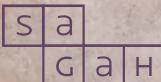


PROCESSOS DE DESENVOLVIMENTO DE SOFTWARE

Carlos Alessandro Bassi Viviani



SOLUÇÕES
EDUCACIONAIS
INTEGRADAS

Scrum e o planejamento

Objetivos de aprendizagem

Ao final deste texto, você deve apresentar os seguintes aprendizados:

- Discutir sobre as estimativas em planejamento de projetos ágeis.
- Analisar a técnica *planning poker* e sua aplicação em Scrum.
- Explicar a técnica de estimativa por escala corporativa e sua aplicação no Scrum.

Introdução

Neste capítulo, você vai entender os principais processos que envolvem a definição de estimativas em planejamento de projetos ágeis. Todos os sistemas de gerenciamento de projetos utilizam ferramentas e estratégias para definição de estimativas para a conclusão do projeto, e, quando estamos falando de projetos ágeis, essas ferramentas são ainda mais importantes, pois necessitam de um forte alinhamento estratégico que seja eficaz e efetivo.

Uma das técnicas mais utilizadas no Scrum é o *planning poker*, que nada mais é do que uma estratégia lúdica de estimar a complexidade de cada atividade dentro do projeto. Essa estratégia é realizada diretamente pela equipe que desenvolverá o produto, sendo assim, as estimativas tendem a refletir com mais exatidão a realidade.

Além do método citado, existem outros métodos mais objetivos que são utilizados pelas equipes para definir estimativas mais precisas, e geralmente são utilizados para composição de contratos entre a empresa desenvolvedora e o cliente. Um exemplo a ser explicado é a estimativa por escala corporativa que utiliza o princípio do ponto por função.

1 Estimativas em planejamento de projetos ágeis

Um passo fundamental no gerenciamento de qualquer projeto de *software* é definir o conjunto de atividades que permitam estimar o tempo e o custo necessários para completar o projeto do *software* (FENTON; PFLEEGER, 1997). Segundo Jones (1998), as medidas de tamanho de *software* surgiram com o objetivo de estimar o esforço (por exemplo: número de pessoas por hora de trabalho), o prazo e o custo envolvidos no desenvolvimento dos *software*. A principal medida utilizada era a quantidade de linhas de código fonte (SLOC — *Source Line of Code*), sendo considerada uma medida física do tamanho do *software* (KAN, 1995). Embora essa medida possa ser útil em muitos aspectos, ela possui diversas limitações que levaram à criação de outras medidas que procuram de fato medir a funcionalidade do *software*, e não o seu tamanho; essas medidas são conhecidas como medidas funcionais de tamanho do *software* (WEBER; ROCHA; NASCIMENTO, 2001).

Pressman (1995) destaca as principais técnicas de medidas funcionais utilizadas em modelos de estimativas do tempo e do custo do projeto do *software*. São elas: a Análise por Pontos de Função (FPA — *Function Point Analysis*), Pontos por Caso de Uso (UCP — *Use Case Points*) e Modelo de Custo Construtivo (COCOMO — *Constructive Cost Model*).

Para a definição de estimativas são utilizadas métricas para as medições. As medições foram criadas com a finalidade de garantir que indicativos pudessem ser obtidos e assim se pudesse otimizar os custos de produção, já que na década de 90 bilhões de dólares eram gastos em *software* que não atendiam às empresas da época (PRESSMAN, 2011).

Podemos destacar duas categorias de métricas (DCONTE; DUNSMORE; SHEN, 1986):

