PONTIFÍCIA UNIVERSIDADE CATÓLICA DE GOIÁS
ESCOLA DE CIÊNCIAS EXATAS E DA COMPUTAÇÃO.
ANÁLISE E DESENVOLVIMENTO DE SISTEMAS.

CRITÉRIOS DE AVALIAÇÃO E AVALIAÇÃO DE FERRAMENTAS

BRUNO CAMARGO MANSO
JOÃO VICTOR CARDOSO DE OLIVEIRA
NIKOLLY CARDOSO DE FARIA
WESLEY SILVA CAIXETA

GOIÂNIA, GO
2020

BRUNO CAMARGO MANSO
JOÃO VICTOR CARDOSO DE OLIVEIRA
NIKOLLY CARDOSO DE FARIA
WESLEY SILVA CAIXETA

CRITÉRIOS DE AVALIAÇÃO E AVALIAÇÃO DE FERRAMENTAS

O presente trabalho contém estudos sobre critérios de avaliação de ferramentas por fim definir e avaliar um conjunto de ferramentas para desenvolvimento web.

Desenho de Software
Orientador: Aníbal Vicente Vieira

GOIÂNIA, GO
2020

Resumo

Neste trabalho, nos aprofundaremos sobre critérios de avaliação de ferramentas necessárias em processos de desenho de software. Por fim, vamos escolher um conjunto de ferramentas e aplicar tais critérios aprendidos.

Palavras-chave: Desenho de Software, critérios de avaliação, ferramentas.

Sumário

1. Introdução

O desenho de software é uma fase muito importante dentro do desenvolvimento de software, isso porque é necessário desenhar para facilitar o entendimento do que foi proposto e reduzir prováveis problemas futuros que podem ser identificados através do desenho de software. Um erro na produção do desenho, pode ser corrigido alterando o desenho, mas um erro na construção de software, pode significar desde prejuízo financeiro até rompimento de contrato e insatisfação do cliente.

Neste trabalho iremos realizar a avaliação de duas ferramenta de desenho de software. Primeiramente, serão apresentados 5 critérios para essa avaliação, sendo eles: funcionalidade, usabilidade, eficiência, portabilidade, confiabilidade e manutenibilidade. Estes critérios serão desenvolvidos no tópico 2, onde será explicado o que cada um significa e sua importância dentro da avaliação.

Foram escolhidas duas ferramentas muito competentes e utilizadas dentro desse meio de desenho de software, que são o Draw io e o Visual Paradigm, no tópico 3 se encontra detalhes sobre cada uma, assim como suas funcionalidades.

No tópico 4 é onde são aplicados os critérios sobre as ferramentas, elas passaram por avaliações e exibiremos seus respectivos resultados.

2. Critérios para avaliação de ferramentas de desenho de software web

Neste tópico, iremos aprender como seriam os critérios de avaliação para conceituar cada ferramenta proposta posteriormente. Iremos aprofundar nos conceitos sobre termos utilizados, com o objetivo de fazer uma análise comparativa ou, em termos técnicos, fazer o *benchmark* de duas ferramentas escolhidas pelo grupo.

O benchmark é muito utilizado a medida que ele têm a pretensão de comparar, a partir das características, duas ou mais ferramentas distintas. Com isso é possível perceber o custo benefício de cada uma delas, garantindo assim, por parte das empresas um melhor gerenciamento de custos e de tempo.

A partir das comparações efetuadas, decisões sobre o melhor software para esta empresa serão tomadas isso fará com que exista um alinhamento interno tanto sobre a parte executiva quanto da parte administrativa de uma empresa, em relação ao tipo e se tal ferramenta examinada cumprirá o objetivo pretendido.

As ferramentas serão mensuradas a partir dos termos referidos abaixo, trazendo com cada um desses termos uma pergunta, que deverá ser feita a medida que tais ferramentas são testadas.

2.1 - Funcionalidade

A Funcionalidade abrange a **adequação** da ferramenta, se esta ferramenta realmente segue o propósito por ela estabelecido em um projeto, no nosso caso, um projeto de desenho de software: Possui todas as ferramentas internas para o desenvolvimento do projeto?

Também abrange a **acurácia** dessa ferramenta, ou seja, se eles conseguem elaborar um modelo ou diagrama que consiga atender ao projeto: Irá atender o projeto como um todo?

