



INSTITUTO DE GESTÃO E  
TECNOLOGIA DA INFORMAÇÃO



## Ciclo de Vida Ágil

Bootcamp Agile Expert



Isabella Fonseca

2021

## **Ciclo de Vida Ágil**

### **Bootcamp Agile Expert**

Isabella Fonseca

© Copyright do Instituto de Gestão e Tecnologia da Informação.

Todos os direitos reservados.

## Sumário

---

### Capítulo 1. 5

Mundo VUCA	4
Framework Cynefin	5
Organizações Antifrágeis	7

### Capítulo 2. 11

Manifesto Ágil e seus quatro valores:	9
Doze princípios	10
Origem do Scrum	11
O que é o Scrum?	15
Pilares	16
Esqueleto do processo	17
Product Backlog	18
Sprint Backlog	18
Incremento	21
Ciclo de Vida	21
Sprint Planning Meeting	22
Planning Poker	23
Daily Scrum Meeting	24
Sprint Review Meeting	24
Sprint Retrospective Meeting	25
Exemplos de indicadores de acompanhamento de Sprint	26
Checklist	29

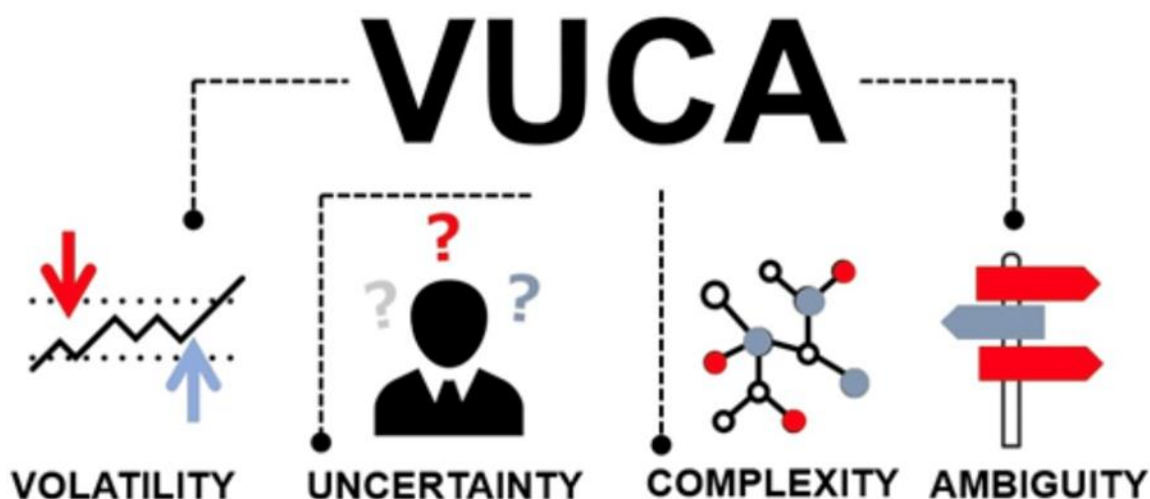
Resumo do Planejamento de um Projeto Scrum	29
Referências.....	32

## Capítulo 1. Conceitos Iniciais

### Mundo VUCA

Vivemos no mundo VUCA! VUCA é uma sigla que representa as diversas situações do nosso dia a dia através de substantivos como volatilidade (Volatility), incerteza (Uncertainty), complexidade (Complexity) e ambiguidade (Ambiguity). Essa é a forma como as empresas se veem agora e também em seu futuro. É claro, portanto, que impacta fortemente na maneira que agimos, pois o significado de um evento/situação atual pode não ser facilmente compreendido. Por isso, faz todo o sentido olhar esse novo mundo sob a luz de novos conceitos.

**Figura 1 – Mundo VUCA.**



<https://comunidadesebrae.com.br/blog/o-desafio-do-planejamento-estrategico-em-um-mundo-vuca>

Mas e o mundo VUCA? Ele veio pedir para sairmos de nossa zona de conforto e nos prepararmos para adaptar rapidamente a este contexto, aprendendo a partir dos erros e procurando insistentemente novas formas de atuação, dando valor à colaboração, às pessoas e às suas interações. Como disse Rodrigo Bastos, em seu artigo no LinkedIn, “O mundo VUCA chegou e todos estão tocando violino no deck do Titanic.”

“Nós estamos saindo de um mundo linear de saber a solução dos problemas e tomar uma decisão clara para um mundo dinâmico de entender o sentido, de tomada de decisão baseada no risco, em condições VUCA.” – Greg Hutchins.

Conceitos ligados à agilidade — métodos e abordagens ágeis — surgiram para preencher muitos “gaps” nesse sentido, por se basearem na experimentação, na resolução rápida de problemas, na inovação, nas equipes multifuncionais, na comunicação direta e transparente, na colaboração e no trabalho real de equipe.

### Framework Cynefin

Criado por Dave Snowden, o modelo Cynefin é um framework para tomada de decisão utilizada nos mais diversos segmentos. Ele foi criado em 1999. É especialmente eficaz para melhorar o gerenciamento nos tempos de mundo VUCA e de tamanha incerteza. Ele tem o objetivo de integrar agilidade, resiliência e inovação.

À direita do framework, estão os domínios "Complicado" (Complicated) e "Simples" (Simple), que representam sistemas Ordenados, ou seja, que apresentam relações claras de causa e consequência. Já ao lado esquerdo, estão os domínios categorizados como “desordenados”, compostos pelos grupos “Complexo” (Complex) e “Caótico” (Chaotic). Nestes domínios não há a possibilidade de se estabelecer claramente as relações de causa e consequência dos sistemas neles contidos.

Na área central, tem-se “sem ordem” (Disorder), onde estão representados os sistemas não categorizados, em que não se sabe ou não se tem certeza sobre o tipo de problema enfrentado.

**Figura 2 – Framework Cynefin.**



“Executivos sábios adaptam suas abordagens para atender a complexidade das circunstâncias que enfrentam.” – David J. Snowden & Mary E. Boone.

As situações de nosso cotidiano não necessariamente estão “paradas” ou podem ser categorizadas de maneira estática em um domínio em especial. A vantagem de se utilizar o framework é entender que as situações podem mudar de domínio, alterando de estado para outro, como do Complicado para o Complexo, quando se busca inovação, ou do Complexo para o Complicado quando se busca otimizar e explorar uma solução. Portanto, ele sugere formas de lidar com o trabalho de acordo com o domínio no qual o sistema se encontra.

Segundo Chris Young, “a habilidade do Cynefin de ressaltar a dinâmica dos sistemas e mostrar que mudanças estão na natureza das coisas torna mais fácil vencer as resistências”.

Para se manterem competitivas, as empresas devem estar na fronteira da inovação. Essa fronteira está representada no Cynefin, no local onde os domínios Complexo e Caótico se encontram, ou seja, “à beira do caos”. Problemas com domínios complexos e caóticos necessitam de criatividade. Uma vez que o seu sistema educacional te priva de errar, você acaba não sendo criativo por causa do medo. Para isso, é necessário estimular momentos de ideação com trabalho em equipe (equipe novamente no foco!).