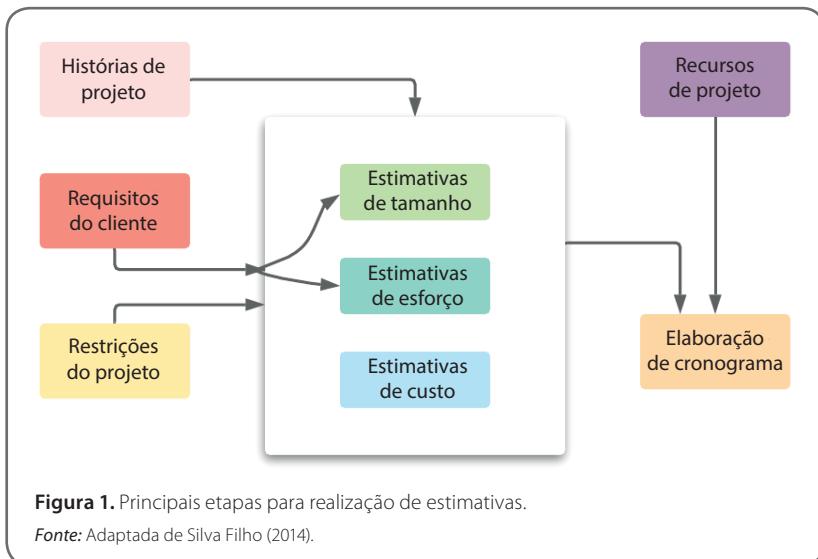
- de processo, que quantificam os atributos do processo de desenvolvimento e do ambiente de desenvolvimento (por exemplo, o custo do desenvolvimento);
- do produto, que são medidas do produto de *software*.

De acordo com o Institute of Electrical and Electronics Engineers (IEEE) (1990), as métricas podem ser separadas através dos aspectos a seguir.

- Atributo: propriedade física ou abstrata mensurável de um objeto.

- Medição: ação ou processo de atribuir um número ou uma categoria a uma entidade para descrever aquele objeto.
- Medida: um número, extensão ou quantidade que resulta de uma medição.
- Medir: aplicar uma métrica, ou atribuir valor por comparação com uma norma.

A Figura 1 a seguir apresenta as principais etapas para a realização de estimativas de projetos.



Com o uso das metodologias ágeis, é possível usar técnicas para estimar e planejar características dos projetos, sobretudo modelos não baseados em algoritmos. Muitas técnicas de estimativa ágil usam “histórias de usuários” (*user story*), as quais foram empregadas pela primeira vez por *eXtreme Programming* (XP) (BECK; ANDRES, 2004 *apud* ARANHA; CARDOSO, 2017) e popularizadas por Cohn (2004 *apud* ARANHA; CARDOSO, 2017). Podemos citar como técnicas *planning poker*, *the planning game*, *ideal day* e análise por ponto de função (APF) (ARANHA; CARDOSO, 2017).



Fique atento

XP é um apelido utilizado para uma metodologia de desenvolvimento designada *Extreme Programming*, com foco em agilidade de equipes e qualidade de projetos, apoiada em valores como simplicidade, comunicação, *feedback* e coragem, que nos submetem ao reconhecimento de que XP é uma metodologia baseada em comportamentos e atitudes. Dessa forma, ela propicia que o projeto seja executado dentro do prazo e do orçamento, fazendo, então, com que o cliente fique satisfeito e a equipe de desenvolvimento não fique maluca por causa do projeto (MEDEIROS, 2006).

2 O *planning poker* e sua aplicação no Scrum

Existem diversas oportunidades para o uso de escalas de complexidade no dimensionamento e monitoramento de estimativas, construção e entregas. O *planning poker* não é uma técnica do Scrum, mas é uma alternativa para estimativas em um time ágil. Exige algum tempo extra, mas é divertida e possui ganhos no foco e atenção aos princípios e ao papel de cada integrante nesta importante tarefa inicial do planejamento do projeto.

O *planning poker* usa o conceito de complexidade. Para exercitar corretamente este conceito, existem baralhos com números baseados na série adaptada de Fibonacci, pois além das cartas especiais é incluído o número 1/2. A sequência faz referência a uma provável curva crescente de incerteza: quanto maior uma tarefa, maior a incerteza, aumentando de forma significativa seu tamanho em pontos. A série acaba impondo esta curva; por exemplo, maior que 3 é 5, e acima deste já é 8, depois 13 (BRASILEIRO, 2018).

