UNIVERSIDADE ESTADUAL DO CEARÁ

CENTRO DE CIÊNCIAS E TECNOLOGIA

PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

MESTRADO ACADÊMICO EM CIÊNCIA DA COMPUTAÇÃO

LUDMILA VARELA ARRUDA

UMA AVALIAÇÃO DE PROPOSTAS GRÁFICAS BASEADAS NA TEORIA DE VISUALIZAÇÃO DA INFORMAÇÃO PARA TRATAMENTO DE DADOS DE UMA FERRAMENTA DE OTIMIZAÇÃO COM A FINALIDADE DE PRIORIZAR CASOS DE TESTE

FORTALEZA – CEARÁ

2017

LUDMILA VARELA ARRUDA

UMA AVALIAÇÃO DE PROPOSTAS GRÁFICAS BASEADAS NA TEORIA DE VISUALIZAÇÃO DA INFORMAÇÃO PARA TRATAMENTO DE DADOS DE UMA FERRAMENTA DE OTIMIZAÇÃO COM A FINALIDADE DE PRIORIZAR CASOS DE TESTE

Dissertação apresentada ao Curso de Mestrado Acadêmico em Ciência da Computação do Programa de Pós-Graduação em Ciência da Computação do Centro de Ciências e Tecnologia da Universidade Estadual do Ceará, como requisito parcial à obtenção do título de mestre em Ciência da Computação. Área de Concentração: Ciência da Computação

Orientador: Prof. PhD. Jerffeson Teixeira de Souza

FORTALEZA – CEARÁ

2017

Dados Internacionais de Catalogação na Publicação

Universidade Estadual do Ceará

Sistema de Bibliotecas

Arruda, Ludmila Varela.
UMA PROPOSTA BASEADA NA TEORIA DE ?VISUALIZAÇÃO DA INFORMAÇÃO? PARA TRATAMENTO DE DADOS DE SAÍDA DE UMA FERRAMENTA DE OTIMIZAÇÃO PARA SELEÇÃO E PRIORIZAÇÃO DE CASOS DE TESTE [recurso eletrônico] / Ludmila Varela Arruda. - 2016.
1 CD-ROM: il.; 4 ¾ pol.

CD-ROM contendo o arquivo no formato PDF do trabalho acadêmico com 53 folhas, acondicionado em caixa de DVD Slim (19 x 14 cm x 7 mm).

Monografia (especialização) - Universidade Estadual do Ceará, Centro de Ciências e Tecnologia, Especializaçao em Engenharia de Software com Ênfase em Padrões de Software, Fortaleza, 2016.
Orientação: Prof. Ph.D. Jerffeson Teixeira de Souza.

1. Otimização Matemática. 2. SBSE. 3. Interação Humano-Computador. 4. Seleção e Priorização de Casos de Teste. I. Título.

LUDMILA VARELA ARRUDA

UMA AVALIAÇÃO DE PROPOSTAS GRÁFICAS BASEADAS NA TEORIA DE VISUALIZAÇÃO DA INFORMAÇÃO PARA TRATAMENTO DE DADOS DE UMA FERRAMENTA DE OTIMIZAÇÃO COM A FINALIDADE DE PRIORIZAR CASOS DE TESTE

Dissertação apresentada ao Curso de Mestrado Acadêmico em Ciência da Computação do Programa de Pós-Graduação em Ciência da Computação do Centro de Ciências e Tecnologia da Universidade Estadual do Ceará, como requisito parcial à obtenção do título de mestre em Ciência da Computação. Área de Concentração: Ciência da Computação

Aprovada em: 31 de agosto de 2017

BANCA EXAMINADORA

Prof. PhD. Jerffeson Teixeira de Souza (Orientador)
Universidade Estadual do Ceará – UECE

Prof. PhD. Paulo Henrique Mendes
Universidade Estadual do Ceará – UECE

Profa. Dra. Maria Elizabeth Sucupira Furtado
Universidade de Fortaleza - UNIFOR

Prof. MsC. Carlos Rosemberg Maia de Carvalho

Mestrado Acadêmico em Ciência da Computação UECE MACC

ATA DA CENTÉSIMA SEXTA DEFESA PÚBLICA DE DISSERTAÇÃO DE MESTRADO

Aos 31 e um dias do mês de agosto de dois mil e dezessete, no miniauditório do prédio de Pesquisa e Pós-Graduação em Computação, do Mestrado Acadêmico em Ciência da Computação – MACC, realizou-se a sessão pública de defesa da dissertação de **Ludmila Varela Arruda,** aluna regularmente matriculada no Mestrado Acadêmico em Ciência da Computação–MACC, Intitulada: **"Uma avaliação de propostas gráficas baseadas na teoria da "visualização da Informática" para tratamento de dados de uma ferramenta de otimização com a finalidade de priorizar caso de teste.".** A Banca Examinadora reuniu-se no horário de 11:00h às 12:30 horas, sendo constituída pelos Professores Doutores: Jerffeson Teixeira de Souza – (Orientador-UECE), Carlos Rosemberg Maia de Carvalho (UNIFOR), Paulo Henrique Mendes Maia (UECE) e Maria Elizabeth Sucupira Furtado (UNIFOR). Inicialmente o mestrando expôs seu trabalho e a seguir foi submetido à arguição pelos membros da Banca, dispondo cada membro de tempo para tal. Finalmente a Banca reuniu-se em separado e concluiu por considerar o mestrando APROVADA, por sua dissertação e sua defesa pública. Eu, PhD. Jerffeson Teixeira de Souza, Orientador e Presidente da Banca, lavrei a presente ata que será assinada por mim e demais membros da Banca. Fortaleza, 31 de agosto de 2017.

Prof. PhD. Jerffeson Teixeira de Souza
(Orientador/UECE)

Prof. MsC. Carlos Rosemberg Maia de Carvalho

Prof. PhD. Paulo Henrique Mendes Maia
(UECE)

Profa. PhD. Maria Elizabeth Sucupira Furtado
(UNIFOR)

Aos meus amados pais, Hilda Varela e Vladimir Arruda

AGRADECIMENTOS

Antes de tudo, agradeço a Deus, por me dar saúde e força para manter uma rotina de estudante e profissional e por me permitir conquistar grandes sonhos ao lado das pessoas que amo. Meu grande pastor, nada tem me faltado!

Agradeço aos meus pais, os melhores do mundo, por sempre incentivarem meus estudos, desde o ensino básico até hoje, e por acreditarem em mim. Aos meus avós Nilza e Paulo Varela, que não puderam vivenciar esse momento, mas a quem eu devo todo o meu agradecimento e admiração, assim como a todos os meus primos e tios dessa família.

Em especial, ao meu tio, prof. Dr. Antonio Themoteo Varela, que sempre me apoiou e quem me ajudou a dar os primeiros passos profissionais dentro do IFCE.

Ao meu namorado Euton Castro Jr, meu amor, por sempre me dar força, conselhos, carinhos e apertos, além de me apoiar incondicionalmente. E também por toda paciência nesse período de tensão pré-defesa. Obrigada, meu anjo.

Aos meus amigos, que quase esqueceram de mim, de tanto que fiquei ausente durante o mestrado, principalmente esses últimos dias.

À Samara Andrade, minha amiga fiel e irmã que torce, com todo amor, pelo meu sucesso e felicidade.

Ao Instituto Atlântico, pela liberação de tempo para o mestrado, e às pessoas da minha equipe, das quais sempre tive todo o apoio possível, em especial Patrícia Beserra, Paulo Abner e Henrique Luz.

Agradeço, de todo o meu coração, ao meu orientador, Jerffeson, por toda a paciência e dedicação que teve comigo, todos os conselhos e o grande empurrão no meu futuro. Em especial, por acreditar em mim, que eu poderia fazer um bom trabalho com o GOES.Uece, apesar do meu tempo dividido com o mercado. Muito mais do que esse trabalho, aprendi coisas que serão lembradas a vida inteira. Obrigada por tudo, inclusive pelos puxões de orelha, pois foram bem importantes!

Aos professores Carlos Rosemberg e Paulo Henrique, pelas maravilhosas observações e contribuições em relação ao meu trabalho.

Agradeço também aos amigos que fiz na Uece, principalmente os do GOES.Uece (Grupo de Otimização em Engenharia de Software), em especial ao Matheus, Altino, Alysson, Duany, Italo, Vanessa, Camila, Thiago e Raphael.

“Entrega teu caminho ao Senhor, confia nEle e o mais Ele fará”

(Salmos 37:5)

RESUMO

A área de Otimização Matemática aplicada à Engenharia de Software, também conhecida como Search-Based Software Engineering (SBSE), tem apresentado soluções satisfatórias para problemas complexos da Engenharia de Software (Harman et al, 2001). No entanto, essas soluções, por vezes, dependem de dados e funções matemáticas que precisam ser apresentadas aos usuários. Os mesmos, na maioria das vezes, não possuem conhecimentos específicos da área, dificultando assim o processo de interpretação e entendimento das soluções propostas. Partindo do problema de tratamento dos dados em SBSE e da necessidade de exibí-los de maneira mais facilitada, propostas gráficas foram criadas e avaliadas com o intuito de melhorar a visualização desses dados e de perceber o comportamento e opinião dos usuários durante a análise, a partir do escopo de testes de software.

O objetivo deste trabalho, portanto, é de realizar uma avaliação de propostas gráficas com base na teoria da visualização da informação para tratamento de dados de modo a priorizar casos de teste.

Palavras-chave: SBSE. Priorização de casos de teste. Visualização de Dados.

ABSTRACT

Mathematics Optimization area applied to Software Engineering, also known as Search-Based Software Engineering (SBSE), has good solutions to complex problems of Software Engineering (Harman et al, 2001). However, the solutions sometimes depends on the data and mathematical functions that users present. This users, for the most part, not has specific knowledge of the area, making it difficult to interpret and understand the proposed solutions. Based on the problem of data processing in SBSE and the need to display them in an easier way, graphic proposals were created and evaluated in order to improve the visualization of the data and the perception of users' behavior and opinion during an analysis, in the software testing scope.

