBOX (HEROSPARK).
Conhecimento precisa ser repassado e nunca guardado para si. Foi com este pensando que criei uma estrutura de trabalho na qual consigo atingir uma quantidade maior de pessoas interessadas em aprender mais sobre esta área sensacional e extremamente desafiadora que é a Tecnologia da Informação.
E nunca se esqueçam que no final das contas...

Somos Todos T.I.

SCRUM
Método ÁGIL

Sobre o SCRUM

Trata-se de um framework com o qual as pessoas podem resolver problemas complexos e adaptáveis, enquanto entregam produtos de forma produtiva e criativa e com o maior valor possível.

É um dos métodos ágeis mais utilizados no mercado atualmente.

O termo é originário de uma jogada realiza em uma partida de rugby, na qual os jogadores precisam se organizar e se ajudar para obter a posse da bola.

Nesta jogada, o trabalho em equipe, a organização e a disciplina são fundamentais para que o objetivo seja devidamente alcançado.

Pode ser definido como: *formação ordenada.*

Fonte: Federação Internacional de Rugby
www.world.rugby

Manifesto Ágil

Este manifesto obedece a 12 princípios. São eles:

1. A maior prioridade é satisfazer o solicitante com a entrega contínua e adiantada de software com valor agregado durante o processo de desenvolvimento do mesmo.

2. Mudanças de requisitos são bem-vindas, mesmo que sejam tardiamente. Processos ágeis tem por característica tirar vantagem das mudanças visando vantagens competitivas para o solicitante do software.

3. Entregar em pequenos intervalos de tempo (normalmente medido em semanas), o software em funcionamento parcial.

4. Solicitante e desenvolvedor devem trabalhar em conjunto diariamente, durante todo processo de desenvolvimento do software.

5. Os desenvolvedores devem estar sempre motivados. Ambiente favorável, suporte adequado e confiança entre as partes são itens fundamentais para motivação de qualquer grupo de trabalho envolvido no processo.

6. Mesmo utilizando ferramentas diferentes para comunicação entre as partes envolvidas (stakeholders), a conversa face a face nunca deve ser descartada.

7. O software funcionando, mesmo que parcialmente, é a meta principal de conclusão de cada etapa do processo de desenvolvimento.

8. Utilizar metodologia ágil, promove um desenvolvimento sustentável. Assim desenvolvedores e usuários devem ser capazes de manter um ritmo constante de comunicação e atendimento das necessidades mútuas.

9. Contínua atenção à excelência técnica, codificação, atendimento de requisitos e um bom design aumentam a agilidadeda equipe de desenvolvimento.

10. A simplicidade é essencial. O menos é sempre mais.

11. As melhores arquiteturas, requisitos e designs emergem de equipes auto-organizáveis.

12. Em intervalos regulares, a equipe deve refletir sobre como se tornar mais eficaz e refinar seu comportamento durante o trabalho.

Fonte: The Home of Scrum
www.scrum.org

Apesar de ter sido criado originalmente para gerenciamento no processo de desenvolvimento de sistemas de software, o Scrum pode ser utilizado em qualquer tipo de projeto, contanto que haja um certo grau de complexidade, já que ele se baseia na teoria de controle empírico de processos.

Um processo empírico é aquele onde diversos imprevistos ocorrem e alguma atitude para resolvê-los deve ser tomada. Desta forma, o processo é melhorado à medida em que o Scrum é aplicado, já que as coisas não ocorrem sempre da mesma forma.

Diferente de uma linha de montagem, onde cada passo pode ser previsto e monitorado, em projetos de desenvolvimento de softwares isso se torna impossível. Por este motivo se justifica a necessidade de utilizar um processo empírico.

Papéis e responsabilidades

O Scrum apresenta poucos papéis, mas muito bem definidos. Além disso, as responsabilidades dentro do projeto são distribuidas de forma diferente do que em metodologias tradicionais.

Esse papéis são:

- *Product Owner*
- *Scrum Master*
- *Team Developer*

Um Times Scrum (TS) deve ser auto-organizado e multifuncional. Por ser auto-organizado, o TS deve decidir qual é a melhor forma de realizar seu trabalho, ao invés de serem direcionados por pessoas de fora do time.