A Figura 2 a seguir apresenta um exemplo de baralho utilizado para o *planning poker*. Nela podemos ver as cartas que são utilizadas na maior parte dos jogos de *planning poker*. Temos as cartas numéricas e as cartas especiais.



Figura 2. Modelo de baralho utilizado no *planning poker*.

Fonte: diPersika/Shutterstock.com.

Veja a seguir o significado das cartas numéricas (DUARTE, 2016):

- Carta 0: a tarefa não precisa ser feita por algum motivo, ou já está pronta.
- Carta ½ ou 1: a tarefa é muito simples, provavelmente levará menos de uma hora de desenvolvimento de qualquer da equipe.
- Carta 2: a tarefa é simples, provavelmente leve menos de um turno de trabalho, como uma manhã, por exemplo.
- Carta 3: a tarefa é simples mas trabalhosa e não deve ser subestimada, ocupando pelo menos um turno de trabalho, como uma tarde (geralmente é mais longa que uma manhã).
- Carta 5: a tarefa é de complexidade mediana, provavelmente tomando um dia de trabalho de um desenvolvedor, se não tiver impedimentos.
- Carta 8: a tarefa é complexa, vai demandar estudo aprofundado ou muito desenvolvimento, provavelmente tomando alguns dias da semana, como 2 ou 3 no máximo.
- Carta 13: a tarefa é muito complexa, vai demandar estudo moderado ou é apenas muito longa de desenvolver, levando em média uma semana de um desenvolvedor (5 dias úteis).

- Carta 20 em diante: a tarefa é complexa demais e não vale a pena ser estimada. Sugere-se quebrar a tarefa em tarefas menores que possam ser estimadas com mais exatidão.

O *planning poker* é uma dinâmica que envolve toda equipe, onde todos podem e devem se posicionar após bom entendimento do que é cada funcionalidade ou história de usuário (*user story*). Ao estimar, é necessário comparar com as outras estimativas anteriores, mantendo uma coerência e refazendo quando necessário. Para começar sugere-se que a equipe escolha a menor de todas as histórias, usualmente um pequeno cadastro, atribuindo 1 a ela, de forma a servir como referência para este início das estimativas.

Nos baralhos distribuídos por grandes empresas de *software* e treinamentos existem também cartas especiais com:

- interrogação (?) para quando não nos sentimos em condições de estimar;
- uma de “café” para pedir uma pausa, e em alguns casos cartas acima de 21 pontos;
- carta “infinito” (caso não exista em seu baralho, deve-se usar a carta 100), que significa “não temos como fazer esta tarefa, Sr.”;
- Scrum Master, ela é longa demais e não cabe em qualquer pipeline de desenvolvimento. Sugiro quebrarmos ela em tarefas menores ou dizer ao Product Owner que não é possível fazer (caso muito raro) (DUARTE, 2016).

No caso de *user stories*, devemos sempre nos questionar se não podem ser quebradas em histórias menores, posto que um dos princípios para estimativas ágeis é termos um nível de granularidade adequado. Por isso chamamos de *user stories*; é para cada uma relatar apenas uma interação do usuário. As regras para o *planning poker* são (DUARTE, 2016):

- cada integrante da equipe de desenvolvimento recebe um baralho de *planning poker*;
- é importante haver uma combinação do número a partir do qual devemos tentar quebrar em mais histórias;
- a equipe seleciona a menor história a ser estimada como “1”, para servir de referência no início;
- a partir desta história serão discutidas as outras de forma comparativa;
- escolhe-se uma história, pergunta-se se todos entenderam e pede-se para que escolham uma carta;

- quando todos escolheram a sua carta, pede-se que mostrem para os demais;
- se houver consenso, está estimado e outra história será analisada;
- se não houver consenso, qualquer um, mas em especial os extremos, devem justificar sua opção;
- após os argumentos ocorre nova rodada de escolha de cartas;
- o limite hipotético é de 3 rodadas;

Caso o impasse se mantenha, existem duas saídas mais usadas: a escolha do maior valor, ou mesmo tirar a média dos valores colocados e fazer sempre uma aproximação para cima.