The objective of this work, therefore, is to carry out an evaluation of graphical proposals based on the theory of information visualization for data processing in order to prioritize test cases.

Keywords: SBSE. Test Cases priorization. Data Visualization.

LISTA DE ILUSTRAÇÕES

Figura 1 – Integração das fases de seleção e priorização de casos de teste 15
Figura 2 – Soluções geradas ao final da segunda fase, de priorização de casos de teste. 16
Figura 3 – Entrada de arquivo de configuração e saída da metaheurística com Ids numéricos. 17
Figura 4 – Quatro etapas do Teste de Software. 21
Figura 5 – Publicações por ano em SBSE 25
Figura 6 – Taxa de publicação em SBSE por área 26
Figura 7 – Distribuição de itens marcantes na Visualização de Dados, ao longo do tempo. 29
Figura 8 – Dados representados através da cor, tamanho e forma. 31
Figura 9 – Processo automatizado de Visualização de Informações. 33
Figura 10 – Modelo de referência para construir a Visualização de Informações. . . 33
Figura 11 – Exemplo de gráfico de linhas para distribuição da população brasileira. 38
Figura 12 – Exemplo de gráfico de barras para temperaturas em várias localidades. 39
Figura 13 – Exemplo de gráfico de pizza para porcentagem da faixa etária de habitantes em determinada cidade. 40
Figura 14 – Exemplo de Scatter Plot (gráfico de dispersão). 40
Figura 15 – Exemplo de Circle Packing. 41
Figura 16 – Exemplo de Gráfico de Área para dados mensais de determinadas cidades. 42
Figura 17 – Exemplo de Gráfico de Clustered layout. 42
Figura 18 – Ilustração das taxas de desemprego e de inflação com um gráfico de barras. 44
Figura 19 – Ilustração das taxas de desemprego e de inflação com um gráfico de linhas. 45
Figura 20 – Ilustração das taxas de desemprego e de inflação com um diagrama de dispersão. 45
Figura 21 – O modelo de três fases da Interação Humano-Computador. 48
Figura 22 – Metacomunicação designer-usuário e usuário-sistema 49
Figura 23 – Modelo de atributos de acessibilidade do sistema de NIELSEN (1993). 50

Figura 24 – Gráfico para número de usuários versus problemas de usabilidade (NIELSEN, 2000). 52
Figura 25 – Ferramenta Raw Graph: inserção de dados (exemplo). 62
Figura 26 – Ferramenta Raw Graph: escolha da forma de visualização (exemplo). 62
Figura 27 – Proposta 1 – Circle Packing. 64
Figura 28 – Proposta 2 – Scatter plot. 64
Figura 29 – Proposta 3: Scatter plot de outra perspectiva. 65
Figura 30 – Proposta 4 – Clustered layout. 66
Figura 31 – Proposta 5 – Gráfico de barras. 67
Figura 32 – Proposta 6 – Gráfico de stream. 68
Figura 33 – Exemplo de soluções numéricas utilizadas. 71
Figura 34 – Registro de tempo utilizado por cada participante na análise das propostas. 73
Figura 35 – Percentual de completude das tarefas para cada uma das propostas. . 74
Figura 36 – Médias dos tempos usados para execução das tarefas, em cada proposta. 74
Figura 37 – Comparação entre o tempo e a compreensão dos participantes (indicado pela intensidade da cor - quanto mais escuro, mais difícil), nas seis propostas. 75
Figura 38 – Descrição dos participantes. 76
Figura 39 – Resultados das percepções dos participantes. 78

SUMÁRIO

1 INTRODUÇÃO 14
1.1 MOTIVAÇÃO 14
1.2 OBJETIVOS 17
1.2.1 Objetivo Geral 17
1.2.2 Objetivos Específicos 17
1.3 ORGANIZAÇÃO DO TRABALHO 18
2 FUNDAMENTAÇÃO TEÓRICA 19
2.1 TESTES DE SOFTWARE 19
2.2 SEARCH-BASED SOFTWARE ENGINEERING 24
2.3 SEARCH-BASED SOFTWARE TESTING 26
2.4 METAHEURÍSTICAS 27
2.5 VISUALIZAÇÃO DE DADOS 28
2.5.1 Breve Histórico 28
2.5.2 Dados Multivariados: definições e práticas 30
2.5.3 Visualização de dados através de suas representações e problemas encontrados 32
2.5.4 Vantagens de uma visualização 36
2.5.5 Tipos de Visualização de Dados e Gráficos Relacionados 38
2.5.6 Ferramentas de Visualização de Dados 43
2.6 VISUALIZAÇÃO DE SOLUÇÕES EM SBSE 43
2.7 IHC (INTERAÇÃO HUMANO-COMPUTADOR) 46
2.8 TESTE DE USABILIDADE 49
2.9 CONSIDERAÇÕES FINAIS 52
3 TRABALHOS RELACIONADOS 53
3.1 CONCEITOS GERAIS 53
3.2 PRINCIPAL REFERÊNCIA DA PESQUISA PRESENTE 54
3.3 SELEÇÃO E PRIORIZAÇÃO DE CASOS DE TESTE A PARTIR DE OUTRAS METAHEURÍSTICAS 54
3.4 FRENTE DE PARETO 55
3.5 VISUALIZAÇÃO DE DADOS 55
3.6 VISUALIZAÇÃO DE SOLUÇÕES EM SBSE 56

4 PROPOSTA 58
4.1 VISÃO GERAL 58
4.2 CONFIGURAÇÃO E SAÍDA DA METAHEURÍSTICA 59
4.3 MODELAGEM DA PROPOSTA 59
4.4 FERRAMENTA ESCOLHIDA PARA VISUALIZAÇÃO DE DADOS . . . 61
4.5 PROPOSTAS DE VISUALIZAÇÃO 63
5 AVALIAÇÃO DAS PROPOSTAS 69
5.1 INFORMAÇÕES DO TESTE DE USABILIDADE 69
5.2 VISÃO GERAL DA AVALIAÇÃO 70
5.3 INDICADORES UTILIZADOS 72
5.4 ANÁLISE DOS DADOS 73
5.5 DISCUSSÃO DA AVALIAÇÃO 77
6 CONCLUSÕES E TRABALHOS FUTUROS 80
6.1 CONTRIBUIÇÕES DO TRABALHO 80
6.2 LIMITAÇÕES 80
6.3 TRABALHOS FUTUROS 81

1 INTRODUÇÃO

A Engenharia de Software evoluiu significativamente nos últimos anos, a partir do aumento de processos baseados em computação, o que também trouxe à realidade o conceito de fábrica de teste, que é um aprimoramento do modelo de fábrica de software. Testar um software, basicamente, consiste na execução de um produto, visando identificar se o mesmo possui as suas especificações atendidas e se funciona corretamente, de acordo com o que foi preconcebido (Molinari, 2008).

O teste de produtos de software envolve basicamente quatro etapas: planejamento de testes, projeto de casos de teste, execução e avaliação dos resultados dos testes. Essas atividades devem ser desenvolvidas ao longo do próprio processo de desenvolvimento de software, e em geral, concretizam-se em três fases de teste: de unidade, de integração e de sistema (Pressman, 1997).

Mais especificamente, o caso de teste determina o fluxo de atividades que devem ser realizadas e validadas para cada requisito existente em um projeto. Muitas pesquisas são elaboradas, portanto, com o intuito de avaliar esses casos de teste, priorizá-los e distribuí-los, de maneira que seja possível garantir uma certa qualidade ao produto.

Unindo as dificuldades relacionadas à essas etapas do desenvolvimento de testes e as ideias propostas pela Engenharia de Software baseada em Busca (do inglês SBSE – Search Based Software Engineering), pode-se encontrar e verificar possíveis soluções para a resolução de problemas da esfera de Teste de Software. Search-Based Software Engineering (SBSE) é, portanto, o termo que trata da resolução de problemas provenientes da Engenharia de Software através da Otimização Matemática. Inclusive, uma linha de estudo mais específica, que será abordada, é a de SBST (Search-Based Software Testing), que trata especificamente dos problemas relacionados aos testes.