Dentro das funcionalidades ainda temos a **interoperabilidade**, muitas ferramentas possuem a possibilidade de, por exemplo no caso dos diagramas UML, a partir de um diagrama, criar e exportar classes para serem usadas em orientação a objetos, é fundamental que estas ferramentas criem um ambiente que sejam devidamente exportados quando necessário e que funcionem de forma correta em sistemas diferentes. Ou seja: Os artefatos exportados irão funcionar em outros softwares de desenvolvimento?

A **conformidade** é parte das funcionalidades que devem ser observadas nos softwares escolhidos. Se o software corresponde à demanda da empresa: É o software que corresponde às expectativas dos desenvolvedores, bem como dos projetistas?

Segurança de Acesso é outra característica a ser observada, uma vez que projetos muitas vezes precisam de permissões de acesso, para evitar assim espionagens industriais, erros de modificação ou acesso indevido: O software escolhido tem controle de acesso?

2.2 - Usabilidade

A Usabilidade é outra importante característica a ser observada. É o que diferencia bons e maus softwares, uma vez que, uma usabilidade eficaz garante um melhor entendimento e maior curva de aprendizado por parte do usuário, consequentemente maior performance e menor tempo de elaboração.

Dentro da usabilidade então, podemos citar a **inteligibilidade** do software, como foi citado acima, se é intuitivo e fácil de entender: É fácil de entender a aplicação? É intuitivo?

Sua **apreensibilidade** também já supracitado é a capacidade de ser aprendido por seus usuários que de repente possam esquecer certas funcionalidades, em que menu se encontra etc, ou seja: É fácil de lembrar onde ficam as principais ferramentas? Existem muitas teclas de atalho, elas são complementares ao software ou devem ser usadas?

Operacionalidade, que também faz parte da usabilidade, se refere ao domínio necessário para efetuar determinadas tarefas. Logicamente que tarefas mais específicas precisam ser mais aprofundadas, como por exemplo os conceitos básicos de orientação a objetos e a linguagem Java, para que a ferramenta seja utilizada em sua plenitude, deve-se conhecer a fundo tais conceitos, não seria então propriamente uma dificuldade a respeito da plataforma, mas sim a dificuldade sobre o que ela vai ser utilizada, pergunta-se então: É fácil de operar e controlar?

2.3 - Eficiência

A **Eficiência** abrange os conceitos de tempo e recursos utilizados pela máquina após implementada e durante o desenvolvimento de um projeto. Se consome muita memória, se eleva muito o consumo de CPU, GPU etc. As perguntas que devem ser feitas aqui são: Qual é o tempo de resposta a velocidade de execução? E também: Quanto recurso usa? Durante quanto tempo?

2.4 - Portabilidade

Uma portabilidade eficaz prevê a **adaptabilidade** tanto em relação a outros programas da mesma estirpe, se os artefatos nele produzidos seriam reproduzidos de forma fidedigna em outros softwares. Também refere-se aos programas que de repente irão utilizar dos artefatos dele produzidos, como por exemplo quando for aberto por uma IDE, o artefato produzido será ‘carregado’ ou ‘lido’ de forma correta. A pergunta que deverá ser feita então é: É fácil de adaptar a outros ambientes, tanto da mesma plataforma, em caso de migração, quanto por softwares que irão utilizar tais artefatos?

Outra questão é sobre sua **capacidade para ser instalado**, se por exemplo, mesmo após a migração de um sistema operacional, funcionará em outros. Ou ainda, se existe dificuldade de instalação. É fácil de instalar em outros ambientes? Funciona em tanto em plataforma Windows quanto Linux e MacOS ou dispositivos móveis?

A **conformidade** diz respeito sobre a forma com que os artefatos produzidos pelo software de desenho são exportados, se teria ele um formato universal ou que pelo menos possa ser portado em diferentes tipos de extensão de arquivos compatíveis aos diferentes programas que o utilizarão. Deve se perguntar então: Está de acordo com padrões de portabilidade? Os formatos exportados são compatíveis com as ferramentas de desenvolvimento de software?

O último aspecto dentro do quesito portabilidade está na sua capacidade de ser substituído. Em caso de descontinuidade do software por exemplo, teria ele a capacidade de ser ‘aberto’ ou ‘lido’ por outros programas da mesma categoria: É fácil usar para substituir outro programa? A extensão de arquivo é compatível com outra plataforma?