O *planning poker* é uma técnica que valoriza o senso de pertencimento, aprendizado contínuo, multidisciplinaridade e consenso. Estimula a interação da equipe. Destaca eventuais más interpretações e conflitos de percepção através do debate entre profissionais com diferentes *expertises*. Vale a pena tentar, mas não é mágico. Nas primeiras vezes podemos errar, mas com o tempo erra-se menos. Se passarmos a acertar sempre, desconfie, porque estimar *software* é uma tarefa complexa, e é mais provável que isso seja um sinal de que estamos com uma boa e generosa margem.



Fique atento

Você sabe o que é a série de Fibonacci? É uma sucessão de números que, misteriosamente, aparece em muitos fenômenos da natureza. Foi apresentada no final do século XII pelo italiano Leonardo Fibonacci, ela é infinita e começa com 0 e 1. Os números seguintes são sempre a soma dos dois números anteriores. Portanto, depois de 0 e 1, vêm 1, 2, 3, 5, 8, 13, 21, 34...

Ao transformar esses números em quadrados e dispô-los de maneira geométrica, é possível traçar uma espiral perfeita, que também aparece em diversos organismos vivos. Outra curiosidade é que os termos da sequência também estabelecem a chamada “proporção áurea”, muito usada na arte, na arquitetura e no *design*, por ser considerada agradável aos olhos. Seu valor é de 1,618 e, quanto mais você avança na sequência de Fibonacci, mais a divisão entre um termo e seu antecessor se aproxima desse número (SAHD, 2020).

3 Técnica de estimativa por escala corporativa e sua aplicação no Scrum

As técnicas de estimativa por escala corporativa são baseadas na APF. As equipes precisam entender a complexidade de um projeto através da definição de estimativas. As estimativas são definidas para que as equipes consigam atender aos prazos com qualidade, e são definidas a partir de técnicas de análise do projeto de desenvolvimento de *software*, como o *planning poker*, que foi explicado anteriormente.

Com a imprecisão dessa técnica, entretanto, e para que os contratos possam ser realizados utilizando parâmetros mais objetivos, tem-se procurado outras abordagens mais confiáveis e estabelecidas no mercado. A APF é uma métrica bastante robusta e é utilizada por diversas empresas. A APF mede o tamanho funcional de um *software* a partir dos requisitos que o projeto possui. Esses requisitos, ou funcionalidades, são obtidos sobre o ponto de vista do usuário (IFPUG, 2010).

Com o emprego cada vez maior de tecnologias ágeis, algumas de suas vulnerabilidades se tornaram evidentes. Uma delas é a estimativa do tamanho do *software*. Para resolver esse problema, uma das estimativas que podem ser utilizadas é a APF, que pode ser empregada por equipes de desenvolvimento, *Product Owner* (Dono do Produto) e *stakeholder*. “A APF já vem sendo utilizada por equipe ágeis, de várias formas e com resultados variados. Estudos indicam que para melhorar a precisão das estimativas em metodologias ágeis a APF se sobressai” (AGUIAR; SYMONS; VLIET, 2017; IFPUG, 2017, *apud* GOMES, 2017, p. 15).

Estimativa baseada em critérios objetivos

Pham e Pham (2011) propõem uma técnica de estimativa por escala corporativa que foi aplicada com sucesso em diversos projetos. Esse método foi baseado em APF. Conforme Rodrigues (2014, p. 34):

Uma aplicação é formada por um usuário da área de negócio, utilizando um código funcional (aplicação), que contém as regras de negócio em um modelo que contém entidades de negócio, onde os dados são armazenados em um banco de dados que deve ser criado, lido, atualizado ou deletado.

A Figura 3 a seguir ilustra estes diferentes aspectos de uma aplicação.