É preciso ter muitas ideias ruins para surgirem algumas boas. A equipe surge, principalmente, porque precisamos de muita colaboração para que apareçam novas ideias, que são construídas a partir de outras para solucionar os problemas atuais. E, como o fracasso faz parte dessa jornada, é preciso perseverar, de forma que a liderança apareça, para que cada ciclo de experimentação sirva de aprendizado.

### Organizações Antifrágeis

Devido a tudo isso, o caminho possível para atuação, nesse cenário, é ter a capacidade de se adaptar e de agir usando um mindset ágil, bem como interagir constantemente para disseminar informações em busca de um ganho contínuo e acelerado de conhecimento pelas pessoas que realizam um projeto.

Outro ponto, é a existência de empresas que não nasceram ágeis e digitais, e com isso já trazem uma realidade de funcionamento e modelo mental muito diferente e distante do que é considerado ideal no mundo VUCA. São organizações de estruturas hierárquicas rígidas, com poder de decisão nas mãos de poucos, informações e dados pouco compartilhados e uma cultura que considera fracasso muito caro e, por isso, minimiza riscos e não maximiza autonomia e agilidade. São as



chamadas organizações Slow By Design. Esse termo foi cunhado por Eric Schmit, presidente e ex-chefe executivo da Alphabet Inc., anteriormente, denominada Google, a principal subsidiária da empresa.

Com estrutura e processos muito bem definidos, essas organizações não ágeis e digitais criam oportunidades para que seus colaboradores possam ficar em sua zona de conforto – fazendo aquilo que é definido com total segurança de que está fazendo a coisa certa. As pessoas não se arriscam, ou não podem arriscar fazer pequenas mudanças ou melhorias sem uma aprovação prévia do superior hierárquico. Por vezes, as pessoas não compreendem onde o seu trabalho agrega para o produto da organização, e essa limitação de visão bloqueia a criatividade. Quando se dá um movimento de transformação nesse contexto, as pessoas não se sentem confortáveis em assumir autonomia, empoderamento e auto organização.

Na contramão, há outro tipo de organização chamado “Antifrágil”. Esse termo foi extraído do livro “Antifrágil: coisas que se beneficiam com o caos”, de Nassim Taleb. São organizações que, para sobreviverem melhor neste contexto, precisam se tornar mais do que resilientes ou o oposto do frágil. E mais, são aquelas que só melhoram com os choques, o caos, a desordem e o estresse. Empresas têm que ser mais que resilientes – o resiliente mantém sua forma, enquanto o antifrágil a melhora!

**Figura 3 – Antifrágil.**



**Fonte:** <https://blog.me.com.br/tag/antifragil/>.

Como diz Nassim Taleb, “antifrágil é tudo aquilo que aproveita o caos (pressões externas que poderiam deteriorar o que é frágil, por exemplo) para evoluir.

A cada queda se aprimora, melhora com o caos! Para chegar ao nível de antifrágil, as organizações precisam pensar em aprender mais rápido, promover uma cultura de aprendizagem sólida e criar momentos eficientes de experimentação e de entendimento de erros, de feedback, de iteração e de muita colaboração.

## Capítulo 2. Manifesto Ágil

---

Os métodos ágeis foram se fortalecendo, dirigindo para este sentido de entendimento de uma nova forma de se construir software, tornando-se uma vertente que está complementando e expandindo o conhecimento do que é software e desenvolvimento de novos produtos em geral.

Em 2001, um grupo interessado em métodos iterativos e ágeis — esses termos estavam, na verdade, sendo ainda atribuídos no contexto do desenvolvimento de software — reuniu-se para encontrar um denominador comum. Muitas das práticas concretas existentes, atualmente, no universo ágil, derivaram desses princípios.

### Manifesto Ágil e seus quatro valores:

---

Apesar de reconhecer que há valor nos itens da direita, valorizamos mais os itens da esquerda:

**Indivíduos e interação entre eles** mais que processos e ferramentas.

**Software em funcionamento** mais que documentação abrangente.

**Colaboração com o cliente** mais que negociação de contratos.

**Responder a mudanças** mais que seguir um plano..

Versão traduzida obtida de: <<http://www.manifestoaquil.com.br/>>

## Doze princípios

---

- Nossa maior prioridade é satisfazer o cliente, por meio da entrega adiantada e contínua de software de valor.
- Aceitar mudanças de requisitos, mesmo no fim do desenvolvimento. Processos ágeis se adequam às mudanças para que o cliente possa tirar vantagens competitivas.
- Entregar software funcionando com frequência, na escala de semanas até meses, com preferência aos períodos mais curtos.
- Pessoas relacionadas aos negócios e aos desenvolvedores devem trabalhar em conjunto e diariamente, durante todo o curso do projeto.
- Construir projetos ao redor de indivíduos motivados, dando a eles o ambiente e suporte necessário, confiando que farão seu trabalho.
- O método mais eficiente e eficaz de transmitir informações para e por dentro de um time de desenvolvimento é através de uma conversa cara a cara.
- Software funcionando é a principal medida de progresso.
- Processos ágeis promovem um ambiente sustentável. Os patrocinadores, desenvolvedores e usuários devem ser capazes de manter conversação pacífica continuamente.
- A contínua atenção à excelência técnica e ao bom design aumenta a agilidade.
- Simplicidade: a arte de maximizar a quantidade de trabalho não feito é essencial.
- As melhores arquiteturas, os requisitos e os projetos emergem de times auto-organizáveis.

- Em intervalos regulares, o time deve refletir em como se tornar mais efetivo e, então, ajustar-se e otimizar seu comportamento de acordo.

Versão traduzida obtida de: <http://www.manifestoagil.com.br/>.

De forma geral, os métodos de desenvolvimento ágil aplicam iterações de tempo fixo para evoluir um produto de software em incrementos, um planejamento adaptativo, entregas evolutivas e outras práticas e valores que encorajam a agilidade — rapidez e flexibilidade em resposta à mudança.

Por que planejar não é uma tarefa fácil?

Porque planejamos para um mundo próximo ao mundo real e esperamos seguir um plano que quase sempre não é cumprido. Podemos definir um “projeto como um empreendimento temporário com o objetivo de criar um produto ou serviço único”. Temporário significa que cada projeto tem um começo e um fim bem definidos, e único significa que o produto ou serviço produzido é de alguma forma diferente de todos os outros produtos ou serviços semelhantes.

## Origem do Scrum

---

Os termos "iterativo e incremental", cunhados no contexto das abordagens ágeis para desenvolvimento de software, possuem , atualmente, uma ampla utilização, porém nem sempre carregando o significado que realmente merecem.

Quando dizemos uma abordagem "iterativa", queremos dizer que o **produto** construído mediante esta abordagem será feito de forma gradual, começando a partir de uma ideia inicial, não acabada, recebendo novas avaliações e aprimoramentos até que esteja em condições aceitáveis em termos de valor para o cliente.

A figura abaixo demonstra uma evolução iterativa, a partir de uma visão inicial.



Quanto ao termo "incremental", significa a produção do produto final por meio de incrementos, criando oportunidades de análise e adaptação do **processo** de produção a cada novo incremento. A figura a seguir demonstra uma abordagem incremental para construção, a partir de uma referência inicial.