1.1 MOTIVAÇÃO

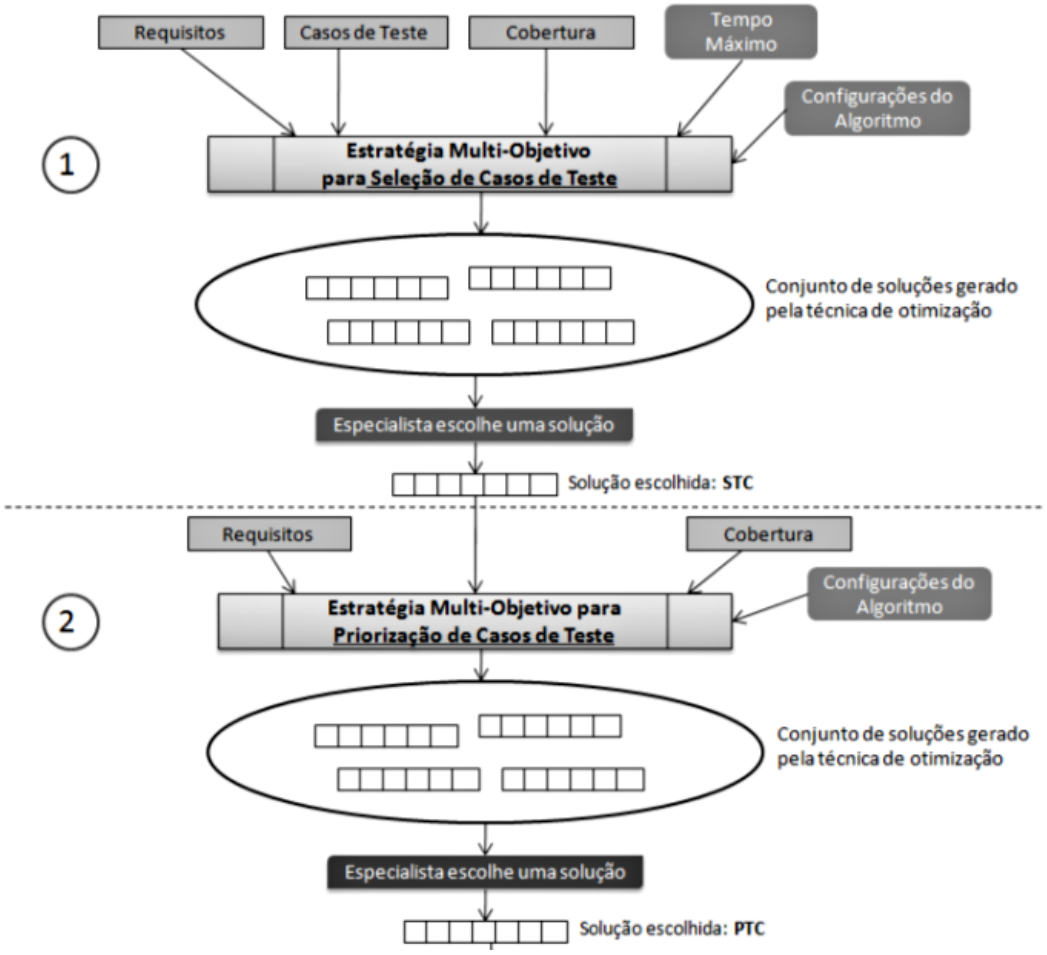
Apresenta-se como base dois trabalhos para o levantamento de propostas de solução, a nível de priorização de casos de teste. No trabalho de (MAIA, 2011), a autora propõe uma forma de selecionar, priorizar e alocar os casos de teste interativamente, de maneira multiobjetiva e integrada. Já no trabalho de (ARRUDA, 2015), essa abordagem é aplicada a dados reais de um sistema, apenas nas primeiras duas fases, de seleção e priorização, respectivamente.

A primeira fase, de seleção de casos de teste, vai gerar um conjunto de soluções

para que casos de teste sejam selecionados para a execução. Consequentemente, a segunda fase envolverá os mesmos casos selecionados na primeira fase, contudo, os mesmos estarão ordenados.

Para selecionar os casos, levou-se em consideração a relevância dos requisitos de acordo com a visão do cliente, a precedência dos casos e o tempo máximo concedido para a execução. As soluções geradas a partir daí são expostas a um especialista para que ele possa escolher a entrada da próxima fase, que ordena os casos, segundo à volatilidade e à complexidade dos requisitos. A aplicação dessas fases são cumpridas por meio de um algoritmo genético e foram verificadas em um sistema, já em produção, o que foi importante para a comprovação dos benefícios gerados através das técnicas de SBSE nesse cenário. A figura 1, a seguir, demonstra a integração das duas fases citadas.

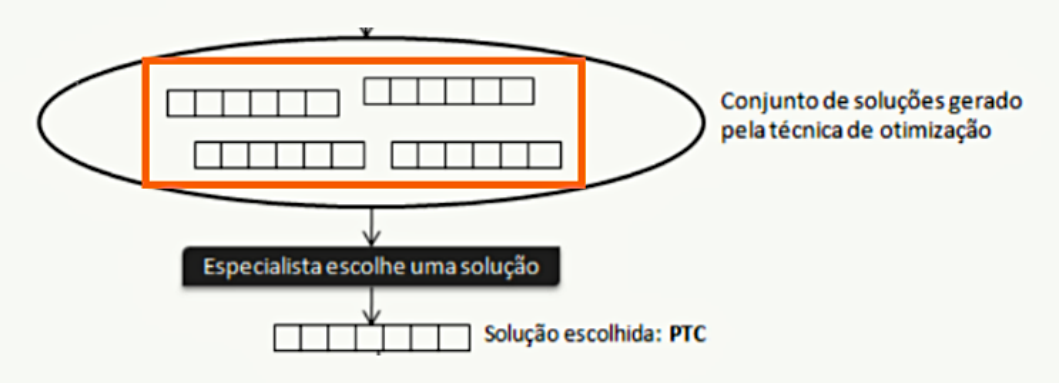
Figura 1 – Integração das fases de seleção e priorização de casos de teste



Fonte: (MAIA,2011)

A saída da segunda fase, portanto, é um conjunto de soluções de casos de teste priorizados. Porém, essas soluções são apresentadas, através da ferramenta de otimização, em um sequencial numérico, que denota os identificadores dos casos de teste. Essa saída é o objeto de estudo para o tratamento de dados proposto.

Figura 2 – Soluções geradas ao final da segunda fase, de priorização de casos de teste.

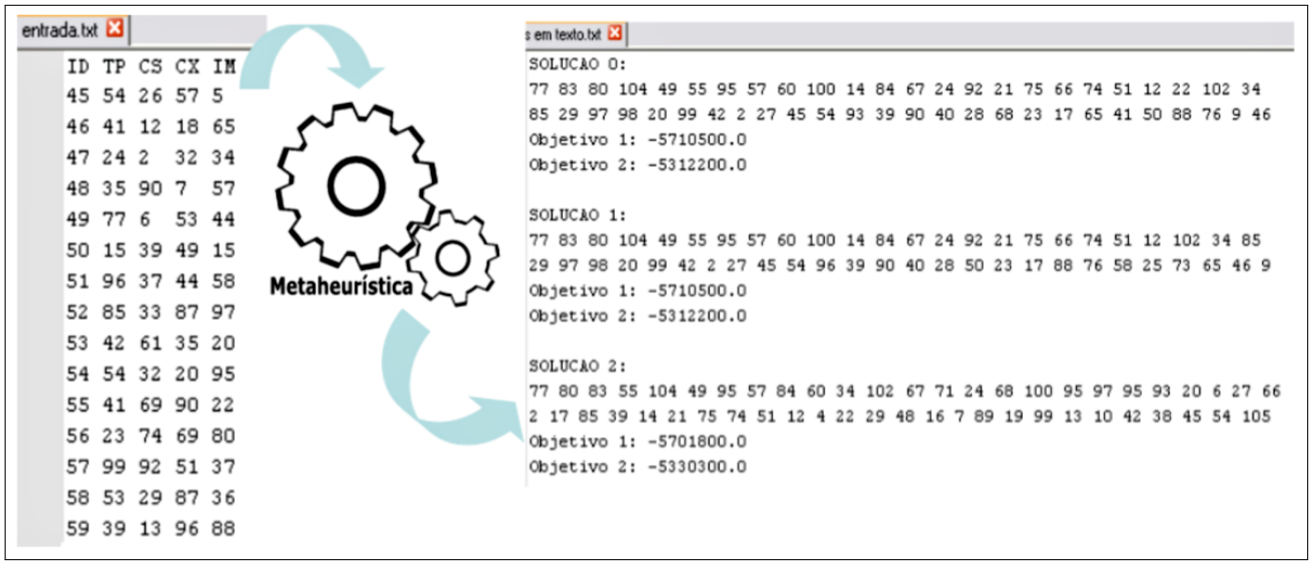


Fonte**:** (MAIA,2011)

Quanto às implementações em SBSE que fazem uso de alguma metaheurística, em geral, as mesmas exigem um ou mais arquivos de entrada de configuração, executam seus processamentos e expõem como saída, para o usuário, um arquivo texto com um conjunto de números, que representam conjuntos de boas soluções para o problema inicial. Quando se trabalha com problemas em que os objetivos se encontram em conflito, não é possível obter uma solução ótima, mas um conjunto de soluções que constituem a frente ótima de Pareto. Nestas circunstâncias, ter como solução do problema um conjunto de soluções "ótimas" pode ser entendido no sentido de não se poder afirmar que, nesse conjunto, uma solução é melhor do que outra. A comparação entre soluções não dominadas poderá ser operacionalizada à custa da estrutura de preferências do participante e a escolha de uma única solução ficará sempre vinculada às suas prioridades.

A questão é que, na maioria das situações, a saída na forma de sequência de números, conforme exposto na figura 3, não apresenta um significado sem algum tipo de tratamento ou decodificação, até mesmo para os profissionais que já tem algum conhecimento do assunto. Explorando essa forma de expor informações para o usuário e conhecendo o problema da grande massa de dados numéricos não significativos gerados por abordagens de SBSE, pensou-se em uma proposta que melhorasse a percepção e o entendimento do usuário a partir soluções geradas por uma ferramenta de SBSE para priorização de casos de teste, ou seja, foi percebida a necessidade de melhorar a visibilidade das soluções geradas.

Figura 3 – Entrada de arquivo de configuração e saída da metaheurística com Ids numéricos.



Para interpretá-las, o analista de teste precisaria buscar as informações (atributos do caso de teste) de cada identificador no arquivo inicial de entrada e saída, para entender o quão boa é aquela solução analisada. Isso acabaria exigindo uma dedicação de esforço e tempo maior para o analista de testes e, por ser uma atividade que envolve valores numéricos, estaria sujeita a erros humanos, seja por fadiga ou por falta de atenção.

1.2 OBJETIVOS

1.2.1 Objetivo Geral

Aprimorar a maneira como os dados gerados por uma ferramenta de SBSE são exibidos em um problema específico de priorização de casos de teste, através do estudo das teorias de Visualização da Informação e técnicas de Interação Humano-Computador, encontrando uma melhor visualização gráfica dentre as propostas apresentadas.