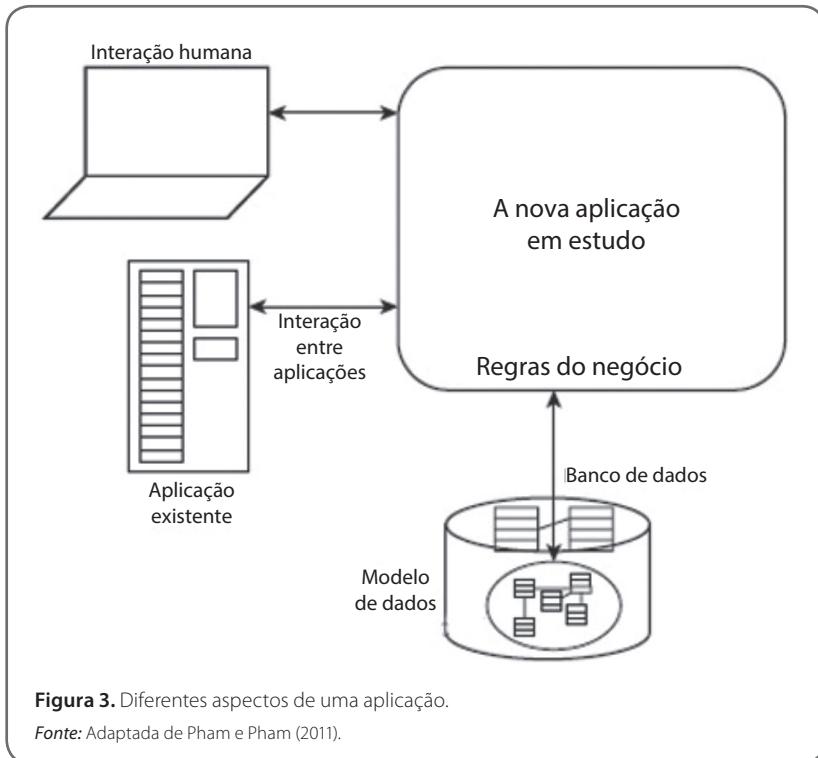


Figura 3. Diferentes aspectos de uma aplicação.

Fonte: Adaptada de Pham e Pham (2011).

Os principais pontos que são utilizados para estimar as histórias de usuário, também chamados de “pontos não ajustados”, ou “itens do *backlog* de produto”, são (PHAM; PHAM, 2011, *apud* RODRIGUES, 2014, p. 34):

- tipos de interação;
- regras de negócio;
- número de entidades manipuladas;
- dados a serem lidos, criados, atualizados e excluídos.

O Quadro 1 apresenta os critérios associados aos tipos de interação.

Quadro 1. Tipos de interação

Tipo de interação	Descrição	Valor
Simples	Interface bem definida	1
Média	Interface dinâmica	2
Complexa	Interface humana	3

Fonte: Adaptado de Pham (2011).

O Quadro 2 apresenta os critérios associados às regras do negócio.

Quadro 2. Regras de negócio

Regras de negócio	Descrição	Valor
Simples	Uma regra	1
Média	Duas a três regras	2
Complexa	Mais de três regras	3

Fonte: Adaptado de Pham (2011).

O Quadro 3 apresenta os critérios associados às entidades de negócio que serão manipuladas.

Quadro 3. Entidades

Regras de negócio	Descrição	Valor
Simples	Uma entidade	1
Média	Duas a três entidades	2
Complexa	Mais de três entidades	3

Fonte: Adaptado de Pham (2011).

O Quadro 4 apresenta os critérios associados aos dados que serão manipuladas.

Quadro 4. Tipos de manipulação de dados

Regras de negócio	Descrição	Valor
Simples	Ler, excluir	1
Média	Criar	2
Complexa	Atualizar	3

Fonte: Adaptado de Pham (2011).

Rodrigues (2014, p. 35–36) afirma:

Outros pontos que devemos levar em consideração são as dimensões de ambiente (DA), que podem [ter] impacto negativo ou positivo sobre a equipe durante o trabalho para entregar a tarefa:

- dimensão organizacional;
- dimensão de infraestrutura de desenvolvimento;
- dimensão de equipe;
- dimensão de tecnologia;
- dimensão de processo;
- dimensão de negocio.