Iterativo e incremental referem-se, portanto, a uma abordagem em que o produto é construído gradualmente, em ciclos que buscam o alinhamento de prioridades e necessidades conforme a evolução do produto alcançada até o momento, permitindo ao mesmo tempo uma revisão constante do processo mediante qual está sendo construído.

Dito isso, para tentar resolver problemas de planejamento, prover feedback constante e entregas curtas, o movimento acerca do Desenvolvimento Iterativo e Incremental (IID) cresceu.

IID é um processo de desenvolvimento de software que prevê planejamento em estágios desenvolvendo e entregando o produto em incrementos de software funcionais, promovendo em tempos regulares previsões para reflexão. Têm-se, com isso, Múltiplos PDCA's: **Plan - Do - Check - Act**. O ciclo de Deming tem como princípio tornar mais claros e ágeis os processos envolvidos na execução, focando

em Melhoria Contínua. A relação desse processo com Scrum é clara, pois as iterações são tidas como primordiais ao melhor entendimento do que é software, como citado anteriormente. Cada iteração precisa passar por todas as fases necessárias ao desenvolvimento de software: análise, projeto, desenvolvimento, testes, entregas - continuamente. No Scrum:

- Cada sprint é uma iteração que segue o ciclo PDCA e entrega incremento de software pronto.
- Cada dia de um ciclo Scrum tem um PDCA menor – Daily Scrum (a ser explicado mais tarde!).
- Scrum permite a criação de equipes auto-organizadas, encorajando a comunicação verbal entre todos os membros da equipe e entre todas as disciplinas que estão envolvidas no projeto, focando na maximização da habilidade da equipe de responder de forma ágil aos desafios emergentes.

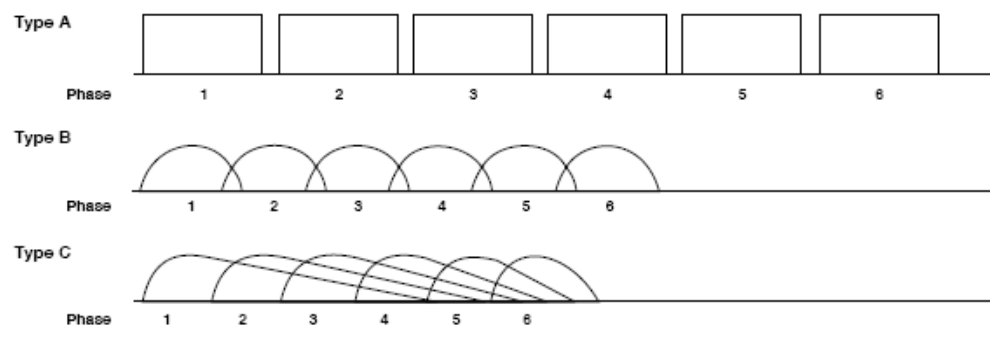
Outro fundamento do Scrum é o Lean Manufacturing. O lean é uma abordagem derivada do Sistema Toyota de Produção, que tem como principal objetivo eliminar o desperdício, tirando do processo tudo que não traga valor ao produto final.

Outra grande contribuição é o artigo de Nonaka e Takeuchi - “The New New Product Development Game”. Foi publicado em 1986 na Harvard Business Review. Esse artigo estuda as práticas de gestão que diferenciavam as empresas Fuji Film, Toyota, 3M e Xerox, Epson, Brother, NEC e Honda, e identificam diferenças chave entre empresas japonesas e americanas inovadoras, que se apresentavam tão à frente das demais.

**Figura 4 – The overlapping of phases does away with traditional notions about division of labor.**

#### EXHIBIT 1

Sequential (A) vs. overlapping (B and C) phases of development



**Fonte: Takeuchi and Nonaka (1986).**

Sob movimento de sequência e revezamento, o projeto segue passo a passo as diferentes fases, mudando para a seguinte somente após todas as expectativas da fase anterior estarem totalmente satisfeitas. Checar todos os itens controla os riscos, mas ao mesmo tempo deixa pouco espaço para integração. Um tumulto durante o processo pode vir a desacelerar ou mesmo parar o desenvolvimento de todo o processo.

O grande ganho nas abordagens B e C é a paralelização das atividades. Acaba antes, pois constrói-se enquanto se planeja. Na velha forma (tipo A), o processo de desenvolvimento do produto é como em uma corrida de revezamento, com um grupo de especialistas em cada função, passando o bastão para o próximo grupo.

O projeto passou sequencialmente de uma fase a outra: conceito em desenvolvimento, teste de viabilidade, desenho do produto, processo de desenvolvimento, produto piloto e produto final. Dessa maneira, as funções foram especializadas e segmentadas: pesquisadores de mercado avaliaram as necessidades e percepções do cliente no conceito de desenvolvimento do produto.



Na abordagem em forma de rugby (jogo britânico tipo o futebol americano), o processo de desenvolvimento do produto surge a partir de constante interação, como que de mãos dadas, em que um grupo extremamente disciplinado trabalha junto do início ao fim. Em vez de um movimento definido, altamente estruturado em suas fases, o processo se inicia com a interação do grupo. Um grupo de engenheiros, por exemplo, pode começar o desenho do produto (3ª fase) antes de todos os resultados do teste de viabilidade (2ª fase) estarem completos. Ou o grupo de engenheiros pode ser forçado a reconsiderar a decisão como resultado de informação tardia. O grupo não para, mas se empenha em experimentar a interatividade. Isso continua até a fase mais adiantada do desenvolvimento do processo.

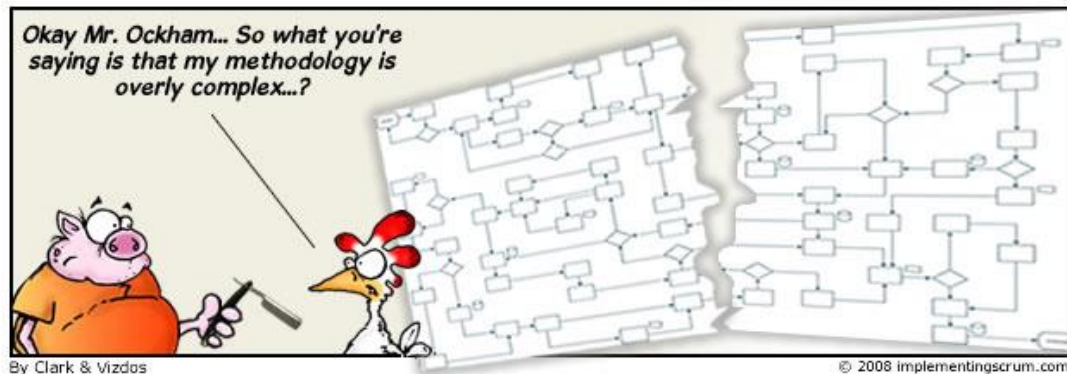
A mudança da aproximação linear (tipo A) para a integrada (tipos B e C) encoraja tentativas e erros. Ela estimula novas maneiras de aprender e de pensar dentro da organização em diferentes níveis e funções. Muito importante, essa estratégia de desenvolvimento do produto pode agir como agente de troca em uma grande organização.