1.2.2 Objetivos Específicos

a) Propor formas gráficas de visualização da informação para uma massa de dados numéricos gerados por uma abordagem SBSE no problema priorização de casos de teste, especificamente;
b) Analisar e propor os parâmetros e atributos dos casos de teste que serão utilizados pela abordagem visual e que estejam cobertas pelos conceitos de IHC;

c) Gerar gráficos de análise para cada parâmetro/atributo observado e para as informações relevantes apontadas pelo avaliador, através da aplicação de um teste de usabilidade, elegendo uma melhor proposta visual.

1.3 ORGANIZAÇÃO DO TRABALHO

Este trabalho está organizado em 6 capítulos.

O Capítulo 2 apresenta a fundamentação teórica correspondente e necessária para a compreensão da abordagem proposta e do experimento aplicado.

O Capítulo 3 detalha as pesquisas anteriores que embasaram o trabalho presente e alguns trabalhos relacionados aos problemas de visualização de dados em sbse e na área de engenharia de software como um todo.

O Capítulo 4 detalha a proposta empregada, bem como os artefatos utilizados na criação dos gráficos e na análise dos experimentos.

O Capítulo 5 apresenta os resultados dos experimentos e exibe a avaliação de forma detalhada, enquanto que o Capítulo 6 cita as conclusões, limitações e discorre sobre possíveis trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo são apresentados os principais conceitos necessários para o completo entendimento da abordagem proposta e do teste aplicado nesta dissertação. São retratados conceitos sobre teste de software, sbse, metaheurísticas, interação humano-computador, usabilidade e, principalmente, visualização de dados.

2.1 TESTES DE SOFTWARE

O processo de desenvolvimento de software envolve uma série de atividades nas quais, apesar das técnicas, métodos e ferramentas empregados, erros no produto ainda podem ocorrer. Atividades agregadas sob o nome de Garantia de Qualidade de Software têm sido introduzidas ao longo de todo o processo de desenvolvimento, entre elas atividades de Verificação, Validação e Teste, com o objetivo de minimizar a ocorrência de erros e riscos associados. Os objetivos da verificação e validação mostram que o software atende à sua especificação e que atendem às necessidades do cliente, onde técnicas de revisão e inspeção também são aplicadas, podendo ser usadas em qualquer atividade de desenvolvimento ou semi-automatizada por análise estática (Sommerville, 2011).

Teste de software é um processo ou um conjunto de processos definidos com o objetivo de garantir que um código faça o que ele foi projetado para fazer (Myers, 2004).

É necessário analisar se o software é executado de forma controlada e se está fazendo o que deveria fazer, como escrito nos requisitos (Moreira, 2013). Todos os objetivos e técnicas de teste são traçados no início do projeto, bem como os recursos que serão utilizados, de acordo com a prioridade das funcionalidades. Normalmente, essas regras estão descritas em um documento chamado Plano de Testes. Essas atividades devem ser desenvolvidas ao longo do próprio processo de desenvolvimento de software e em geral, conforme mencionado anteriormente, concretizam-se em três fases de teste: de unidade, de integração e de sistema (Pressman, 2016).

Segundo Teunissen (2002), pode-se dizer que o objetivo dos testes é o de encontrar defeitos, através dos processos de planejamento, especificação, execução e análise de resultados, sendo considerados os riscos do negócio e a qualidade do produto.

Pode-se também resumir as definições dos autores acima a partir dos seguintes conceitos:

a) Testes são realizados para verificar se o software está fazendo o que foi solicitado pelo cliente através do requisito;

b) O processo de testes deve ser considerado um projeto, pois tem início, fim e todas suas etapas são definidas;
c) Testes são realizados para assegurar a qualidade do software e é importante um processo de testes bem definido;
d) Os testes são realizados para garantir que o produto não correrá riscos provocados por defeitos inseridos na produção.

A fase de planejamento de teste especifica todas as atividades de teste, recursos necessários e métricas utilizadas. A partir daí, os casos de teste, que detém esse fluxograma, devem ser elaborados, visando os requisitos funcionais e não-funcionais levantados pelo cliente. Os casos de teste são os cenários a serem executados para verificar se a implementação está de acordo com a especificação (Bastos et al., 2007). São compostos por pré-condições, pós-condições, o passo a passo de como executar o teste e pelas configurações necessárias para que o teste ocorra. Os casos de teste são originados da especificação de requisitos e regras de negócio, representando o que deve ser testado no sistema. Um conjunto desses casos é chamado de suíte de testes.

A execução desses casos avalia a qualidade do código e das funcionalidades que foram implementadas, como também a sua consistência. A cada nova tarefa executada dentro do projeto, executa-se os casos para garantir que nada do que foi construído anteriormente tenha sido afetado.

Os casos de teste utilizados durante a atividade de teste podem ser facilmente obtidos para revalidação do software após uma modificação. Com isso, é possível checar se a funcionalidade do software foi alterada, reduzir o custo para gerar os testes de regressão e comparar os resultados obtidos nos testes de regressão com os resultados do teste original. Testes de regressão garantem que o programa ainda satisfaz seus requisitos e não descartam a necessidade de testes para as capacidades novas e alteradas (Binder, 2000). Finalmente, com a avaliação dos resultados, um conjunto de correções e melhorias pode ser apontado para aprimorar a capacidade do software de atender às necessidades pré-estabelecidas.

Dessa maneira, a atividade de priorizar casos de teste torna-se cada vez mais importante dentro desse ciclo, visto que a complexidade dos sistemas tem crescido, ao passo que as variáveis de tempo e custo devem seguir as especificações do projeto.

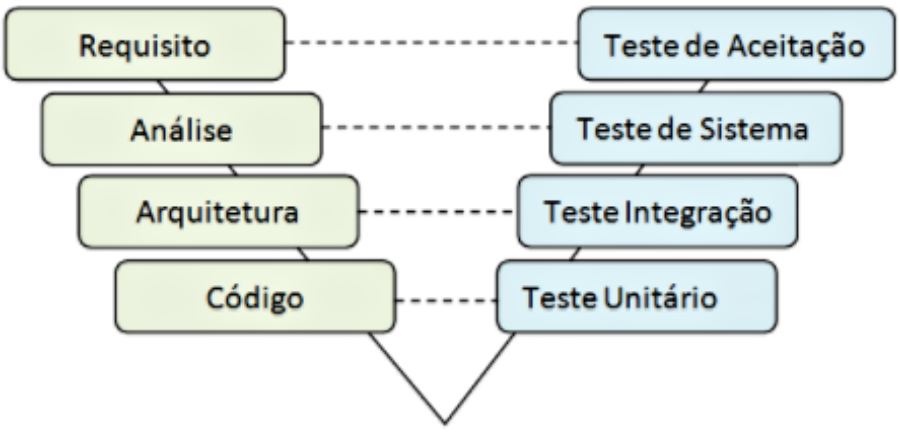
Apesar de possuir fases bem definidas, o ciclo de desenvolvimento de Teste nem sempre é estabelecido por completo dentro das organizações. A maioria das empresas realizam adaptações que satisfaçam suas principais necessidades e outras sequer aplicam este ciclo. A

ausência de profissionais especializados nessa área também contribui para que a qualidade na implementação dos softwares cresça de maneira mais lenta (Arruda, 2015).

Em Pressman (2001), o autor afirma que o Princípio de Pareto pode ser aplicado ao teste de software: a maioria dos problemas encontrados estão em uma pequena parte de seus componentes. Sendo assim, é importante saber quais casos de teste tem a maior probabilidade de detectar esses problemas. Caso seja necessário escolher ou ordenar esses casos de teste para a execução, isso deve ser levado em consideração, para que esses casos de teste pertencentes a menor parte, a qual detecta mais problemas, possam ser selecionados e/ou priorizados. Vários fatores podem influenciar nesta probabilidade de detectar problemas, dentre os quais pode-se destacar as características dos requisitos e características dos casos de teste (Maia, 2011).

O Teste de Software compõe-se, portanto, de uma série de quatro etapas, que são implementadas sequencialmente: o teste de unidade, o teste de integração, o teste de validação e o teste de sistema (Pierri, 2013).

Figura 4 – Quatro etapas do Teste de Software.



Fonte**:** (Pierri, 2013)

Inicialmente, os testes focam em cada componente individualmente, garantindo que ele funcione como uma unidade. O teste de unidade usa técnicas de teste com caminhos específicos na estrutura de controle de um componente para garantir a completude de cobertura completa e detectar o máximo de erros (Pressman, 2011). Em seguida, o componente deve ser montado ou integrado para formar o pacote completo de software.

Já o teste de integração cuida dos problemas associados aos aspectos de verificação e construção do sistema. Técnicas de projeto de casos de teste que focam em entradas e saídas

são mais predominantes durante a integração. Depois que o software foi integrado, é executada uma série de testes de ordem superior (Pressman, 2011).

O teste de validação, por sua vez, proporciona a garantia final de que o software satisfaz a todos os requisitos informativos, funcionais, comportamentais e de desempenho. O software uma vez validado, deve ser combinado com outros elementos do sistema, como por exemplo, o hardware e a base de dados.

O teste de sistema, por fim, verifica se todos os elementos combinam corretamente e se a função global do programa é alcançada (Pressman, 2011).

Cada vez que um novo módulo ou componente é integrado, o software é transfomado. Novos caminhos de fluxos de dados são estabelecidos, novas entradas e saídas podem surgir e uma nova lógica de controle pode ser acionada (Pressman, 2011). Essas alterações podem causar problemas com funções que antes funcionavam corretamente. O teste de regressão é a reexecução do mesmo subconjunto de testes que já foram executados para assegurar que as alterações não tenham propagado efeitos colaterais gerando comportamentos indesejáveis ou erros adicionais. Nesse contexto, é interessante observar que, na maioria das vezes, não é eficaz aplicar o teste de regressão para todas as funcionalidades do sistema. Logo, será necessŕaio selecionar e priorizar os casos de teste, reduzindo-os, de modo a se obter uma eficácia equivalente ao conjunto completo de casos de teste, por exemplo.