Para cada uma das dimensões será definido um valor entre zero e dois, que indica a habilidade ou capacidade da equipe; quanto mais baixo o valor ou menor, a habilidade ou capacidade da equipe será menor. Dessa forma, como foi apresentado nas tabelas anteriores, é necessário percorrer todas as dimensões e verificar qual o valor será atribuído para as questões.

O Quadro 5 apresenta a síntese de questões a serem avaliadas por dimensão.

Quadro 5. Questões por dimensão

1. Dimensão organizacional	
Fator	Escopo valor (0/2)
Possui diferentes departamentos que trabalham previamente, em conjunto e com sucesso, em um projeto com metodologia Scrum?	
Há alguma resistência forte dentro da empresa em relação ao Scrum?	
Existe um grande suporte ao Scrum entre diferentes departamentos da empresa?	
2. Dimensão infraestrutura de desenvolvimento	
Fator	Escopo valor (0/2)
Já existem testes automáticos e eles são uma prática comum?	
Já existem testes de integração e eles são uma prática comum?	
Já existe um ambiente de construção diária e ele é uma prática comum?	
3. Dimensão de equipe	
Fator	Escopo valor (0/2)
A equipe nunca utilizou antes o Scrum?	
Os membros da equipe já trabalharam juntos e foram bem-sucedidos anteriormente?	
Os membros da equipe se conhecem bem e apreciam a companhia uns dos outros?	

(Continua)

(Continuação)

Quadro 5. Questões por dimensão

4. Dimensão de tecnologia	
Fator	Escopo valor (0/2)
A equipe de desenvolvimento possui bastante experiência na linguagem de programação?	
Os membros da equipe de desenvolvimento possuem bastante experiência na tecnologia a ser empregada?	
O ambiente de produção Scrum já está pronto?	
5. Dimensão de processo	
Fator	Escopo valor (0/2)
O Scrum é a estrutura de processo adotada pela empresa?	
Existe um bom suporte para Scrum dentro da empresa?	
Existe alguma forte resistência ao Scrum dentro da empresa?	
6. Dimensão de negócio	
Fator	Escopo valor (0/2)
Existe um Product Owner completamente disponível e engajado com a equipe?	
O Product Owner está familiarizado com o Scrum, mas não possui experiência prática?	
O Product Owner já usou o Scrum anteriormente com sucesso?	

Fonte: Adaptado de Pham (2011).

Dependendo desse valor total das questões associadas às 6 dimensões, existem três cenários possíveis (PHAM; PHAM, 2011, *apud* RODRIGUES, 2014, p. 38). Confira a seguir.

1. Se o valor de DA ficar entre 0 e 11, o coeficiente de multiplicação C será 2. Isso significa que as DA não são favoráveis e a equipe não será

capaz de entregar tantas histórias durante uma *Sprint* quanto se o valor das DA fosse maior.

2. Se o valor de DA ficar entre 12 e 23, o coeficiente de multiplicação C será 1. Isso significa que o ambiente não facilita nem dificulta o trabalho da equipe.
3. Se o valor de DA ficar entre 24 e 36, o coeficiente de multiplicação C será 1/2. Isso significa que as DA são favoráveis e a equipe será capaz de entregar mais histórias durante uma *Sprint*.

Para calcular o valor total de uma única história em pontos, usaremos a seguinte fórmula:

$$\text{PA (Pontos ajustados)} = \text{PNA (Pontos não ajustados)} \times C (\text{Coeficiente})$$

$$\text{PPH (Pontos por história)} = (\text{PA} \times \text{DA}) / 36$$

Um dos problemas na implantação ou extensão do Scrum a todo o departamento de TI de uma empresa é infelizmente o fato que, o valor dos pontos de história não ser comparável entre as equipes. Com a velocidade variando tanto entre as equipes podemos ver o porquê disso ser um problema na implantação da metodologia Scrum em todos os setores da empresa (PHAM, 2012 *apud* RODRIGUES, 2014, p. 38).