Estudando as práticas de gestão que diferenciavam as empresas japonesas, os seguintes fatores de sucesso comuns foram encontrados:

- Instabilidade Interna;
- Times auto organizados;
- Fases de Desenvolvimento Sobrepostas;
- Multi-aprendizado (Multi-Nível e Multi-Funcional);
- Controle Sutil;
- Transferência de Conhecimento Organizacional.

## O que é o Scrum?

---



O Scrum auxilia a controlar projetos de desenvolvimento de software, porém isso não significa que o projeto irá exatamente como esperado, produzindo resultados idênticos aos previstos. O Scrum controla o processo de desenvolvimento de software para guiar o trabalho em direção ao resultado de maior valor possível.

Seguem abaixo alguns tópicos para nos auxiliar a entender o que o Scrum é:

- Abordagem iterativa e incremental para o gerenciamento de projetos de software;
- Processo empírico para gerenciamento de projetos complexos, nos quais é impossível prever tudo que irá ocorrer;
- Processo criado para ampliar a probabilidade de desenvolver software com sucesso;
- Processo de gerenciamento de projetos de software que permite antecipar e avaliar com maior precisão o retorno do investimento;
- Oferece um arcabouço (framework) e um conjunto de práticas que mantêm tudo visível;

- Permite aos seus praticantes saber exatamente o que está acontecendo e realizar ajustes locais para manter o projeto movendo na direção de seus objetivos;
- Permite exercitar constantemente o bom-senso.

## Pilares

---

**Visibilidade:** os aspectos do processo que afetam o resultado devem estar visíveis àqueles que estão controlando o processo. Além de visíveis, devem ser também precisos (verdadeiros), uma vez que não há espaço para o engano das aparências no controle empírico de processos.

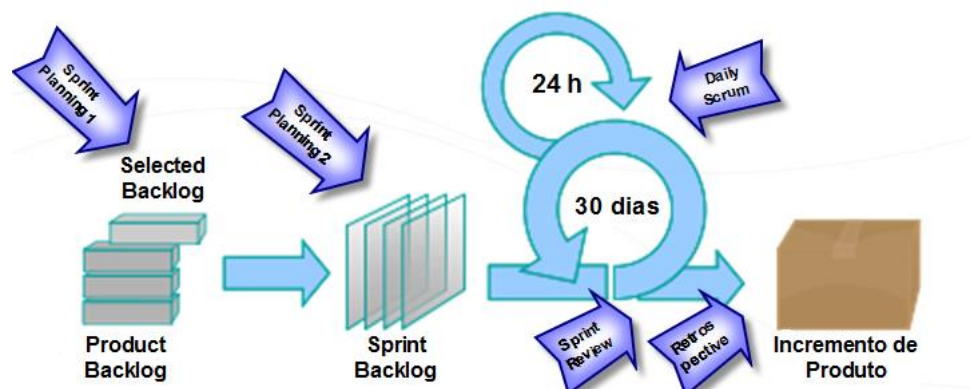
**Inspeção:** os vários aspectos do processo precisam ser frequentemente inspecionados, de forma que variações inaceitáveis no processo sejam detectadas.

**Adaptação:** se a inspeção detectar que um ou mais aspectos do processo estão fora dos limites aceitáveis e que o produto resultante será inaceitável, o processo ou o material sendo processado precisa ser ajustado. O ajuste deve ser feito o mais rápido possível para evitar novos desvios.

O Scrum endereça a complexidade de projetos de desenvolvimento de software por meio da implementação desses três pilares em um conjunto de práticas e regras simples.

## Esqueleto do processo

O Scrum sustenta todas as suas práticas em um esqueleto de processo iterativo e incremental.



Fonte: [http://en.wikipedia.org/wiki/Scrum\\_\(development\)](http://en.wikipedia.org/wiki/Scrum_(development)).

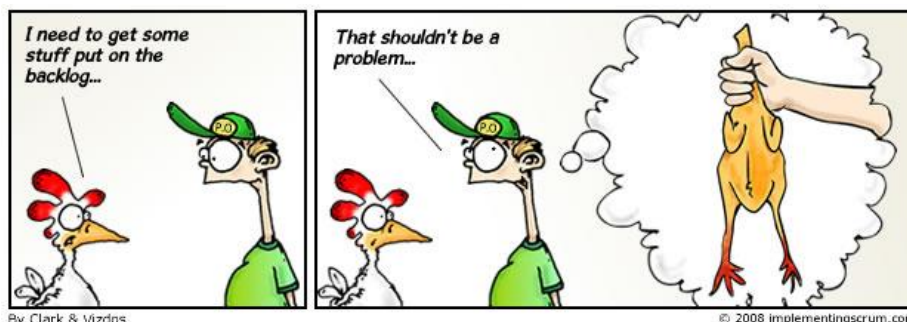
O ciclo inferior (30 dias) representa uma iteração de atividades de desenvolvimento que ocorrem em sequência, tendo como saída um incremento de produto. O ciclo superior (24 horas) representa a inspeção diária que ocorre durante a iteração, na qual os membros da equipe se reúnem para inspecionar as atividades e encaminhar as adaptações necessárias. O que direciona a iteração é uma lista de requisitos. O ciclo é repetido até que o projeto não tenha mais fundos.

O processo se desenvolve da seguinte forma: no início da iteração, a equipe revisa o que precisa ser feito. Em seguida, seleciona aquilo que considera um incremento entregável de produto para constituir-se no objetivo da iteração. A equipe, então, foca no trabalho para produção do objetivo acordado; durante a iteração não há alteração de prazo ou de objetivo. Ao final da iteração, a equipe apresenta o incremento de software produzido para que possa ser avaliado pelas partes interessadas do projeto e, se necessário, possam ser realizadas adaptações necessárias no projeto.

A produtividade do Scrum reside no processo criativo que se realiza dentro de cada iteração. A equipe é responsável por organizar-se em torno do objetivo da

iteração, analisando questões tecnológicas, arquiteturais e de requisitos, conforme as competências e habilidades de seus membros.

## Product Backlog



O Product Backlog é uma lista priorizada de requisitos funcionais e não funcionais do projeto, com tempos estimados para transformá-los em funcionalidade do sistema. As estimativas são em dias e devem ser mais precisas para os itens mais altos em termos de prioridade. A lista evolui, modificando-se na medida em que as condições de negócio ou a tecnologia mudam.

"O Product Backlog é o coração do Scrum. É como tudo se inicia. O Product Backlog é basicamente uma lista priorizada de requisitos, ou histórias, ou funcionalidades, ou qualquer outra coisa. Coisas que o cliente quer, descritas usando a terminologia do cliente."

O Product Backlog pode ser composto de requisitos funcionais, requisitos não funcionais e problemas, priorizados em ordem de importância para o negócio e as dependências e, então, estimados. A precisão da estimativa depende da prioridade e granularidade do item, sendo que os itens de mais alta prioridade, que podem ser selecionados para a próxima Sprint, têm estimativa bem granular e precisa.

O Product Backlog deve conter, além dos requisitos do produto, também os requisitos do projeto, tais como a realização de auditorias da qualidade, preparação de ambiente de produção etc. Segue abaixo um exemplo de backlog de produto.

