

(E)

(1) 1/1

Rafael Maiani de Wello

Tema 1

Os processos ágeis de software foram particularmente impulsionados a partir do Manifesto Ágil, cerca de duas décadas atrás. Neste manifesto, princípios como KISS e YAGNI foram usados como inspiração, levando a comunidade de software a refletir sobre suas prioridades e uma visão de processo predominantemente descritiva.

Neste sentido, os processos ágeis de software incluem métodos compostos por determinadas práticas que se apresentam com possíveis respostas aos anseios do Manifesto Ágil e de seus signatários e apoiadores. Dentre estes métodos, podemos destacar o XP e o Scrum.

O Scrum é um método focado nas ações de comunicação e gestão do projeto de desenvolvimento de um software, sendo atualmente largamente adotado como referência na definição de processos de desenvolvimento em projetos na indústria.

Inspirado por valores como coragem e feedback, o ciclo do Scrum é fundamentado por conceitos-chave como sprint e backlog do produto. Associado a este ciclo, podemos destacar cerimônias como o planejamento do sprint,

as reuniões diárias e uma reunião ao encerramento de ² μ cada sprint para coleta de feedback e ações de melhoria para o próximo sprint.

Um sprint consiste em cada ciclo de desenvolvimento que visa levar à entrega de um novo incremento funcional do sistema em desenvolvimento. Os sprints são planejados com base no backlog do produto, que é composto pelo conjunto de funcionalidades a serem implementadas no sistema, tipicamente a partir de histórias de usuário sendo definidas. Sprints devem ser de curta duração, normalmente levando entre duas a três semanas.

Ao longo do ciclo do Scrum, o Scrum Master é o indivíduo que atua como facilitador do processo, conduzindo as cerimônias previstas com o time de desenvolvimento e o Product Owner (PO). O PO possui, por sua vez, um papel fundamental no processo, representando o cliente e suas prioridades ao longo do processo.

O Extreme Programming ou XP é outro método ágil consideravelmente difundido, fundamentado em um conjunto de práticas que visam estimular a entrega contínua de novas funcionalidades ao cliente. Para tal, o cliente (representante) deve estar presente e atuando com o time de desenvolvimento ao longo de todo o processo.

3

No XP, é prevista a prática de integrações contínuas do trabalho diário do time de desenvolvimento, que devem se auto-organizar implementando o código em pares. Além disto, são também previstas reuniões diárias (em pé) no time de desenvolvimento, visando deixar todos a par de suas atividades e eventuais obstáculos. Estas reuniões devem ser de curta duração, preferencialmente não passando de 15 minutos de duração.

No XP, o time de desenvolvimento possui a propriedade coletiva do código, compondo um time completo de desenvolvimento que deve zelar por um ritmo sustentável.

Na prática, é comum observar organizações e projetos adotando versões híbridas ou adaptadas de métodos ágeis como o Scrum e o XP. Por exemplo, é comum observar processos que adotam parte das cerimônias Scrum ao mesmo tempo que recomendam práticas específicas do XP.

Outro exemplo inclui a combinação de técnicas mais formais de desenvolvimento, como a especificação de casos de uso, para a definição de um backlog do produto dentro de um ciclo de vida baseado no Scrum.

Tópico (3)

Casos de uso é uma técnica de especificação de (4)^{xy} requisitos centrada em uma descrição organizada das interações entre um ator e um sistema.

A partir da elicitação dos requisitos de um sistema, o analista de requisitos pode conduzir reuniões com o cliente e demais partes interessadas para detalhar os casos de uso modelados. Considerando o rigor da especificação, é comum que os casos de uso sejam parte do documento de requisitos do sistema, indicando detalhadamente do ponto de vista do usuário o que deve ser implementado e eventuais restrições.

Para a modelagem de casos de uso, é fundamental ter em mente os conceitos de ator e transação. Um ator endossa um possível papel exercido sobre o uso de um sistema. Apesar de comumente atores representarem usuários e grupos de usuários humanos específicos, é comum que outras entidades, como outros sistemas, sensores e atuadores também sejam atores. Em todo caso de uso, deve haver sempre um único ator primário, que visa alcançar algum objetivo com o uso do sistema.

Neste sentido, uma transação descreve uma interação única e completa entre o ator primário e o sistema. Esta transação deve ser iniciada através de um estímulo do ator (ação) que deverá ser respondido pelo sistema (ação do sistema). Eventualmente, antes de responder ao ator, o sistema pode realizar algum processamento, como a verificação de alguma regra de negócio.

A sequência de transações entre ator e sistema deve resultar em um caso de uso completo, que deve assim possuir ao menos seu fluxo principal. Entretanto, um caso de uso também pode permitir que o ator primário desvie do fluxo principal, seguindo um ou mais fluxos alternativos. Por sua vez, o sistema também pode apresentar desvios do fluxo principal, que são conhecidos como fluxos de exceção. Na descrição de um caso de uso, é importante que as regras de negócio ~~em~~ aplicadas sejam identificadas. Também é recomendável mapear os requisitos não-funcionais específicos de cada caso de uso.

A seguir, apresentamos o exemplo de um modelo de caso de uso, representado a partir do diagrama de caso de uso da UML. Em seguida, apresentamos ~~uma~~ a descrição do caso de uso "Fazer pedido".