Um time multifuncional possui capacidades que o torna autossuficiente. Este modelo foi desenhado para otimizar a flexibilidade, criatividade e produtividade deste time.

Product Owner (PO)

Responsável por maximizar os valores do software e todos resultados obtidos pela equipe de desenvolvimento.

É o responsável por gerenciar a lista de pendências do produto (product backlog) do software a ser desenvolvido.

Suas principais incumbências consistem em:

- Expressar claramente para a equipe de desenvolvimento todos os itens contidos no product backlog.
- Ordenar os itens desta lista de modo a atingir objetivos e metas da melhor forma possível.

Fonte: The Home of Scrum
www.scrum.org

Team Developer

É a denominação utilizada para definir um membro da equipe de desenvolvimento.

A equipe deve ser auto-organizada.

Cada integrante da equipe tem autonomia para criar e gerenciar sua própria maneira de agir durante a execução de um ciclo de trabalho, podendo inclusive escolher quais são as tarefas que irá realizar em cada ciclo.

Um Team Developer deve assumir todas as habilidades necessárias para transformar as pendências do product backlog em funcionalidades entregáveis (deliverables).

Não há títulos entre os membros da equipe, todos são developers (desenvolvedores).

Scrum Master (SM)

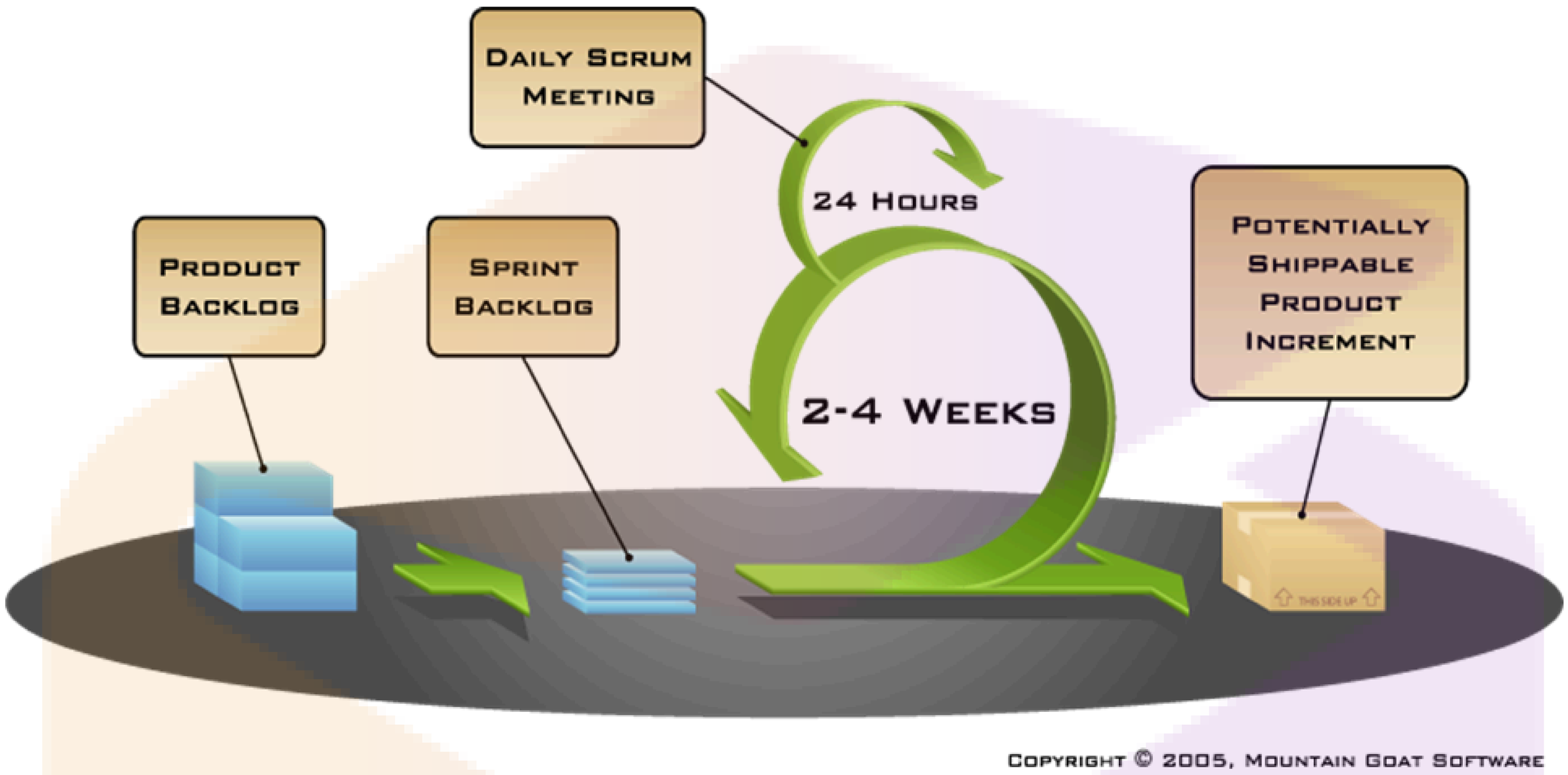
É o intermediário entre o as partes envolvidas no projeto: *PO, os Team Developers e o solicitante do software.*