Um processo de Teste de Software tem como objetivo definir as etapas, os artefatos, as atividades, definir os papéis e responsabilidades, permitindo a empresa controlar e minimizar os riscos e agregar valor ao software. O ciclo de vida do processo consiste em uma sequência de etapas dependentes, considerado o fluxo do processo de teste, que visa definir as atividades e como os testes serão conduzidos.

A execução de cada etapa do ciclo de vida tem um tempo estimado dividido em planejar, projetar, executar e entregar. É a etapa inicial do processo que define o objetivo dos testes, evidenciando-o através do artefato Plano de Teste e é também a etapa responsável pela identificação e escolha das técnicas que serão aplicadas ao projeto. Durante o planejamento são executadas algumas atividades como definição do escopo de testes, definição dos métodos e técnicas de teste, planejamento das atividades de teste, planejamento do ambiente de testes, designação dos recursos envolvidos no teste, definição dos riscos do projeto de teste e definição das métricas para monitorar e controlar os testes. Alguns outros tipos de teste, segundo Molinari (2014), são:

a) Teste de Segurança: Testa a segurança da aplicação nas diversas formas;
b) Teste de Funcionalidade: É o teste mais importante, pois tem como objetivo mostrar que funciona ou não em tudo aquilo que a aplicação foi projetada para atender em termos de funcionalidades;
c) Teste de Estresse: Tem como objetivo avaliar como a aplicação irá reagir em condições dentro ou fora dos limites estabelecidos no projeto e
d) Teste de Performace: Tem como objetivo demonstrar como a aplicação se comporta em uma determinada carga de informações, medindo se o tempo atende as metas da aplicação.

Em cada etapa do ciclo de vida do processo de teste, os artefatos são produzidos com o propósito de registrar e acompanhar a evolução do projeto e verificar se os resultados obtidos estão de acordo com que foi solicitado. São eles (Molinari, 2014):

a) Plano de Teste: é o documento que define o nível de cobertura que os testes deverão alcançar e tem a finalidade de fornecer uma visão geral do projeto, definindo as diretrizes que possibilitarão a execução do Processo de Teste. É responsável pela definição do escopo dos testes, permitindo que os mesmos possam ser controlados e repetidos;
b) Casos de Teste: conjunto de entradas e saídas e suas condições, que tem como objetivo verificar e validar o projeto, a partir do resultado esperado;
c) Massa de Teste: conjunto de dados inseridos na base que permitirá a realização dos testes, preparando o ambiente para realização dos mesmos;
d) Roteiro de Teste: é formado pelo conjunto de um ou mais casos de teste, que informa a sequência em que os testes devem ser executados;
e) Relatório de Bugs: é o documento que registra e controla as ocorrências e defeitos no projeto de teste, que necessitam de correção ou investigação;
f) Resultado do Teste: é o documento que formaliza os resultados executados pelos testes, em cada iteração do sistema;
g) Sumário do Teste: artefato que apresenta uma análise resumida dos resultados do teste, informando as principais medidas que deverão ser adotadas para revisão e avaliação do projeto.

O processo de teste contribui, portanto, para melhorar a eficiência e a efetividade do teste em um projeto, agregando qualidade ao software.

2.2 SEARCH-BASED SOFTWARE ENGINEERING

A partir do contexto dos problemas encontrados durante o processo de desenvolvimento de um software, percebeu-se que estes poderiam ser modelados matematicamente, sendo resolvidos por meio de otimização matématica e utilizando metaheurísticas.

Search-Based Software Engineering (SBSE) é, portanto, o termo que trata da resolução de problemas provenientes da Engenharia de Software através da otimização matemática. Inclusive, tem-se outro termo, mais específico, o SBST (Search-Based Software Testing), que trata especificamente dos problemas da área de Testes, como neste trabalho (Freitas, 2010).

Além dos benefícios para resolução de impasses que a união das duas áreas oferece à Engenharia de Software em si, o estudo dessas áreas e a tomada de decisões referente a esses problemas tem tomado um rumo mais objetivo, já que as constantes matemáticas estão envolvidas nesse processo. As soluções assimiladas pelas técnicas de otimização apresentam uma melhor possibilidade de solução para o problema, trazendo vantagens ao tempo e custo envolvido no desenvolvimento.

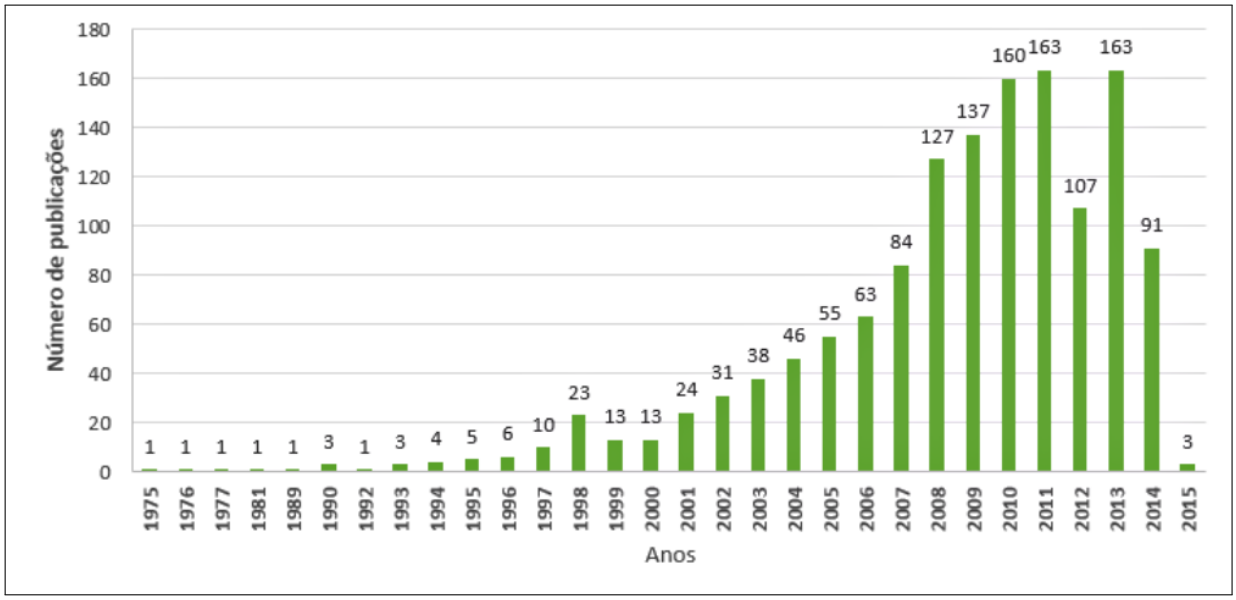
O primeiro trabalho em SBSE é datado de 1975, por W. Miller, D. L. Spooner, entitulado "Automatic generation of floating-point test data", onde dados de teste foram gerados a partir de uma maximização numérica. Entretanto, somente em 2001, a área recebeu esse termo, com o trabalho “Search-based software engineering” de Harman (2001). A partir daí, os demais pesquisadores incorporaram as técnicas de busca nos problemas de Engenharia de Software, utilizando o mesmo termo.

A área de Teste de Software é a de maior destaque em SBSE e os problemas de seleção e priorização de casos de teste e de geração de dados de teste recebem o maior destaque. O trabalho presente também enquadra-se, portanto, como uma evolução da monografia "Uma abordagem de Otimização para Seleção e Priorização de Casos de Teste a partir de dados reais"de ARRUDA (2015), onde foi realizada a coleta de dados reais de um projeto em uma empresa de sofware, para executar os experimentos.

A necessidade de formalização e automatização de processos na Engenharia de Software, aliadas à ampla consolidação do conhecimento na área de otimização têm feito da Engenharia de Software Baseada em Busca um promissor campo de pesquisa. A Figura 5 mostra um levantamento das publicações por ano em SBSE. Percebe-se que apesar do primeiro trabalho caracterizado dentro dessa subárea ser datado de 1975, conforme mencionado, a comunidade tornou-se bem mais ativa nas duas últimas décadas, chegando hoje a mais de 1300 produções

científicas publicadas em eventos específicos de SBSE ou ES e de computação em geral.

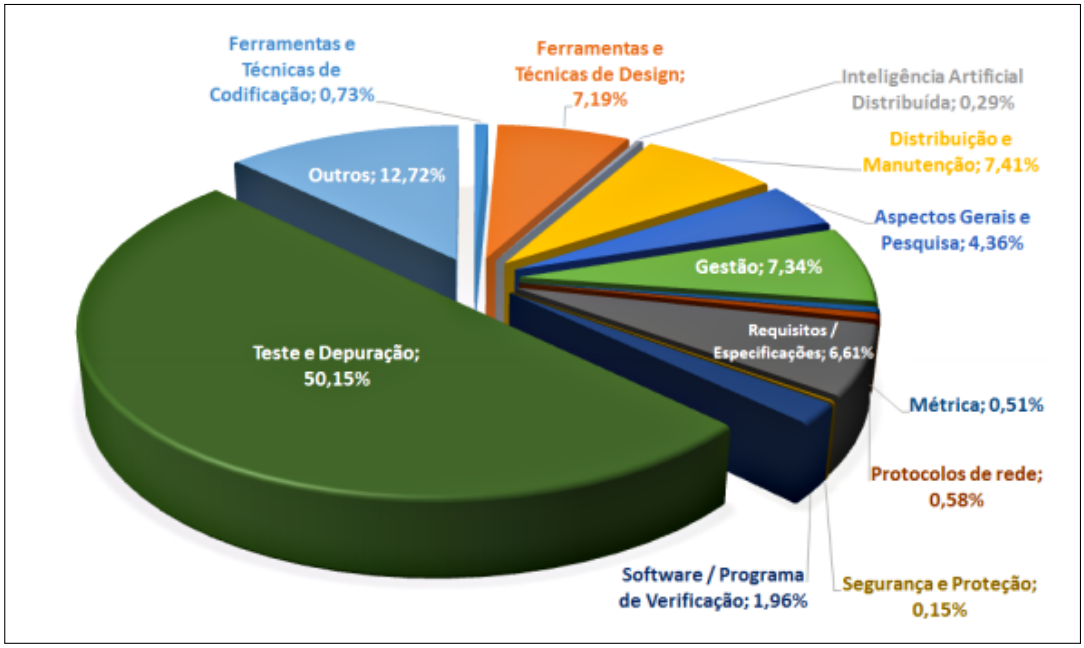
Figura 5 – Publicações por ano em SBSE



Fonte**:** adptado de (ZHANG; MANSOURI, 2015)

As aplicações de SBSE têm apresentado considerável variação, uma prova disso são trabalhos nos mais diversos campos de atuação da Engenharia de Software, por exemplo, no contexto de ferramentas e técnicas de codificação (KUKUNAS; CUPPER; KAPFHAMMER, 2010), (JIANG et al., 2007) e (HART; SHEPPERD, 2002); em ferramentas e técnicas de design (GUIZZO; COLANZI; VERGILIO, 2014), (SIMONS; SMITH; WHITE, 2014) e (RAMREZ; ROMERO; VENTURA, 2013); em testes e depuração (ASSUNçãO et al., 2014), (ALI; IQBAL, 2014) e (ABURAS; GROCE, 2014). A Figura 6 mostra a distribuição dos trabalhos de SBSE em 11 áreas de aplicação catalogadas por Zhang e Mansouri (2015).