Exemplo

Vamos apresentar um exemplo da contagem dos fatores ambientais que podem influenciar a contagem das estimativas de qualquer projeto.

A empresa Acme pretende implantar a metodologia Scrum para desenvolvimento de seus projetos. Já existem algumas ações isoladas que trabalham com Scrum de forma segmentada. Para realizar um novo projeto, a equipe de desenvolvimento pretende estimar quanto os fatores ambientais poderão impactar na entrega com a adoção do Scrum. Para estimar os fatores ambientais, o seguinte questionário foi respondido:

1. Dimensão organizacional	
Fator	Escopo valor (0/2)
Possui diferentes departamentos que trabalham previamente, em conjunto e com sucesso, em um projeto com metodologia Scrum?	1
Há alguma resistência forte dentro da empresa em relação ao Scrum?	0
Existe um grande suporte ao Scrum entre diferentes departamentos da empresa?	1
2. Dimensão infraestrutura de desenvolvimento	
Fator	Escopo valor (0/2)
Já existem testes automáticos e eles são uma prática comum?	0
Já existem testes de integração e eles são uma prática comum?	1
Já existe um ambiente de construção diária e ele é uma prática comum?	1
3. Dimensão de equipe	
Fator	Escopo valor (0/2)
A equipe nunca utilizou antes o Scrum?	0
Os membros da equipe já trabalharam juntos e foram bem-sucedidos anteriormente?	1
Os membros da equipe se conhecem bem e apreciam a companhia uns dos outros?	2
4. Dimensão de tecnologia	
Fator	Escopo valor (0/2)
A equipe de desenvolvimento possui bastante experiência na linguagem de programação?	2
Os membros da equipe de desenvolvimento possuem bastante experiência na tecnologia a ser empregada?	2
O ambiente de produção Scrum já está pronto?	1

(Continua)

(Continuação)

5. Dimensão de processo	
Fator	Escopo valor (0/2)
O Scrum é a estrutura de processo adotada pela empresa?	1
Existe um bom suporte a Scrum dentro da empresa?	1
Existe alguma forte resistência ao Scrum dentro da empresa?	0
6. Dimensão de negócio	
Fator	Escopo valor (0/2)
Existe um Product Owner completamente disponível e engajado com a equipe?	1
O Product Owner está familiarizado com o Scrum, mas não possui experiência prática?	1
O Product Owner já usou o Scrum anteriormente com sucesso?	1

Fonte: Adaptado de Pham (2011).

A somatória total deste questionário foi 17. Isso significa que o ambiente não facilita nem dificulta o trabalho da equipe. Portanto a estimativa para esse projeto será realmente baseada na complexidade da tarefa, pois os fatores ambientais não influenciarão neste caso.

As definições das estimativas em projetos são imprescindíveis para a gestão de todo processo. Em projetos ágeis as estimativas podem ser calculadas de maneiras mais lúdicas como o *planning poker*, para projetos menores e através de métodos baseados em pontos de função em projetos de escala corporativa.

Neste capítulo foi possível entender como funcionam as estimativas em projetos ágeis de *software* e como é possível implementá-los em projetos reais em sua corporação.