## Sprint Backlog

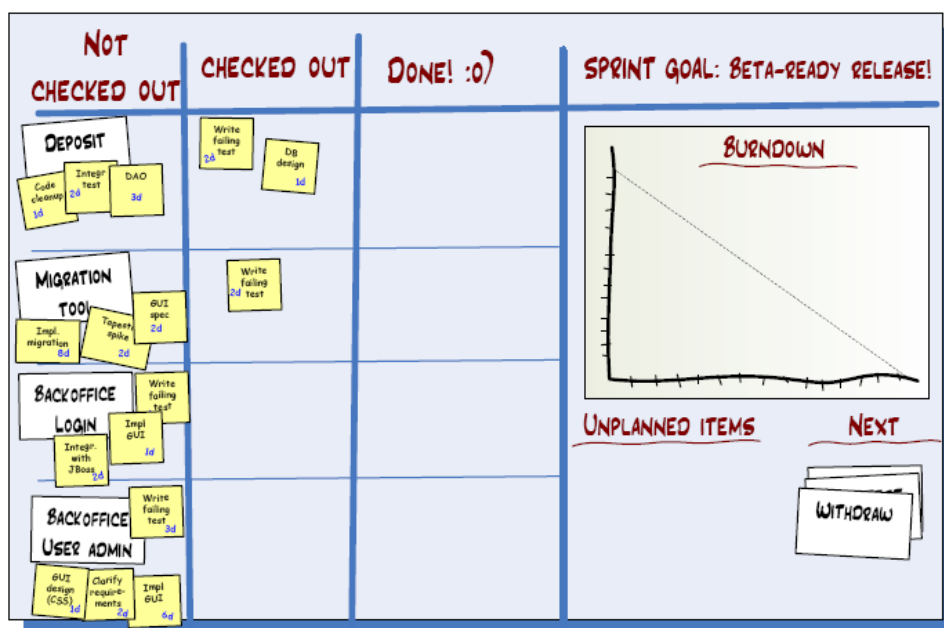
---

É uma pilha de tarefas (atividades) definida pelo time e para o time, a partir dos itens de Selected Backlog, contendo detalhamento em 2-3 itens de Sprint Backlog, do que deve ser feito em um Sprint. Esta atividade é primariamente feita durante o Sprint Planning Meeting 2, sendo revisada diariamente durante o Daily Scrum.

- É utilizada para o Sprint Burndown Chart e não deve ser utilizada para gerenciamento externo (realizado por resultados);
- Unidade em HH Previsto (Tamanho x Velocidade Atual)

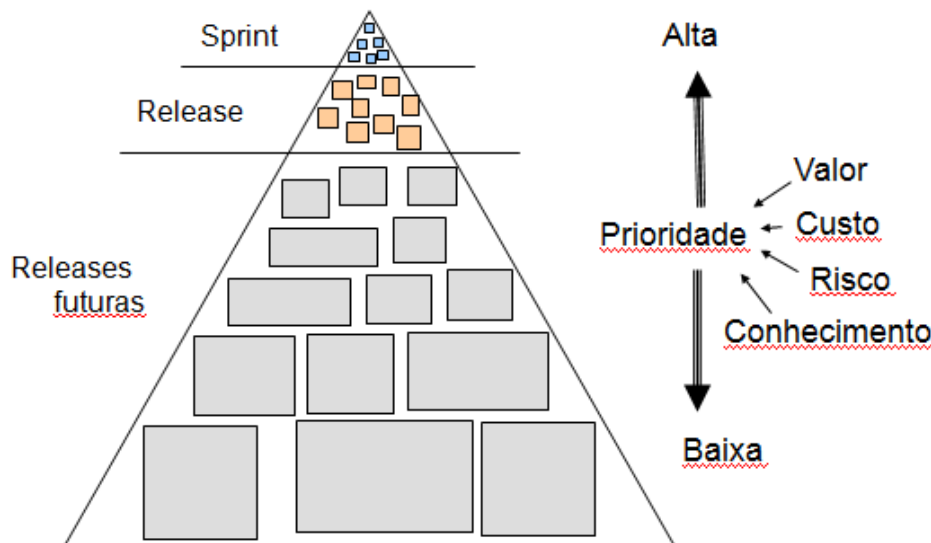
O Sprint Backlog é uma lista de tarefas que define o trabalho da Equipe para uma Sprint. A lista emerge durante o Sprint, sendo que uma primeira compilação é gerada na segunda parte da reunião de planejamento da Sprint. É responsabilidade da Equipe gerar e manter atualizado o Sprint Backlog. Cada tarefa identifica o responsável por executar o trabalho e o esforço remanescente para concluir a tarefa em um dia específico durante a Sprint. Tipicamente, uma tarefa possui estimativa entre 4 e 16 horas, e tarefas com estimativa maior significam que ainda não foram suficientemente detalhadas.

Existem várias maneiras práticas de se trabalhar com um Sprint Backlog, desde planilhas Excel até Sistemas de Gestão baseados em Scrum. Um mecanismo bastante simples e difundido, no entanto, é a utilização de um quadro de tarefas, conhecido também como Agile Radiator, conforme exemplo abaixo.



Durante o projeto, itens a serem trabalhados nos Sprints devem estar melhor definidos e possuem maior prioridade em relação aos demais. O desenho abaixo confirma a granularidade maior que itens do Product Backlog devem ter quando estão planejados para serem executados durante o Sprint. Quando estão planejados em uma próxima release, podem ter uma granularidade menor – e em releases futuras, menores ainda.

A priorização dos itens de um Product Backlog deve seguir premissas estratégicas definidas pela organização, levando em consideração aspectos relativos a custo, dificuldade de implementação (conhecimento da tecnologia), risco, ROI (valor na entrega ao cliente) etc.



No entanto, deve haver clareza e alinhamento entre o Product Owner, o Time e o Scrum Master acerca do que realmente está contemplado por esta estimativa inicial: todos os testes unitários, de integração, desempenho etc., estão contemplados? Está previsto esforço para desenvolvimento de testes automatizados e refactoring do código? Esse alinhamento chama-se **conceito de pronto** e será muito importante ao longo de todo o ciclo do Scrum.

O planejamento do projeto deve evoluir conforme são vencidas as barreiras iniciais do projeto. Após a primeira Sprint, o planejamento pode ser ajustado para considerar com maior precisão a velocidade da equipe nas tecnologias envolvidas, questões ambientais da empresa etc. Isso permitirá ajustes como a alteração da equipe, a revisão do escopo ou outras iniciativas que possam favorecer que o projeto atinja as expectativas das partes interessadas.