Figura 6 – Taxa de publicação em SBSE por área



Fonte**:** adptado de (ZHANG; MANSOURI, 2015)

2.3 SEARCH-BASED SOFTWARE TESTING

Falando especificamente da área de Testes, define-se a utilização de uma técnica de busca, tal como um algoritmo genético, para automatizar parcialmente ou completamente uma tarefa de teste, por exemplo, a geração automática de dados de teste. A chave para o processo de otimização é uma função de aptidão específica do problema. O papel da função de adequação é orientar a busca de boas soluções a partir de um espaço de busca potencialmente infinito, dentro de um limite de tempo prático (Harman, 2001). Os primeiros trabalhos nesse sentido surgiram a partir de 1975, mas somente na década de 1990 começou a ganhar um ritmo mais relevante. Mais recentemente, portanto, tem-se verificado um crescimento acentuado na quantidade de trabalhos publicados, como foi visto na figura 6.

As metaheurísticas, portanto, são técnicas alternativas de otimização frequentemente empregadas no campo da SBST. Pode-se citar as técnicas de busca locais e globais: a) Hill-Climbing; b) Simulated Annealing; c) Greedy Randomized Adaptive Search Procedure (GRASP); d) Algoritmos Genéticos e e) In VitroFertilization Genetic Algorithm (Yamanaka, 2011).

Um problema de teste de software e/ou verificação e validação de domínio podem ser resolvidos através da utilização de uma estratégia de pesquisa, como pesquisa aleatória ou busca local, algoritmos evolutivos etc. Tem-se também outros estudos para testes baseados em

modelos, testes em tempo real, testes de interação, testes de arquiteturas orientadas a serviços e casos de testes para serem selecionados e priorizados (Ferreira, 2012).

O Teste de Software Baseado em Busca (SBST), portanto, trata da aplicação de técnicas de busca otimizadas para resolver problemas no teste de software. É usado para gerar dados de teste, priorizar casos de teste, minimizar conjuntos de testes, otimizar oráculos de teste de software, reduzir custos, verificar modelos de software, testar arquiteturas orientadas a serviços, construir conjuntos de testes para testes de interação e validar propriedades em tempo real (Kapfhammer, 2016).

Devido a grande proporção de trabalhos, foi possível unir diversas pesquisas e disseminar o conhecimento para diferentes ramos do desenvolvimento de testes de software, porém, apesar do grande número de publicações, ainda existem poucas pesquisas relacionadas ao problema de visualização de atributos e saída de dados, como proposto neste trabalho.

2.4 METAHEURÍSTICAS

Um algoritmo é considerado um método heurístico quando não há um conhecimento matemático completo sobre seu comportamento. Sua importância deve-se ao seu bom desempenho quando aplicados a problemas NP-difíceis, onde não são conhecidos métodos eficientes que apresentem garantias (Sucupira, 2010).

O estudo, ao longo do tempo, do desempenho dos métodos heurísticos conduziram os pesquisadores à elaboração de técnicas mais genéricas, que receberam o nome de metaheurísticas. Uma metaheurística é um conjunto de conceitos que pode ser utilizado para definir métodos heurísticos aplicáveis a um extenso conjunto de diferentes problemas. Em outras palavras, uma metaheurística pode ser vista como uma estrutura algorítmica geral que pode ser aplicada a diferentes problemas de otimização com relativamente poucas modificações que possam adaptá-la a um problema específico (Glover et al., 2002).

Apesar de não atingir o mesmo desempenho dos métodos heurísticos especializados, as metaheurísticas são importantes por descreverem métodos adaptáveis, apresentando ideias que podem ser aplicadas aos problemas de otimização para os quais não são conhecidos algoritmos e por marcarem o conhecimento sobre o processo de resolução de problemas complexos.

Uma desvantagem, entretanto, do uso de metaheurísticas está no ambiente mercadológico. A adaptação de uma metaheurística para a solução de novos problemas e o tratamento para visualização dos dados gerados, que surgem frequentemente na indústria de software, ainda

é custosa, dependente de fortes conhecimentos. Se a disponibilidade de recursos for pequena, os usários acabam por utilizar métodos mais clássicos e fracos.

2.5 VISUALIZAÇÃO DE DADOS

Visualização de Dados é uma área emergente da Ciência que pesquisa maneiras de apresentar informações visualmente de tal modo que as relações entre as mesmas sejam melhor compreendidas ou que novas informações possam ser descobertas. A presente seção discute vários aspectos relacionados com a apresentação visual de dados e introduz tipos efetivos e bem conhecidos para visualização de informações. As características estudadas têm aplicações práticas em vários campos da Ciência. Em Ciência da Computação, particularmente, elas ganham atenção especial na mineração de dados e na engenharia de software (Nascimento, 2015).

2.5.1 Breve Histórico

A representação gráfica de informações quantitativas tem raízes profundas. Essas raízes alcançam as histórias da primeira criação de mapas e representação visual e mais tarde se apresenta em cartografia temática e estatística, com aplicações e inovações em vários campos da medicina e ciência que estão interligados entre si (Hardle, 2014).

Alguns relatos históricos, mais importantes, de desenvolvimentos dentro dos campos de probabilidade (Hald, 1990), estatísticas (Pearson, 1978), astronomia (Riddell, 1980) e cartografia (Wallis, 1987) contribuíram de forma relevante para a visualização moderna de dados. Outros trabalhos mais especializados, que se concentram na história inicial de gravação gráfica (Geddes, 1962), gráficos estatísticos (Funkhouser, 1936), equações apropriadas para dados empíricos (Farebrother, 1999), economia e gráficos de séries temporais (Klein, 1997) e mapeamento temático (Robinson, 1982) apresentaram também uma excelente visão de alguns dos importantes desenvolvimentos científicos, intelectuais e técnicos dos séculos quinze a dezoito, que levaram à cartografia temática e ao pensamento estatístico. Wainer et al. (2001), por exemplo, já fornece uma contribuição mais recente de alguns dos antecedentes de gráficos estatísticos.

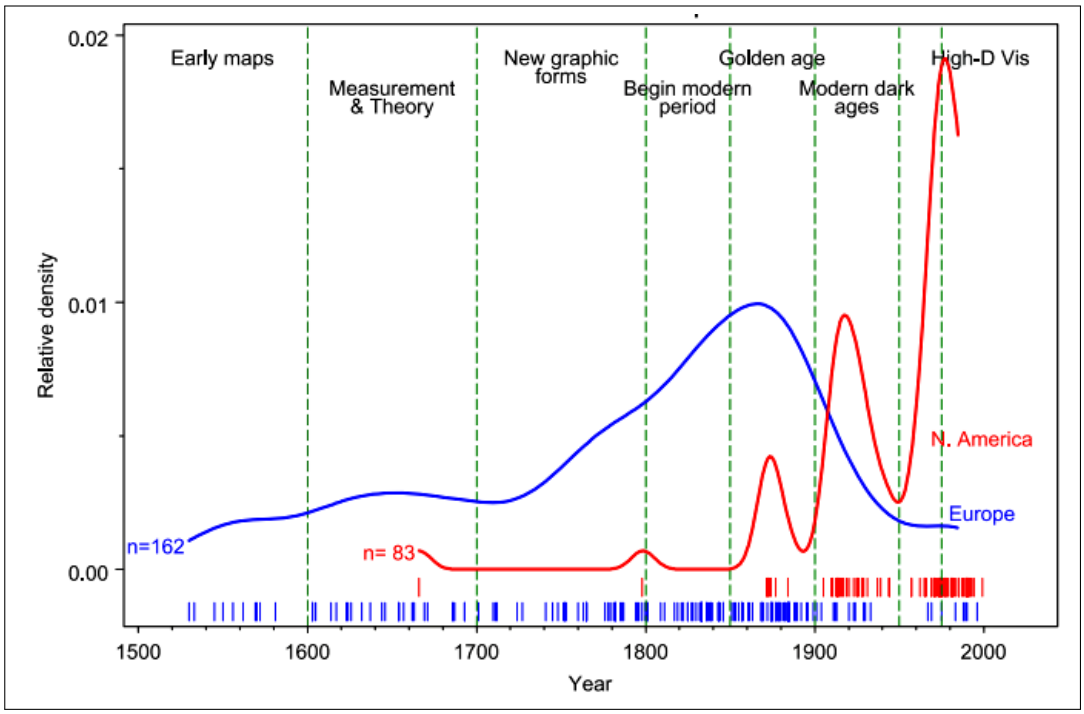
No último quarto do século vinte, a visualização de dados floresceu mais rapidamente, como uma área de pesquisa vibrante e multidisciplinar, fornecendo também ferramentas de software para uma ampla gama de métodos de visualização e tipos de dados, a partir dos computadores desktop. Surgiu, assim, a necessidade de aplicação de métodos de visualização a uma série cada vez maior de aspectos substantivos, problemas de estruturação de dados e uma

atenção substancialmente aumentada aos aspectos cognitivos e perceptivos dos dados de exibição (Hardle, 2014).

