

8 Conclusão

O principal objetivo desse trabalho foi estudar formas de se utilizar as especificações de testes como especificações do sistema a ser construído. Para tal foram empregadas as técnicas de desenvolvimento dirigido por comportamentos, que propõem a utilização de linguagens de especificação de testes mais próximas do domínio e divisão das unidades de teste com base nos comportamentos desejados. Para a utilização de linguagens mais próximas do domínio foi construída uma ferramenta capaz de utilizar diversas linguagens de entrada. Essa ferramenta foi utilizada para a aplicação de uma linguagem baseada em máquinas de estado em um projeto real, para que fosse possível verificar a utilidade das técnicas de desenvolvimento dirigido por comportamentos.

Podemos classificar os *frameworks* analisados nesse trabalho com relação à linguagem de escrita dos testes em duas categorias distintas:

- A primeira categoria é a dos *frameworks* que possuem as especificações dos testes escritas em código nativo da ferramenta, tornando mais difícil que integrantes do projeto sem conhecimentos técnicos os leiam e revisem.
- A segunda categoria é a dos que apresentam arquivos textuais, os *scripts*, contendo as especificações dos testes, separados do código responsável por servir de interface com os módulos sob teste, os módulos específicos de testes. Esta categoria representa uma alternativa melhor para manter a especificação dos comportamentos em um arquivo com maior densidade de informações e de mais fácil leitura e localização

Entretanto, mesmo os *frameworks* que se enquadram na segunda categoria geralmente apresentam linguagens em que pouca ou nenhuma alteração é possível. Alguns dos *frameworks* também não apresentam formas suficientemente boas de apresentar os dados de saída. A maior evolução deste *framework* com

relação aos pesquisados é a possibilidade de realizar modificações na linguagem de especificação dos casos de teste.

Muitos dos *frameworks* observados já possuíam a capacidade de alterar a forma como os resultados são exibidos, entretanto, embora tal característica também seja bastante importante para este trabalho, apenas permitindo modificações também nas entradas é possível realizar a sua integração com outras ferramentas. Com estas capacidades de personalização, projetos que se beneficiem de formas de modelagem específicas e distantes dos formatos atualmente propostos pelas ferramentas de desenvolvimento dirigido por testes e de desenvolvimento dirigido por comportamentos podem realizar sua especificação de testes em linguagens mais próximas do domínio da aplicação. Isso acarretaria também em redução do custo para a co-evolução de diversos documentos de especificação através da utilização da própria especificação de testes como documentação do sistema que será construído. No trabalho, a modelagem utilizando máquinas de estado gráficas foi suficientemente clara para ser utilizada como linguagem de especificação do sistema, porém, a linguagem textual baseada em máquinas de estado proposta não foi suficientemente clara, o que tornou necessária a co-evolução entre documentos de modelagem e de especificação dos testes. A utilização de uma linguagem mais gráfica como linguagem de especificação da ferramenta provavelmente resolveria essa questão.

Uma das preocupações no início do trabalho era com o esforço necessário para a criação dos novos módulos de extensão da ferramenta. Para que tal esforço seja realizado, é necessário que existam ganhos reais ao utilizar tais módulos, e, evidentemente, também é necessário que as organizações percebam essas vantagens. Este esforço inicial pode ser um grande empecilho à adoção desta ferramenta. Entretanto, vencida esta barreira, existe a possibilidade de organizações criarem repositórios de módulos de forma a tornar muito mais simples a utilização de novas instâncias através do reuso de módulos antigos.

A criação do módulo de máquinas de estados se mostrou simples o suficiente para que as vantagens obtidas com a nova linguagem justificassem sua utilização. A concentração de falhas observadas antes da entrega dos componentes provavelmente pode ser creditada ao maior cuidado na etapa de especificação. Esses resultados levaram a crer que as alterações no processo decorrentes da utilização da ferramenta foram de fato eficazes para causar reduções nos custos de

desenvolvimento das novas funcionalidades. Entretanto, como anteriormente nenhuma forma de automatização de testes era realizada, provavelmente outras técnicas de automatização também teriam apresentado melhoras no processo.

A contribuição da especificação para as etapas de levantamento de requisitos das novas funcionalidades, entretanto, provavelmente foi mais significativa por conta das técnicas de desenvolvimento dirigido por comportamentos. A modelagem dos comportamentos aparentemente ajudou a equipe a compreender melhor as necessidades dos clientes e fornecer soluções de melhor qualidade e mais rapidamente.