Deve atender aos interesses de todas essas partes, principalmente da equipe de desenvolvimento.

É o responsável por gerenciar toda equipe de desenvolvimento de forma técnica, produtiva e efetiva. Pode-se dizer que o Scrum Master é o capitão do time.

O SM é a pessoa que mais conhece Scrum e seu papel não é fácil, já que orientar o PO na criação e ordenação do Product Backlog, garantir que as regras de Scrum estejam sendo cumpridas e seus valores seguidos, ser o facilitador nos eventos de Scrum e também ajudar a remover impedimentos que o time venha a enfrentar. E tudo isso deve ser feito sem fazer uso de qualquer autoridade.

A arquitetura do Scrum



A sprint é o coração do Scrum. É definida como um curto período de tempo de algumas semanas no qual o software será desenvolvido e entregue, de forma parcial, para o solicitante.

Uma nova sprint se inicia imediatamente após o término de outra.

A repetição (iteração) é fundamental para o sucesso de uma sprint.

O processo pode conter quantas sprints forem necessárias.

Fonte: The Home of Scrum
www.scrum.org

Durante a sprint:

- Nenhuma mudança que afete o objetivo da sprint deverá ser realizada.
- A composição e as metas da equipe de desenvolvimento deverão permanecer constantes.
- O escopo deve ser esclarecido e renegociado entre o PO e a equipe de desenvolvimento à medida que o conhecimento evolua.
- Todos os envolvidos devem seguir à risca as normas pregadas pelo Scrum.
- Não devem ser feitas quaisquer alterações que afetem a Meta da Sprint.
- O escopo pode ser renegociado entre o Development Team e o Product Owner conforme o conhecimento evolua durante o desenvolvimento, novas tecnologias sejam criadas.

Standup Meeting

Conhecida também como "reunião em pé", ou Daily Scrum Meeting. Deve ser realizada todos os dias, com a equipe em pé, com duração de 15 a 20 minutos.

São realizadas três perguntas entre os membros da equipe:

- *O que foi feito ontem?*
- *O que será feito hoje?*
- *O que impede que as ações sejam realizadas?*

Product Backlog

É uma lista ordenada contendo todas as funcionalidades que o software deverá realizar após ser concluído.

É a única fonte de requisitos para quaisquer mudanças a serem realizadas no software.

O product backlog é dinâmico e está sempre pronto para sofrer alterações. Estas alterações são necessárias para que o software seja útil, apropriado e competitivo, atendendo assim às necessidades do solicitante.

Sprint Backlog

É um conjunto de itens do product backlog que são selecionados para serem desenvolvidos na sprint atual, com o intuito único entregar algo de valor ao final desta etapa.

É o guia de referência do time para as ações a serem realizadas no desenvolvimento do software durante cada sprint.

Deve ser exposto a todos integrantes do time e também aos demais envolvidos no processo de desenvolvimento de software.

É uma visão ampla, em tempo real, do trabalho realizado pela equipe de desenvolvimento.

Assim, somente o Scrum Master, em conformidade com o time de desenvolvimento, pode efetuar modificações neste documento no decorrer do sprint.

À medida que um novo trabalho é requerido pelo solicitante ou pelo PO, este deve ser prontamente adicionado ao sprint backlog.