A partir de 1990, essas ferramentas foram reunidas para fornecer informações mais gerais com os sistemas de gráficos dinâmicos e interativos, combinando manipulação de dados e análise em ambientes informatizados coerentes e extensíveis. Tierney (1990), por exemplo, forneceu um ambiente orientado a objetos facilmente extensível para estatística informatizada. Nesses sistemas, controles deslizantes, caixas de seleção, listas de seleção, gráficos, tabelas e modelos estatísticos comunicaram-se com o usuário através de mensagens, inovando a comunicação e interação entre o usário e a máquina.

A maioria das idéias e métodos por trás dos gráficos interativos atuais são descritos e ilustrados em Young et al. (2006). A figura 7, por exemplo, ilustra a distribuição, no tempo, de itens marcantes no processo de visualização de dados, comparando América do Norte e Europa.

Figura 7 – Distribuição de itens marcantes na Visualização de Dados, ao longo do tempo.



Fonte: (YOUNG et al., 2006)

O desenvolvimento da área de Visualização de Informações foi motivado por alguns fatores, todos eles relacionados ao desenvolvimento tecnológico, pois isto disponibilizou inúmeras ferramentas para manipulação e desenvolvimento de todas as técnicas atualmente apresentadas.

Entre os elementos que tornaram a área de Visualização de Informações tão ampla recentemente, está a emergência de computadores pessoais poderosos e relativamente baratos, o aprimoramento de dispositivos eletrônicos dedicados à construção e visualização de gráficos de alta qualidade, por exemplo, placas de vídeo, monitores, impressoras, etc, a alta disponibilidade de memórias RAMs e de discos de baixo custo, além do desenvolvimento de técnicas e equipamentos para a interação com programas de computadores.

2.5.2 Dados Multivariados: definições e práticas

Os seres humanos possuem dificuldades em processar dados no formato de textos e tabelas; por isso, freqüentemente, recorre-se a meios visuais para interpretar as diversas informações no contexto da Engenharia de Software. O objetivo da visualização de informação é, portanto, possibilitar que dados sejam apresentados através de formas simples e intuitivas (Setzer, 1999).

A grande preocupação, entretanto, está na maneira como esses dados serão apresentados, pois uma boa escolha da forma de apresentação resulta na facilidade do entendimento e possibilita a descoberta de novas informações.

Atualmente, técnicas de visualização de informação são comumente encontradas, até mesmo em players de música para dispositivos móveis, onde cada música é representada por um ponto e este ponto é inserido em um plano. Os pontos mais a esquerda indicam que aquelas músicas possuem um ritmo mais rápido, pontos localizados a direita, músicas lentas e assim por diante. Com isso, para escolher as músicas que se deseja ouvir, marca-se o plano com o ponto, então o programa definirá um raio e tocará apenas as músicas que estiverem dentro da região especificada (Setzer, 1999).

Muitas maneiras podem ser utilizadas para apresentar dados, dentre elas, o formato, cores, movimentos, posicionamento na tela e tamanho (Fisher, 2013). Um exemplo dessa abordagem pode ser visto na figura 8. Através dela, é possível notar como a forma, o tamanho e a cor podem ser usados para diferenciar os dados.

Figura 8 – Dados representados através da cor, tamanho e forma.

Fonte**:** (Fisher, 2013)

É importante ressaltar que o formato visual da apresentação dos dados pode não somente facilitar, mas também dificultar a análise que se deseja fazer. Neste sentido, a escolha correta do paradigma visual e os atributos visuais que serão usados para representar os dados tornam-se uma escolha crucial à atividade de visualização. O paradigma visual utilizado deve se adaptar à natureza dos dados representados. Por exemplo, grafos são adequados para representar dados que descrevem relacionamentos (sites de relacionamento podem ser representados através de grafos). Árvores são adequadas para representar dados hierárquicos (por exemplo, estrutura de diretórios). Uma vez definido o paradigma visual, vários atributos visuais podem ser associados aos dados a serem representados. A adequação de um atributo visual ao dado depende da escala e tipo desse último atributo (Nascimento et al., 2014).

Os dados podem ser do tipo categórico ou numérico. Enquanto os dados categóricos não possuem um sentido de ordem ou unidade, os dados numéricos possuem. Considerando, por exemplo, a análise de sistemas de softwares; neste caso, o nome dos pacotes, classes e métodos são dados categóricos, enquanto o tamanho e a complexidade dos métodos são dados numéricos. Atributos numéricos podem ser mapeados naturalmente para atributos visuais como tons de cores, tamanho e posicionamento na tela. Atributos categóricos podem ser mapeados naturalmente para formas e conjuntos diferentes de cores. É importante ressaltar que o tipo e a natureza do dado pode também sugerir qual o mapeamento mais indicado para a sua representação visual.

Para uma boa escolha da forma de apresentação, é preciso conhecer a respeito do domínio no qual os dados serão utilizados, como esses dados serão utilizados, as informações que serão apresentadas e o objetivo da apresentação dos dados. Após escolher a forma de apresentação, o projetista deve levar em consideração alguns dos fatores como: o tipo dos dados, se estes são textuais e/ou numéricos; se existe a necessidade de interação entre o usuário e os dados; com que frequência os dados apresentados devem ser atualizados e se o usuário está interessado em informações precisas ou na relação entre o conjunto de dados (Nascimento et al., 2014).

Visualizações de informações têm sido desenvolvidas para uma grande variedade de outras aplicações práticas, tais como: monitoramento de bolsas de valores (Dwyer et al., 2002), consulta a bases de dados de filmes (Shneiderman, 1994) e desenho de organogramas de empresas e de árvores genealógicas (Battista et al., 1999). Particularmente na área da Computação, a visualização de informações tem aplicações especias na mineração de dados e na engenharia de software. Visualizações de software, por exemplo, auxiliam o programador a analisar e a entender a estrutura e o funcionamento de um programa, em um nível maior de abstração do que quando comparadas a uma simples leitura do código fonte.

Os tipos de visualização de informações também podem ser utilizados para filtrar, analisar e gerenciar grandes quantidades de dados, como as informações que são geradas e disponibilizadas diariamente na Internet.

2.5.3 Visualização de dados através de suas representações e problemas encontrados

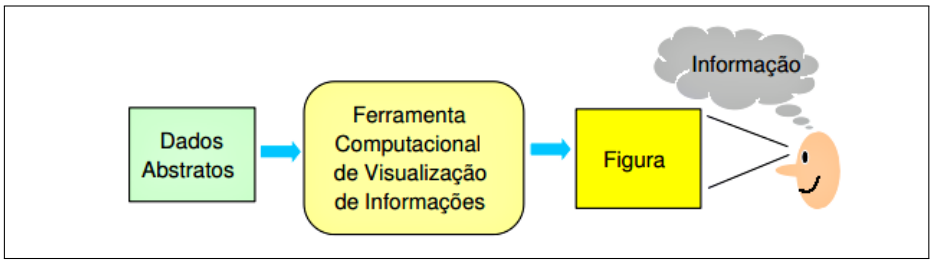
A visualização de software tem como objetivo facilitar o entendimento das informações relacionadas ao desenvolvimento de software (Cemin, 2001) e ainda atuar como uma forma de comunicação entre as pessoas que estão explorando ou manipulando a mesma informação (Shneiderman, 2000). Ela inclui técnicas de desenho, coloração, agrupamento de informações e animação. Com a visualização, é possível explorar o potencial da cognição humana e habilidades de percepção que não são possíveis apenas com representações textuais (Lintern et al., 2003). Representações visuais tendem a proporcionar uma forma mais simples e intuitiva de entender melhor e mais rapidamente o significado dos dados representados. Essa necessidade se torna ainda mais importante à medida que aumenta o volume de informação.

A constatação de padrões ou de características visuais presentes em imagens contribue de forma muito mais significativa para o processo de compreensão do que a simples

observação dos dados em sua forma bruta. A construção de uma visualização que possibilite essa observação pode ser conseguida organizando os dados segundo algum critério e apresentando-os de maneira visual. Em geral, visualizações acabam possibilitando a recuperação de informações relevantes e a construção de novos conhecimentos.

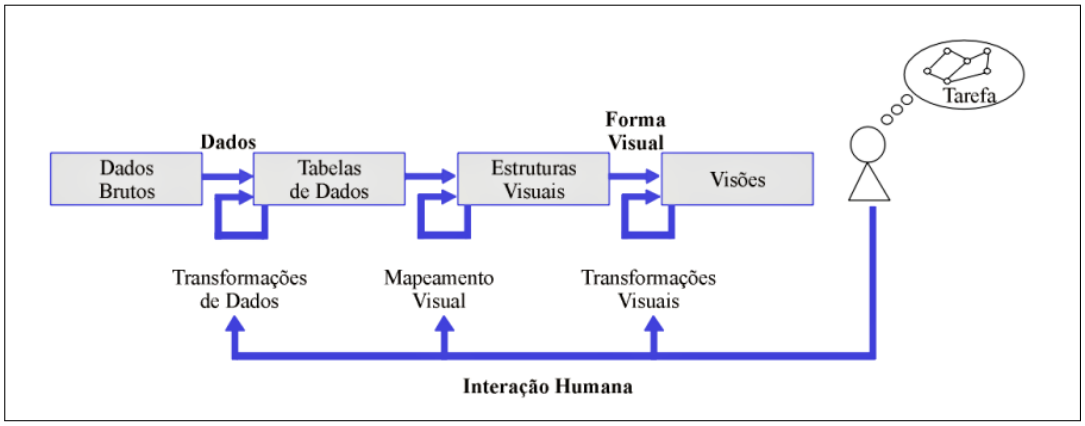
Em uma Visualização de Informações, os dados são abstratos, não havendo necessariamente uma representação geométrica inerente aos mesmos. Neste caso, uma imagem deve ser gerada com base nos relacionamentos ou informações que podem ser inferidos acerca dos dados. A Figura 9 ilustra, de modo simplificado, um processo automatizado de Visualização de Informações, onde uma imagem é produzida a partir de dados de entrada (Nascimento et al., 2014). Um modelo de referência para construir visualização de informações é apresentado por Card e outros (Card et al., 1999) e é mostrado na Figura 10. Esse modelo prevê a divisão do processo de gerar uma imagem para um conjunto de dados em três etapas descritas a seguir.