## Incremento

A cada Sprint, a equipe deve construir um incremento de produto de software que seja potencialmente entregável, ou seja, que possa a qualquer momento ser colocado em produção por determinação do Product Owner. Para isso, é necessário

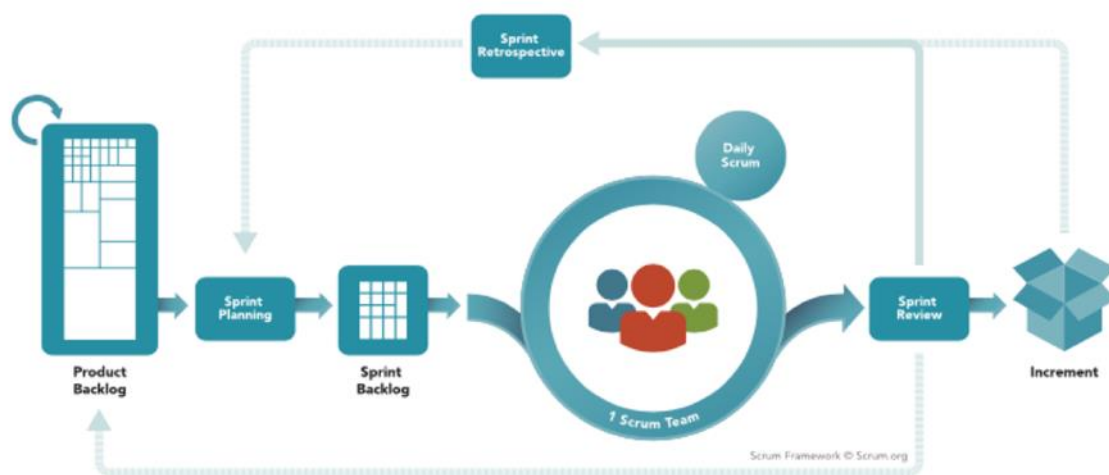


que o produto de cada Sprint seja um software executável, cujo código seja bem estruturado e bem escrito, tenha sido consistentemente testado e possua a documentação necessária.

### Ciclo de Vida

Abaixo, figura com as reuniões que compõem um Sprint:

## SCRUM FRAMEWORK



## Sprint Planning Meeting

---

Reunião dedicada ao planejamento da Sprint, limitada a 8 horas de duração, sendo essas divididas em duas partes de até 4 horas cada: a primeira parte é dedicada à seleção dos itens do backlog do produto que o time irá se comprometer a entregar na Sprint, enquanto a segunda parte é dedicada à criação do Sprint Backlog.

Características principais:

- Realizada todo início de **Sprint**, com duração de 4 (quatro) horas;
- Participação do Product Owner (PO), Scrum Master (SM) e Scrum Team;
- Passagem de entendimento sobre cada item do **Product Backlog** pré-selecionado do PO para o Scrum Team;
- O Scrum Team realiza a estimativa inicial de tamanho, para cada item do Selected Backlog;
- O PO estabelece o **Sprint Goal**.

**IMPORTANTE:** o Sprint Goal é uma meta “visionária” do Sprint estabelecido pelo Product Owner ao final da reunião de Sprint Planning 1. É sempre verificado durante a reunião de Sprint Review, confrontado com a demonstração do Scrum Team e os itens de backlog planejados e comprometidos.

A reunião de planejamento poderá ter uma duração menor, desde que isso seja acordado entre Product Owner, Scrum Team e Scrum Master e se mantenha o limite acordado em todas as reuniões. A aplicação do conceito de etapas delimitadas por tempo ("time-boxed") é muito importante no Scrum.

O Product Backlog deve estar preparado antes da reunião. Essa preparação é muito importante para o correto desenvolvimento da reunião.

O Time pode fazer sugestões, mas a decisão final sobre os itens do Backlog a integrarem a Sprint é do Product Owner. As estimativas iniciais dos requisitos no backlog do produto são revistas ao serem abordados os detalhes de cada um com o Product Owner. A técnica de estimativa mais popular que pode ser utilizada durante esta reunião é conhecida como Planning Poker, explicada a seguir.

### Planning Poker



O “Planning Poker” é uma técnica de estimativa baseada em consenso, criada a partir de um método denominado “Whiteband Delphi”, e que é utilizada para estimar o tamanho de requisitos em projetos de software. Esse método vem sendo bastante utilizado, principalmente, entre a comunidade ágil, como uma ferramenta importante para a produção de estimativas acordadas entre os membros da equipe, em um tempo significativamente reduzido se comparado a discussões livres acerca do esforço de cada requisito.



O grande benefício do Planning Poker é que todos os participantes têm que expressar simultaneamente sua estimativa de tamanho para determinado requisito, evitando que a opinião da maioria se ancore a partir da opinião dos primeiros a se manifestar.

### *Daily Scrum Meeting*

---

A reunião diária do Scrum (Daily Scrum Meeting) é um rápido momento de reunião de todos os membros do Time, dedicado à resposta das três perguntas mencionadas anteriormente, visando exercitar os conceitos da inspeção e da adaptação, diariamente, entre a equipe. Essa reunião tipicamente dura 15 minutos e acontece no mesmo local e horário todos os dias.

Outro aspecto é a transparência que a Daily Scrum confere aos processos que estão, atualmente, em execução na organização, de forma que, caso haja algum processo organizacional ou outra interferência qualquer presente diariamente na reunião como um elemento impeditivo do avanço, caberá uma ação para eliminar esse impedimento. O Scrum Master terá, dessa forma, os fundamentos necessários para acionar as engrenagens da empresa na busca do melhor contexto para a produtividade do Time.

Além disso, o Scrum Team deve promover a atualização do Agile Radiator com as informações levantadas durante essa reunião.

### *Sprint Review Meeting*

---

- Realizada após cada Sprint, com duração entre 2 a 4 horas e participação do Product Owner;
- O Scrum Team demonstra o trabalho realizado no Sprint e é avaliado pelo Product Owner;

- A demonstração deve ser do produto funcionando e comprovar que os itens de Selected Backlogs prioritários (e associado ao Sprint Goal) foram realizados com objetivo de maximizar o ROI.

O Time e o Product Owner devem estar em constante colaboração, definindo em conjunto como atingir o maior valor ao negócio a partir das tecnologias disponíveis. A reunião de apresentação da Sprint (Sprint Review Meeting) é um momento muito oportuno e conveniente para que haja forte interação face a face entre o Time e os stakeholders, em um espírito de colaboração, que traz um valor muito maior do que qualquer relatório escrito, porque as ideias são trabalhadas em cima de algo real, que é o software produzido durante a Sprint.

Um aspecto importante da Sprint Review Meeting é que, ao criar uma oportunidade de colocar frente a frente, periodicamente, clientes e equipes de desenvolvimento, cria-se um ambiente de colaboração e uma maior percepção nos clientes de “resultados entregues” pela equipe de desenvolvimento. Em pouco tempo, o que foi priorizado, na última reunião, encontra-se entregue na forma de software real e não de documentação intermediária que pouco ou nada interessa aos clientes. Esse ambiente de colaboração, se bem conduzido pelo Product Owner, tende a formar um ciclo virtuoso entre equipe e clientes, ou ao menos diminuir os conflitos e reclamações de insatisfação por parte de clientes.

Apesar de haver, em casos especiais, Sprints que produzem artefatos intermediários, como especificações ou ainda atividades de implantação, tipicamente a Sprint produz software. O objetivo da Sprint review é **apresentar esse software em funcionamento** para todos os interessados que queiram e que seja conveniente que participem.

Idealmente, o Product Owner já conhece o produto que está sendo desenvolvido e não se espera que existam surpresas na reunião de apresentação.