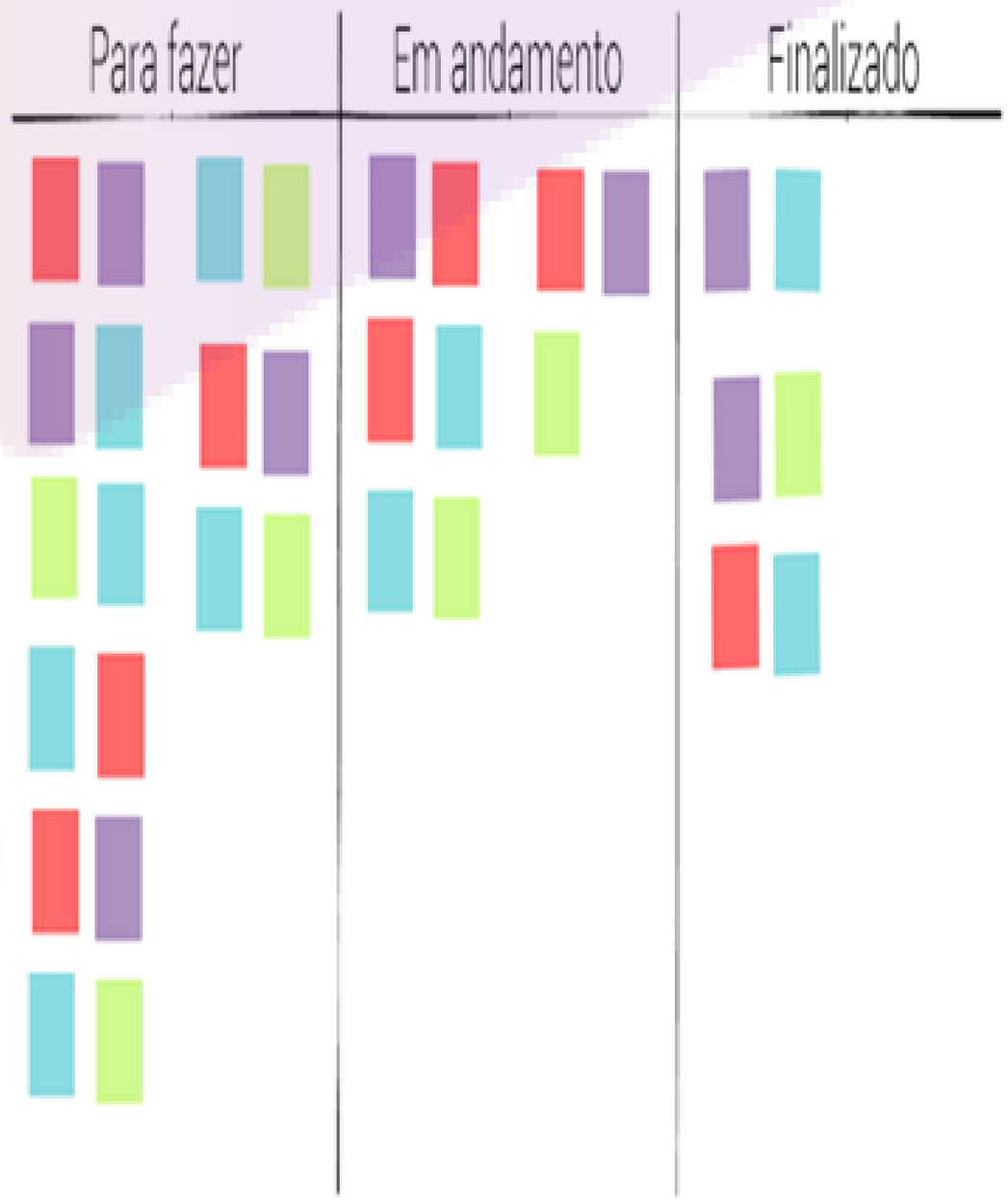
Uma das formas mais comuns de criação de um Sprint Backlog utilizadas atualmente é o Kanban.

Kanban

É um termo de origem japonesa que significa cartão ou sinalização.

Este conceito está relacionado com a utilização de cartões para indicar o andamento dos fluxos de produção em empresas de fabricação em série.