2.5 - Confiabilidade e Manutenibilidade

O último aspecto a ser observado é sobre sua confiabilidade e manutenibilidade. Nesta etapa *bugs* de sistema serão observados. Testes em outros sistemas operacionais também devem ser feitos para que se observe em qual plataforma foi gerado algum erro, erros de instalação ou erros durante o uso do aplicativo. A manutenibilidade refere-se à capacidade de reinstalação, atualização e suporte técnico, os dois últimos oferecidos pela empresa.

É confiável? Existem *bugs* ou *glitches* que interferem em algum aspecto do software? A empresa cobra pelas atualizações e/ou manutenções, bem como no suporte se caso ocorra algum defeito? Estas são algumas perguntas que devem ser feitas no momento de avaliação do software escolhido.

3. A escolha das ferramentas

3.1 - Draw.io

Totalmente gratuito, o Draw.io é uma ferramenta competente e muito utilizada para a criação de diagramas UML (e de outros tipos). O site apresenta componentes que se adequam à criação de vários tipos de diagramas, como o de sequência, de atividades, de estados e o tradicional caso de uso, entretanto, pode não ter grande variedade de componentes e recursos.

O Draw.io possui uma versão para desktop que pode ser instalada diretamente na máquina, e apresenta as mesmas ferramentas da contraparte online. A vantagem é exatamente não precisar de conexão com a internet para funcionar, o que pode ser útil em alguns casos. Para usar este serviço não é necessário a criação de qualquer conta específica, basta ter conta no Gmail, OneDrive (Hotmail ou Outlook) ou Dropbox e permitir que a aplicação tenha acesso ao armazenamento. Se preferir, pode guardar directamente para o disco do seu dispositivo.

Se não quiser começar do zero, tem à sua disposição um conjunto de templates que se podem adequar ao que pretende. Podem ser ajustados e modificados da forma que preferir. Se nenhum dos templates oferecidos se enquadrar à sua necessidade, pode importar um via URL. Os conteúdos podem ser exportados em vários formatos: PDF, PNG, JPEG, GIF, SVG, HTML e XML. Pode obter o código de um div ou iframe para embeber numa página Web e apresentar publicamente.

Uma vez que o Draw.IO assenta inteiramente sobre a plataforma Google Drive, este recorre ao sistema de partilha do serviço cloud da Google, o que permite trabalhar em colaboração de forma grátis.

3.2 - Visual Paradigm

O Visual Paradigm é uma plataforma gratuita, mas que oferece planos pagos para os usuários que quiserem desfrutar de recursos avançados. O foco não é necessariamente os diagramas UML, mas há muitos outros formatos suportados; ele tem um procedimento de guia que incorpora instruções, referências de entrada e amostras para executar todos os passos com as ferramentas necessárias.

O Visual Paradigm fornece um ambiente colaborativo e automatizado para gerenciar as partes interessadas, tarefas e agendamento de tarefas com o gráfico PERT, gerando relatórios. Integra diferentes padrões, frameworks e processos de forma transparente, a variedade para o padrão UML é bastante grande. Além do diagrama de caso de uso, também tem o diagrama de classes, de sequências, de atividades, de componentes, de pacotes e de muitos outros. Além dos modelos, há templates que prometem facilitar a vida dos usuários.

3.2.1 - Algumas funcionalidades do Visual Paradigm

Ferramentas UML & SysML- Captura os requisitos funcionais com o diagrama de caso de uso UML. Cada caso em um diagrama de caso de uso representa um objetivo de negócios de alto nível que produz um resultado mensurável de valores empresariais. Os atores (UML) estão conectados com casos de uso para representar os papéis que interagem com as funções.

Ferramentas e diagrama BPMN- Visualiza os fluxos de trabalho empresariais. Comunica ideias de processo de negócios com os diagramas BPMN.

UML para código, código para UML- Gere código fonte do modelo de classe UML ou vice-versa.

Ferramentas de gerenciamento de projetos- Compreende as suas necessidades de gerenciamento de projetos com PERT, plano de implementação e análise de maturidade.

Formatos de saída versáteis- Exporta seu design em uma gama de formatos de arquivos.

Ferramenta de geração de documentos- Gere documentos completos e profissionais.

Guia de ciclo de vida do gerenciamento de projetos- Processo passo-a-passo para o gerenciamento de projetos, com instruções, exemplos e ferramentas para o desenvolvimento incremental de entregas.

Diversidades- Outras grandes ferramentas são o suporte a várias línguas, desenvolvimento de plug-ins e mais.