### *Sprint Retrospective Meeting*

---

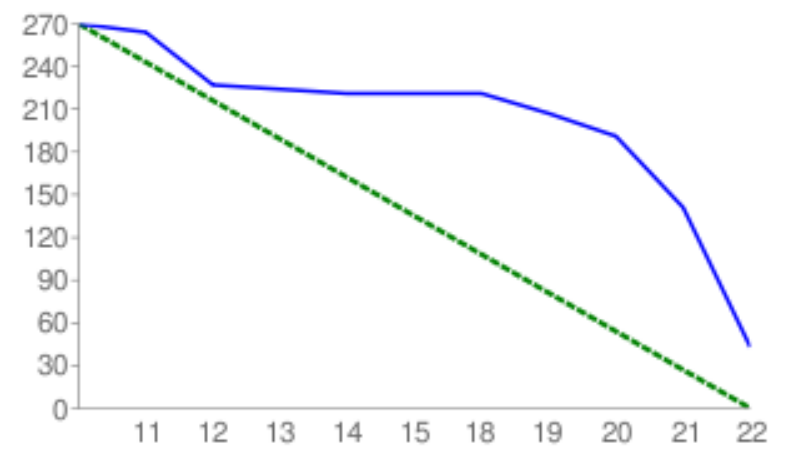
O objetivo da reunião retrospectiva da Sprint é promover reflexões, alavancar adaptações e melhorias na equipe nas práticas do Scrum. O funcionamento do processo na última Sprint é analisado e ajustado para funcionar melhor na próxima Sprint. Essas reuniões são limitadas ao tempo de duas horas cada.

- Realizada após o Sprint Review, com duração de 1-2 horas -> lições aprendidas.
- Timeline pode ser relembrado em grupo.
- Duas perguntas: “O que foi bem?” (WWW-What Went Well?) e “O que pode ser melhorado?” (WCBI - What Can Be Improved?).
- Separação de responsabilidades pelo WCBI (organizacional e time) e atualização no Agile Radiator.

### *Exemplos de indicadores de acompanhamento de Sprint*

---

**Gráfico Burndown:** indicador que mede o progresso do trabalho refletindo diariamente seu trabalho. Sua curva indica se o Scrum Team está se adaptando para o plano definido e comprometido por todos – Sprint Goal – e conseguindo trabalhar de forma realmente iterativa ou sofrendo impedimentos em excesso.



**Histórico de sucesso de alcance de Sprint Goal:** indicador simples de resultado de alcance de goal por iteração, a fim de manter a equipe focada em melhorar seu desempenho e comparar resultados anteriores. Além disso, de forma lúdica, pode-se criar mecanismos de premiação em caso de alcances sucessivos.

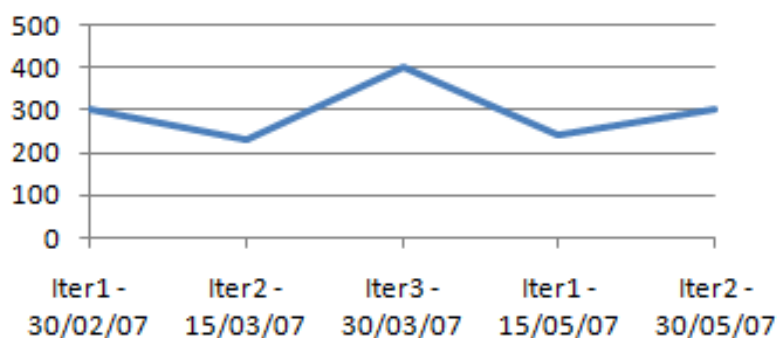
eCompany Suite - Release 5.6 e 5.5.1

							
Sprint 1 BV: 130 Velocidade: 51	Sprint 2 BV: 40 Velocidade: 39	Sprint 3 BV: 210 Velocidade: 34	Sprint 4 BV: 230 Velocidade: 49	Sprint 5 BV: 240 Velocidade: 50	Sprint 6 BV: 175 Velocidade: 43	Sprint 7 BV: 485 Velocidade: 49	Sprint 8 BV: 85 Velocidade: 55

Fonte: <http://www.powerlogic.com.br>.

**Priorização por valor de negócio:** indicador que mede a efetividade das entregas em relação à liberação de máximo valor de negócio ao cliente. Dessa forma, deve-se medir se o Product Owner está priorizando corretamente as demandas de trabalho nas liberações ao longo do tempo.

## Priorização por Valor de Negócio



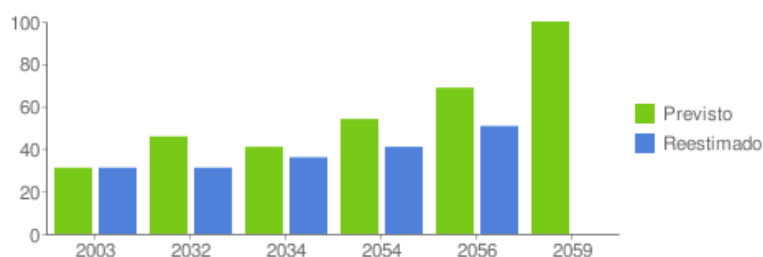
Fonte: <http://www.powerlogic.com.br>.

Ao estabelecer um número de Valor de Negócio (BV ou Business Value) – juntamente com a Alta Gestão – para os grandes Temas a serem desenvolvidos em um produto, e ao desenvolver Requisitos (itens do Product Backlog) rastreados com estes Temas e também priorizados por Valor de Negócio, o PO estabelece uma fórmula e práticas para priorização que objetivam “maximizar o número de BVs liberados”.

Esse indicador mostra, a cada iteração e projeto, ao longo do tempo, o total de BVs liberados. Desse modo, é um importante sinal sobre o trabalho de priorização e de concepção de ideias do Product Owner.

**Tamanho previsto x realizado:** indicador que mede a eficácia do dimensionamento do tamanho dos requisitos ao longo do tempo (pode ser a cada iteração). Dado os itens de Selected backlog do Sprint, compara tamanhos previstos x realizados após a reunião de Release/Sprint Review.

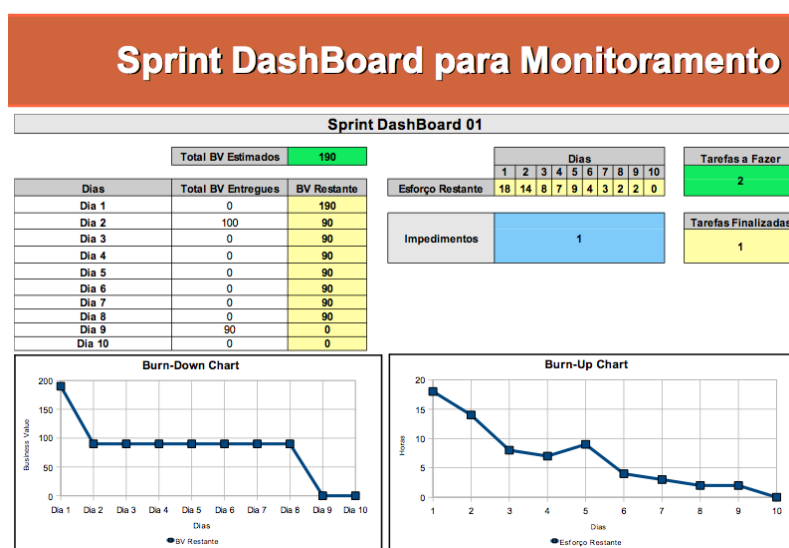




Fonte: <http://www.powerlogic.com.br>.