Em um processo de desenvolvimento de software, o kanban representa, em tempo real, a execução de cada ação realizada por cada integrante do Time Scrum.



Eventos do Scrum

Sprint Planning Meeting

Trata-se de uam reunião realizada no início de cada sprint, onde o Time Scrum planeja quais são as tarefas a serem realizadas nesta sprint.

A Sprint Planning Meeting é dividida em duas partes sendo que cada um destas partes responde às seguintes perguntas:

- *O que será entregue como resultado nesta sprint?*
- *Como faremos para entregar o resultado nesta sprint?*

Na primeira parte da Sprint Planning Meeting, o time de desenvolvimento faz uma previsão das funcionalidades que serão desenvolvidas durante a sprint.

Na segunda parte, o time de desenvolvimento decide como transformará os itens selecionados em resultado durante a sprint.

Sprint Review

Trata-se de uma cerimônia realizada ao final da sprint, na qual podem participar qualquer pessoa que esteja interessada no produto final.

Embora essa cerimônia também seja utilizada para demonstrar as funcionalidades implementadas durante a sprint, seu principal motivo é o de inspecionar o que a equipe de desenvolvimento produziu e colher opiniões e/ou impressões dos presentes para, caso seja necessário, adaptar o plano para a sprint seguinte. Então, o foco desta cerimônia é aprimorar o produto.

Nesta cerimônia, o PO valida ou não a sprint, de acordo com a meta que tenha sido acordada com o time de desenvolvimento durante a *Sprint Planning Meeting*. O time discute sobre a sprint, sobre o que correu bem e os problemas enfrentados além das soluções encontradas.

Sprint Retrospective

Este é o último evento de uma sprint que ocorre imediatamente após a Sprint Review.

Participam desta reunião todos os membros do Time Scrum, tendo como foco é o aprimoramento do processo, a interação entre os integrantes do time, as práticas e ferramentas utilizadas, o que funcionou e o que precisa ser melhorado na próxima sprint.

Identificar problemas somente não é suficiente. Assim, é importante também identificar medidas a serem tomadas para a melhoria do processo para as próximas sprints.

Como o objetivo é a melhoria do processo, os integrantes do TS não devem encarar alguma crítica como algo pessoal.

Artefatos do Scrum

Product Increment

Trata-se do resultado obtido pelas ações realizadas pelo Time Scrum alcançado ao final de cada sprint. Assim o time de desenvolvimento entrega um incremento, uma parte do produto final.

Este é um dos conceitos principais de Scrum e para o time de desenvolvimento é importante entender que este Incremento deve ser algo potencialmente entregável, o cliente pode optar por colocar imediatamente em produção.

Definition of Done (DoD)

Trata-se do ponto que, mediante as práticas adotadas pelo TS, é considerado a definição do que é uma funcionalidade pronta. No mínimo, a Definition of Done deve conter “potencialmente entregável”.

Uma funcionalidade somente é considerada pronta se tiver passado por todas as etapas definidas pelo time de desenvolvimento. E todo o TS deve entender o que significa pronto.

Caso uma funcionalidade que não esteja pronta ao final da sprint deve retornar ao Product Backlog para que seja incluída em uma próxima sprint.