Requisitos Funcionais

6

xy

RFO1.	O sistema deve permitir que o cliente faça pedidos na plataforma
RFO.2	O sistema deve permitir que o cliente pague seus pedidos



CASO DE USO UC01: FAZER PEDIDO	
ATOR: CLIENTE	
OBJETIVO: PERMITIR QUE UM CLIENTE FAÇA UM PEDIDO NO SISTEMA	
PRÉ-CONDICÃO: O CLIENTE DEVE ESTAR AUTENTICADO NO SISTEMA	
PÓS-CONDICÃO: O CLIENTE REALIZA SEU PEDIDO, EFETUANDO SEU PAGAMENTO	
ATOR	SISTEMA
1. SOLICITA FAZER UM PEDIDO	2. SOLICITA INFORMAR O ITEM E A QUANTIDADE
3. INFORMA O ITEM E A QUANTIDADE	4. PERGUNTA SE O CLIENTE DESEJA INCLUIR UM NOVO ITEM [A1: INCLUIR NOVO ITEM]
5. INFORMA QUE NÃO DESEJA INCLUIR NOVO ITEM	[INCLUDE: PAGAR PEDIDO]
	FIM DO CASO DE USO
[A1: INCLUIR NOVO ITEM]	RETORNA AO PASSO 2 DO FL. PRINCIPAL

Tópico 9

A contagem de pontos de função é baseada na abordagem APF, que visa estimar o tamanho funcional de um sistema em desenvolvimento ou já desenvolvido.

Para calcular este tamanho funcional, é preciso definir a fronteira da aplicação, o que consiste em identificar os dados e funções tratados e armazenados pelo sistema, diferenciando-os daqueles que são externos a esta fronteira.

Na prática, a contagem de pontos de função parte, portanto, da definição desta fronteira e do objetivo da contagem (novo sistema, manutenção, sistema pronto). Em seguida, é necessário identificar os conjuntos de dados que cruzam a fronteira e são consumidos pelo sistema (AIE - arquivos de interface externa) e os dados mantidos dentro da fronteira da aplicação (ALI - arquivos lógicos internos).

Para a contagem de pontos de função, também é necessário identificar todas as funções transacionais a serem realizadas pelo sistema a partir do ponto de vista de seus usuários. Assim, três tipos de funções transacionais são possíveis:

- Entradas Externas (EE) - funções em que dados são incluídos/modificados/removidos de uma ALI.
- Consultas Externas (CE) - referente a consultas de dados sem que haja alguma manipulação (transformação) nos dados recuperados.
- Saídas Externas (SE) - referente a consultas de dados que envolvam uma ou mais transformações nos dados para apresentação ao usuário.

Uma vez identificados estes elementos, é necessário contabilizar os itens de dados e arquivos que os compõem. Conforme os intervalos de tamanho em que estes elementos se inserem, digo, inserem, a APF irá determinar um valor. Quanto maior este valor, maior seria a complexidade daquele arquivo ou transformação. Por exemplo, consideremos um módulo de cadastro de clientes de um sistema que comunica-se com a base de inadimplentes do SERASA. Assim, poderíamos ter:

- | |
|---|
| 1 |
|---|

 ALI → CLIENTES (NOME, CPF, ENDEREÇO, TELEFONE, CÊNGRO)
- | |
|---|
| 5 |
|---|

 ITENS DE DADOS
- | |
|---|
| 1 |
|---|

 A/E → SITUAÇÃO-CLIENTES (CPF, SITUAÇÃO, ~~DATA~~ DATA)
- | |
|---|
| 3 |
|---|

 ITENS DE DADOS

- (9) jul
- [3] EE : INCLUIR CLIENTE → 6 ITENS DE DADOS, 1 ALI
 ALTERAR CLIENTE → 6 ITENS DE DADOS, 1 ALI
 EXECUIR CLIENTE → 2 ITENS DE DADOS, 1 ALI

Nas entradas externas, é importante ter em mente que eventuais itens de navegação também podem ser contados quando fazem parte da transação.

- [1] CE : CONSULTAR CLIENTE → 5 ITENS DE DADOS,
 1 ALI

- [1] SE : CONSULTAR INADIMPLENTES → 4 ITENS DE DADOS,
 1 ALI, 1 AIE
 (2 ARQUIVOS)

Aqui, cabe ressaltar que "consultar clientes" é definido como consulta externa porque simplesmente reproduz os dados recuperados da ALI. Já "consultar inadimplentes" requer uma transformação da "data" da "Situação-Clientes" (AIE), pois anota ~~o~~ quantos dias o cliente está inadimplente. Ou seja, para o usuário, não é o dado original sendo após ~~sentado~~. Para cada elemento identificado no exemplo, a tabela de referência APF irá indicar seu tamanho funcional, medido em APF. A soma de todos os APF contabilizados em uma aplicação definem seu tamanho não-ajustado. A partir das respostas a um questionário de fatores de ajuste, será possível obter finalmente o total de APF ajustado.

10

Até hoje, a contagem de pontos de função tem sido adotada com métrica em contratos e licitações. Um grande desafio é efetivamente estimar o esforço envolvido na implementação de TAPF. Alguns levantamentos com base histórica de projetos sugerem intervalos de homem-hora para se implementar TAPF em tecnologias específicas. Por outro lado, existem críticas quanto a atualizações e a considerável subjetividade envolvida no cálculo dos fatores de ajuste.

AP