**Sprint Dashboard:** indicador que acompanha diariamente a evolução de entregas de BV e de esforço restante, consolida número de impedimentos, número de tarefas a fazer e finalizadas, além dos gráficos Burndown e BurnUp.

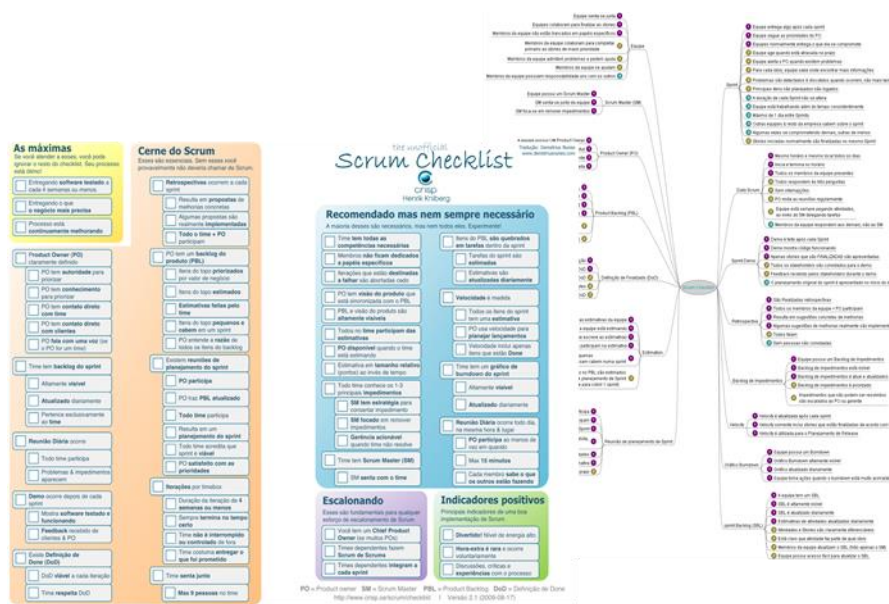
Figura 5 – Gestão Ágil de Projetos com Scrum e FDD.



Fonte: Manoel Pimentel Medeiros. Revista Visão Ágil.

### Checklist

Existem alguns checklists que auxiliam na condução de equipes Scrum e verificam se pontos importantes no framework foram seguidos.



Fonte: Henrik Kniberg – Traduzido para português –

<http://www.slideshare.net/demetriusnunes/scrum-checklist-2766568>.

## Resumo do Planejamento de um Projeto Scrum

O Product Owner é o responsável por obter os fundos necessários para que o projeto entregue um sistema que atenda à visão concebida, de forma a maximizar o ROI. Ele (ou ela) elabora um plano para realizar essa visão, criando o Product Backlog. Essa lista de requisitos é priorizada de maneira que os itens de maior valor para o negócio tenham maior prioridade e, então, é dividida em releases propostas.

O Product Backlog e a proposta de divisão em releases são neste momento um ponto de partida, e certamente irão modificar-se no início do pro-----jeto.

Todo trabalho é realizado em sprints, que são iterações com tipicamente 30 dias de calendário, podendo variar em cada empresa. Uma Sprint é iniciada com uma reunião de planejamento (Sprint Planning Meeting), na qual o Product Owner e a Equipe se unem para definir colaborativamente o que será realizado na próxima Sprint. Partindo dos itens prioritários do Product Backlog, o Product Owner apresenta

as necessidades, e a Equipe identifica o que pode ser realizado. A reunião de planejamento da Sprint é limitada por tempo ("timeboxed"), não devendo ultrapassar a duração de oito horas, para evitar muita especulação sobre o que pode ou não ser produzido; a meta é iniciar rapidamente o trabalho.

A Reunião de Planejamento da Sprint pode possuir duas partes (ou fazer tudo de uma só vez): a primeira é dedicada à apresentação, pelo Product Owner, dos requisitos de mais alta prioridade do Product Backlog para a Equipe, que apresenta suas perguntas acerca desses requisitos, tanto em relação ao entendimento quanto ao seu valor para o negócio e a outros aspectos. Ainda na primeira parte (dentro das primeiras quatro horas), a equipe seleciona o maior número de itens do Product Backlog que ela acredita que será possível desenvolver e entregar na próxima Sprint, comprometendo-se com o Product Owner, que fará o melhor possível.

A segunda metade da reunião é dedicada ao planejamento da Sprint pela Equipe, que gera um plano inicial, preenchendo o Sprint Backlog com as tarefas desse plano. Novas tarefas podem emergir no decorrer da Sprint. A partir da segunda metade da reunião, já está contando o tempo da atual iteração.

Diariamente, a equipe se reúne em uma reunião rápida, de 15 minutos, denominada Daily Scrum. Nesta reunião, cada membro da equipe responde a três perguntas:

- O que você fez neste projeto desde a última reunião diária?
- O que você planeja realizar neste projeto entre hoje e a próxima reunião diária?
- Que impedimentos existem para que você atinja seus compromissos para essa Sprint e para o projeto?

O objetivo dessa reunião é sincronizar diariamente o trabalho de todos os membros da equipe e agendar as reuniões necessárias para que a equipe avance nas atividades.

No final da Sprint, uma reunião de avaliação é realizada (Sprint Review Meeting). Essa reunião também é limitada por tempo, não devendo passar de quatro horas de duração. Nessa reunião, a equipe apresenta o que foi desenvolvido durante a Sprint para o Product Owner e outras partes interessadas que queiram participar. Essa reunião tem o objetivo de reunir as pessoas para que, colaborativamente, ajudem a Equipe a determinar o que deve ser feito em seguida.

Após a apresentação da Sprint (e antes da reunião de planejamento da próxima Sprint), o Scrum Master conduz uma reunião de revisão ou de avaliação da Sprint com a equipe – Retrospective Meeting. Novamente, trata-se de uma reunião com tempo fixo, que não deve ultrapassar três horas de duração. Nessa reunião, o Scrum Master encoraja o time a revisar o processo de desenvolvimento realizado, tendo em vista as práticas e os processos do framework Scrum, de forma a torná-lo mais efetivo e agradável para a equipe na próxima Sprint.



## Referências

---

AGILE MANIFESTO. *Home*. 2001. Disponível em: <<http://agilemanifesto.org/>>. Acesso em: 18 fev. 2021.

FONSECA, I. ALVES, M. e ALVES, F. *Ideal Day e Priorização?*. 6. ed. ESM, 2008.

IMPLEMENTING SCRUM. *Cartoons Scrum*. 2020. Disponível em: <<https://www.implementingscrum.com/cartoons.html>>. Acesso em: 18 fev. 2021.

SCHWABER, Ken. *Agile Project Management with SCRUM*. Microsoft Press, 2004.

SCRUM.ORG. *Home*. 2020. Disponível em: <<https://www.scrum.org/>>. Acesso em: 18 fev. 2021.

SCRUMALLIANCE. *Home*. 2020. Disponível em: <<http://www.scrumalliance.org>>. Acesso em: 18 fev. 2021.

SUTHERLAND, J.; *Scrum: A arte de fazer o dobro do trabalho na metade do tempo*. Leya: 2014.

TAKEUCHI, H.; NONAKA, I. *The New New Product Development Game*. Harvard BusinessReview, 1986.