Referências

- ARANHA, D. P.; CARDOSO, M. O. *Métodos e métricas para estimativas e planejamento de projetos ágeis de software*: estudo comparativo entre pontos de função e story points. 2017. 65 p. Trabalho de Conclusão de Curso (Engenharia de Software) — Universidade de Brasília (UnB)/Faculdade UnB Gama (FGA), Brasília, 2017. Disponível em: https://bdm.unb.br/bitstream/10483/20412/1/2017_DandaraAranha_MaxwellCardoso_tcc.pdf. Acesso em: 7 jul. 2020.
- BRASILEIRO, R. *Planning poker*: a melhor maneira de estimar qualquer atividade. 2018. Disponível em: <http://www.metodoagil.com/planning-poker/>. Acesso em: 4 jul. 2020.
- DCONTE, S. D.; DUNSMORE, H. E.; SHEN, V. Y. *Software engineering metrics and models*. Boston, MA: Addison-Wesley, 1986.
- DUARTE, L. *Planning poker*: como estimar tempo de desenvolvimento de software. 2016. Disponível em: <https://www.luiztools.com.br/post/planning-poker-como-estimar-tempo-de-desenvolvimento-de-software/>. Acesso em: 4 jul. 2020.
- FENTON, N. E.; PFLEIDERER, S. L. *Software metrics*: a rigorous and practical approach. 2nd ed. Boston: PWS Publishing, 1997.
- GOMES, L. C. P. *Scrum Root*: ferramenta para integrar APF e Scrum. 2017. 85 p. Trabalho de Conclusão de Curso (Tecnólogo em Análise de Desenvolvimento de Sistemas) — Universidade Tecnológica Federal do Paraná, Ponta Grossa, 2017. Disponível em: http://repositorio.roca.utfpr.edu.br/jspui/bitstream/1/8344/1/PG_COADS_2017_2_08.pdf. Acesso em: 4 jul. 2020.
- INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS – IEEE. *IEEE Standard Glossary of Software Engineering Terminology*. New York: IEEE, 1990.
- JONES, C. *Estimating software costs*. New York: McGraw-Hill, 1998.
- KAN, S. H. *Metrics and models in software quality engineering*. Boston, MA: Addison-Wesley, 1995.
- MEDEIROS, M. P. *Extreme programming*: conceitos e práticas. 2006. Disponível em: <https://www.devmedia.com.br/extreme-programming-conceitos-e-praticas/1498>. Acesso em: 4 jul. 2020.
- PHAM, A.; PHAM, P.-V. *Scrum em ação*: gerenciamento e desenvolvimento ágil de projetos de software. Santos, SP: Novatec, 2011.
- PRESSMAN, R. S. *Engenharia de software*. São Paulo: Makron Books, 1995.
- PRESSMAN, R. S. *Engenharia de software*. 7. ed. Porto Alegre: AMGH, 2011.
- RODRIGUES, R. S. *Desenvolvendo software com a metodologia ágil*: Scrum. 2014. 80 p. Trabalho de Conclusão de Curso (Bacharelado em Sistemas de Informação) — Centro

Universitário Eniac, Guarulhos, 2014. Disponível em: <http://bnportal.eniac.com.br/scripts/bnportal/bnportal.exe/upload?arquivo=4126>. Acesso em: 4 jul. 2020.

SAHD, L. O que é a sequência de Fibonacci? *Mundo Estranho*, 14 fev. 2020. Disponível em: <https://super.abril.com.br/mundo-estranho/o-que-e-a-sequencia-de-fibonacci/>. Acesso em: 4 jul. 2020.

SILVA FILHO, A. M. Estimativa de custo de software: roteiro e dicas para estimativas de projeto. *Revista Espaço Acadêmico*, v. 13, n. 156, maio 2014. Disponível em: <http://www.periodicos.uem.br/ojs/index.php/EspacoAcademico/article/download/23850/12975/>. Acesso em: 4 jul. 2020.

WEBER, K. C.; ROCHA, A. R. C.; NASCIMENTO, C. J. *Qualidade e produtividade em software*. São Paulo: Makron Books, 2001.



Fique atento

Os *links* para *sites da web* fornecidos neste capítulo foram todos testados, e seu funcionamento foi comprovado no momento da publicação do material. No entanto, a rede é extremamente dinâmica; suas páginas estão constantemente mudando de local e conteúdo. Assim, os editores declaram não ter qualquer responsabilidade sobre qualidade, precisão ou integralidade das informações referidas em tais *links*.

Encerra aqui o trecho do livro disponibilizado para esta Unidade de Aprendizagem. Na Biblioteca Virtual da Instituição, você encontra a obra na íntegra.

Conteúdo:



SOLUÇÕES
EDUCACIONAIS
INTEGRADAS