Burndown Chart

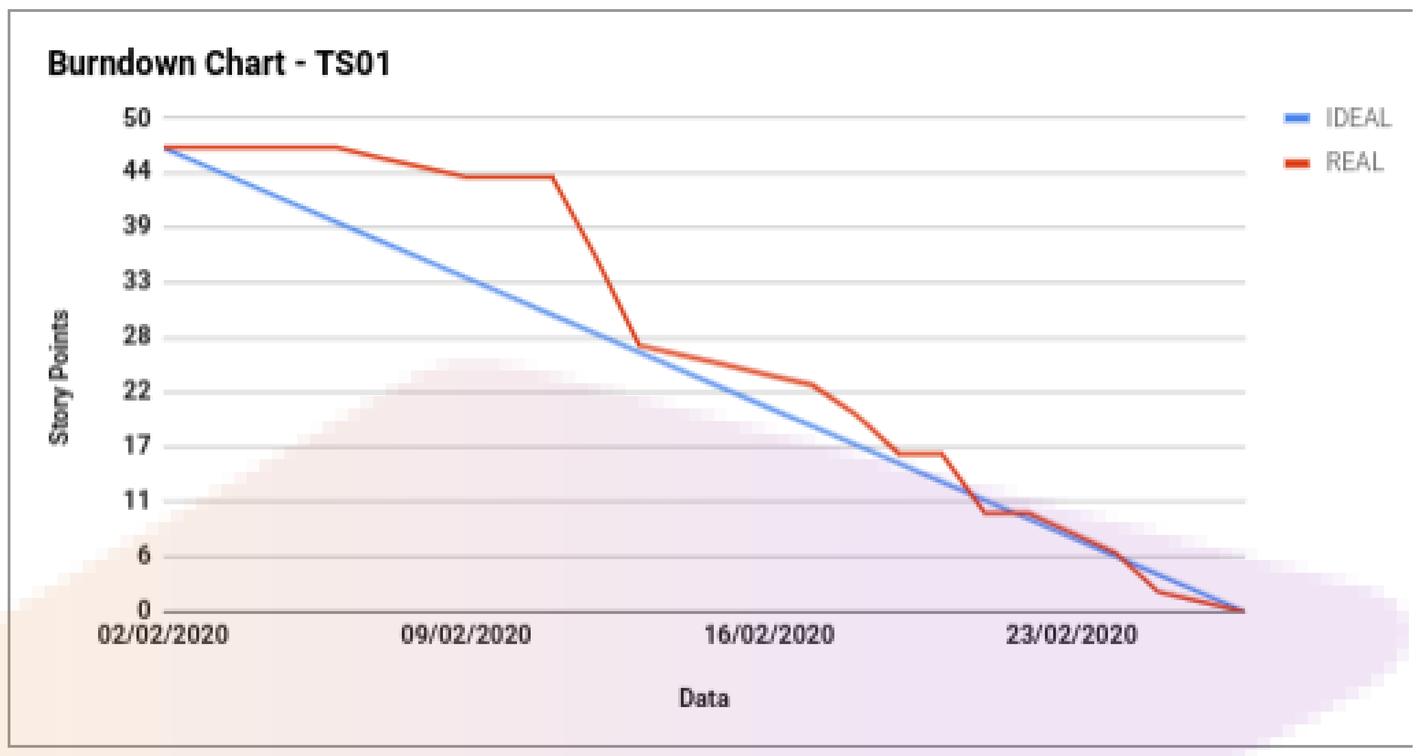
Trata-se de um gráfico que torna visível a evolução diária do trabalho do time de desenvolvimento, pois mostra a comparação entre o trabalho estimado inicialmente no Sprint Planning Meeting com a quantidade restante estimada de trabalho.

Normalmente, as unidades utilizadas são de esforço (em horas) planejado pelo tempo decorrido.

A figura a seguir mostra um exemplo de um Burndown Chart:

BURNDOWN CHART - #TS01

DATA	IDEAL	REAL
02/02/2020	47	47
03/02/2020	45	47
04/02/2020	43	47
05/02/2020	41	47
06/02/2020	39	47
07/02/2020	38	46
08/02/2020	36	45
09/02/2020	34	44
10/02/2020	32	44
11/02/2020	30	44
12/02/2020	28	36
13/02/2020	26	27
14/02/2020	24	26
15/02/2020	23	25
16/02/2020	21	24
17/02/2020	19	23
18/02/2020	17	20
19/02/2020	15	16
20/02/2020	13	16
21/02/2020	11	10
22/02/2020	9	10
23/02/2020	8	8
24/02/2020	6	6
25/02/2020	4	2
26/02/2020	2	1
27/02/2020	0	0



Time Auto-organizado

A autoorganização de um time é focada em quatro estágios:

a) forming: início da formação do time, onde ninguém ainda se conhece.

b) storming: os membros do time passam a se conhecer e ainda não estão habituados ao comportamento uns dos outros.

c) norming: os membros começam a aparar as arestas, criando algumas normas ou regras implícitas.

d) performing: o time passa a funcionar de forma sincronizada, melhorando bastante o desempenho do time como um todo.

Somos Todos T.I.

instagram.com/somostodosti

facebook.com/somostodosti

somostodosti.com.br

somostodosti.com.br/canal_youtube

github.com/andreolimpio/somostodosti