4. Aplicando critérios avaliativos sobre o conjunto de ferramentas escolhidas

Característica	Subcaracterística	Pergunta chave	Draw.io	Visual Paradigm
Funcionabilidade	Adequação	Possui todas as ferramentas internas para o desenvolvimento do projeto?	Não	Sim
	Acurácia	Irá atender o projeto como um todo?	Não	Sim
	Interoperabilidade	Os artefatos exportados irão funcionar em outros softwares de desenvolvimento?	Sim	Sim
	Conformidade	É o software que corresponde às expectativas dos desenvolvedores, bem como dos projetistas?	Sim	Sim
	Segurança de acesso	O software escolhido tem controle de acesso?	Sim	Sim
Eficiência	Tempo	A velocidade de execução gera atrasamento de rotinas de trabalho?	Não	Não
	Recursos	Usa muito recurso da máquina?	Não	Depende da máquina
		Usa muito recursos durante muito tempo?	Não	Depende da máquina
Usabilidade	Inteligibilidade	É fácil de entender a aplicação(Intuitivo)?	Sim	Não
	Apreensibilidade	É fácil de lembrar onde ficam as principais ferramentas?	Sim	Sim
		As teclas de atalho são de uso obrigatório ao software?	Não	Não
	Operacionalidade	É fácil de operar e controlar?	Sim	Sim
Portabilidade	Adaptabilidade	É fácil de adaptar a outros ambientes, tanto da mesma plataforma, em caso de migração, quanto por softwares que irão utilizar tais artefatos?	Sim	Não
	Capacidade para ser instalado	É fácil de instalar em outros ambientes?	Sim	Sim

		Funciona em tanto em plataforma Windows quanto Linux e MacOS ou dispositivos móveis?	Sim	Sim
	Conformidade	Está de acordo com padrões de portabilidade?	Sim	Sim
		Os formatos exportados são compatíveis com as ferramentas de desenvolvimento de software?	Não	Não
	Outros	É fácil usar para substituir outro programa?	Sim	Sim
		A extensão de arquivo é compatível com outra plataforma?	Sim	Sim
Confiabilidade e Manutenibilidade	Confiabilidade e Manutenibilidade do sistema	É confiável?	Sim	Sim
		Existem *bugs* ou *glitches* que interferem em algum aspecto do software?	Não	Não aparente
		A empresa cobra pelas atualizações e/ou manutenções, bem como no suporte se caso ocorra algum defeito?	Não	Não

5. Conclusão

Existem ferramentas que podem diminuir o risco na entrega de produtos de software, e por isso elas são tão importantes e devem ser utilizadas. Através dos resultados da fase de desenho aumenta-se a qualidade na passagem de informação para a fase de construção e, através da UML, podemos ter uma comunicação única, visual e bastante completa, da visão de como deve ser o produto em seu estado final.

Após analisarmos as duas ferramentas escolhidas, Draw.io e Visual Paradigm, podemos concluir que ambas são bastante completas e possuem muitos recursos. Porém, é possível ver também que o Visual Paradigm deixa um pouco a desejar se comparado ao Draw.io, nos seguintes aspectos: Adaptabilidade e Inteligibilidade. Dois pontos importantes quando se trata de uma ferramenta de desenho de software.

De forma geral, não existem muitas diferenças entre as duas ferramentas, e ambas entregam um ótimo serviço a quem as utiliza.

6. Bibliografia

Costa, Alexandre; Wernek, Vera; Campos, Francisco. **Avaliação de Ferramentas para Desenvolvimento Orientado a Objetos com UML.** Universidade do Estado do Rio de Janeiro, 2020. Disponível em: http://tinyurl.com/y4jsb4x6. Acesso em 13 de Outubro de 2020.

Ferramentas Online gratuitas para criar diagramas UML. **Profissionaisti.com.br, 2018.** Disponível em: http://tinyurl.com/y63upfqv. Acesso em 13 de Outubro de 2020.

Draw.io desenhar diagramas nunca foi tão fácil. **Peopleware, 2015**. Displonível em: http://tinyurl.com/yyrfuk47. Acesso em 13 de Outubro de 2020.

Visual Paradigm. **Software.com.br, 2020**. Disponível em: http://tinyurl.com/y2hors5y. Acesso em 13 de Outubro de 2020.

Critérios de Avaliação de ferramentas de Software de gerenciamento de processos. **PMKB.com.br, 2017.** Disponível em: http://tinyurl.com/y4ce46ck. Acesso em 13 de Outubro de 2020.