Figura 9 – Processo automatizado de Visualização de Informações.



Fonte: (Nascimento et al., 2014)

Figura 10 – Modelo de referência para construir a Visualização de Informações.



Fonte: (CARD et al., 1999)

A primeira etapa é chamada de Transformações dos Dados. Nessa etapa, um conjunto de dados brutos é processado e organizado em uma representação lógica mais estruturada, geralmente na forma de uma ou mais tabelas. O processamento pode envolver a eliminação de dados redundantes, errados ou incompletos, bem como a filtragem e o agrupamento dos dados relevantes. Além disso, pode ser feita a inclusão de novas informações, como, por exemplo, de resultados de análises estatísticas (média, soma total, desvio padrão, etc.) realizadas sobre os dados brutos. Uma forma comum de organizar os dados em tabelas é alocar uma linha para cada dado e uma coluna para cada atributo diferente dos dados. Dessa forma, a quantidade de linhas informa o número total de dados a serem visualizados, e o total de colunas representa a dimensão dos dados (Nascimento et al., 2014).

A próxima etapa é o Mapeamento Visual. A tarefa é construir uma estrutura visual que represente os dados da tabela. Toda estrutura visual pode ser decomposta em três partes: substrato espacial, marcas e propriedades gráficas das marcas. O substrato visual caracteriza o espaço para a visualização, sendo normalmente representado por eixos, tais como os eixos X e Y do plano cartesiano. Há quatro tipos elementares de eixos:

a) U = eixo não estruturado (ou sem eixo);
b) N = eixo nominal (região dividida em subregiões);
c) O = eixo ordenado (região dividida em subregiões, onde a ordem das mesmas tem importância);
d) Q = eixo quantitativo (a região tem uma métrica).

As marcas visuais são símbolos gráficos utilizados para representar os itens de dados. Dessa forma, o Mapeamento Visual consiste em associar os itens de dados a marcas visuais em um substrato visual. Cada atributo dos dados pode ser associado a propriedades gráficas das marcas (Nascimento et al., 2014).

A última etapa é a de Transformação Visual, na qual é possível modificar e estender as estruturas visuais interativamente através de operações básicas como:

a) Testes de localização, que possibilitam obter informações adicionais sobre um item da tabela de dados;
b) Controles de ponto de vista, os quais permitem ampliar, reduzir e deslocar a imagem com o objetivo de oferecer visões diferentes e
c) Distorções da imagem, visando criar ampliações de uma região específica em detrimento de outra.

É importante ressaltar que os mecanismos de interação implementados nessa etapa permitem ao usuário explorar diferentes cenários para um melhor entendimento dos dados visualizados. Além disso, o esforço de exploração dos dados é repassado em parte para o computador, uma vez que os cálculos e o redesenho da imagem são realizados pela máquina, deixando para o usuário a tarefa de observar o que acontece quando a visualização se modifica (Nascimento et al., 2014).

Como a área de Visualização de Informações envolve o ato de ver (Marr, 1982), a representação e o armazenamento adequado das mesmas e a extração de informações relevantes dentro de um contexto são tarefas complexas que abrangem a aquisição de imagens do ambiente, onde os estudos sobre o assunto não podem ser aprofundados sem o conhecimento de contribuições importantes advindas de diversos ramos da Ciência. Sendo assim, Visualização de Informações se encontra intimamente relacionada à Psicologia, à Linguística e às Artes Visuais, no que se refere à forma como o ser humano vê e interpreta o que está sendo visto. A Visualização de Informações também possui uma forte relação com algumas subáreas da Computação, como Visão Computacional e Interação Homem-Computador.

Por outro lado, uma visualização pode ser considerada expressiva ou não, se ela é capaz de mostrar todos os dados de interesse do usuário e nada mais que isso. Já a efetividade está relacionada com a facilidade de se compreender os dados apresentados. Para ser efetiva, uma visualização deve ser de rápida percepção e deve induzir à uma quantidade menor de erros de interpretação do que outras formas de se visualizar os mesmos dados. Efetividade e expressividade são aspectos importantes, porque sem essas características, uma visualização pode não ser capaz de enfatizar padrões relevantes nos dados, não trazendo, assim, quaisquer informações novas além daquilo que já é trivialmente conhecido. Além disso, uma visualização também pode ser de difícil entendimento ou, até mesmo, sugerir padrões que na verdade não existem, o que pode levar a uma interpretação errônea dos dados.

Logo, os problemas mais comuns que podem comprometer a efetividade de uma visualização são (Nascimento et al., 2014):

• Não colocar dados suficientes na visualização de forma a contextualizar as informações mais relevantes apresentadas;

• Desconsiderar atributos importantes dos dados;

• Utilizar gráficos sobrepostos em escalas diferentes ou com sistemas de coordenadas distintos, o que impede uma comparação justa entre os dados e

• Não fazer um mapeamento dos dados para marcas e atributos visuais de forma

adequada.

Características como cor, dimensionalidade, perspectiva, luminosidade, tamanho e forma dos objetos são fatores que auxiliam no processo de cognição e que podem ser explorados na construção de visualizações efetivas. Outros aspectos como existência de mecanismos de interação com os dados e a possibilidade de compactar uma grande quantidade de informações úteis em uma mesma imagem contribuem para a efetividade de uma visualização (Nascimento et al., 2014).

2.5.4 Vantagens de uma visualização

A quantidade de dados disponíveis e aos quais as pessoas estão expostas aumenta constantemente, sendo habitual chegar-se a milhares de elementos de dados, possuindo cada elemento diversos atributos. Isto ocorre em muitos domínios do saber e faz com que as aplicações de métodos e sistemas tradicionais para a visualização e análise de dados se tornem insuficientes, complexos e ineficientes (Healey, 2000).

Diante desta dificuldade, a área de visualização da informação fornece meios de analisar visualmente conjuntos de dados de elevada dimensão e complexidade, sendo uma importante contribuição para a descoberta de novos relacionamentos e dependências entre os dados. Isso acontece porque as visualizações garantem um grande apoio cognitivo através de vários mecanismos que exploraram as potencialidades da percepção humana, bem como a rapidez do processamento visual.

Panoramas, visão e perspectiva oferecem aos espectadores a liberdade de escolha que deriva de uma visão mais geral, uma capacidade de comparar e classificar através de detalhes. Uma micro-informação, como a textura menor na paisagem percepção, oferece um refúgio credível onde o ritmo de visualização é condensado, retardado e personalizado. Essas experiências visuais são universais, enraizadas na capacidade de processamento de informação dos humanos e na abundância e complexidade das percepções cotidianas. Tais projetos podem reportar imensos detalhes, organizando a complexidade através múltipla e, muitas vezes, camadas hierárquicas de leitura contextual (Tufte, 2001).

Algumas das vantagens da visualização citadas são (Ware, 2004):

- A visualização possibilita a capacidade de compreender grandes volumes de dados, ficando as informações importantes disponíveis imediatamente;

- A visualização permite a percepção de características que não são antecipadas

apenas com os dados originais. Frequentemente, a percepção de um padrão pode ser a base para uma nova visão sobre a temática;

- A visualização permite que problemas relativos aos dados tornem-se imediatamente aparentes. Com uma visualização apropriada, os erros ou anomalias presentes nos dados são rapidamente identificados. Por esta razão, visualizações podem ter um valor inestimável, por exemplo, em controle de qualidade;

- A visualização facilita a compreensão de características dos dados, seja em grande ou em pequena escala. Isto pode ser especialmente útil ao permitir a percepção de padrões;

- A visualização facilita a formulação de hipóteses e possibilidades.

A forma como as pessoas percebem e reagem ao resultado da visualização influencia fortemente no seu entendimento dos dados e na sua real utilidade. Por outro lado, os tipos de dados, como eles são estruturados e quais as suas finalidades contribuem significativamente no processo de visualização e devem desempenhar um papel importante no projeto, escolha e implementação da forma adequada de visualização.

Uma outra razão para se explorar o processo de visualização é que ele envolve o sentido humano, que possui maior capacidade de captação de informações por unidade de tempo. O sentido da visão é rápido e paralelo, sendo possível, inclusive, prestar atenção em um objeto de interesse especial, sem perder de vista (obviamente, com menos detalhes) o que está acontecendo ao redor (Ware, 2004).

Finalmente, as visualizações por si só trazem benefícios, uma vez que podem funcionar como uma extensão da memória humana e como um auxílio para o processo cognitivo. Por exemplo, anotações são feitas em uma agenda ou em um calendário e funcionam como lembretes de assuntos a serem discutidos ou de eventos que ocorreram ou que irão ocorrer. Também desenha-se diagramas e organiza-se informações espacialmente em uma folha de papel quando é abordado um problema que envolve diversas partes. As imagens ajudam a entender o problema e/ou a encontrar uma solução para o mesmo. Elas, inclusive, facilitam a memorização do objeto em estudo (Ware, 2004).

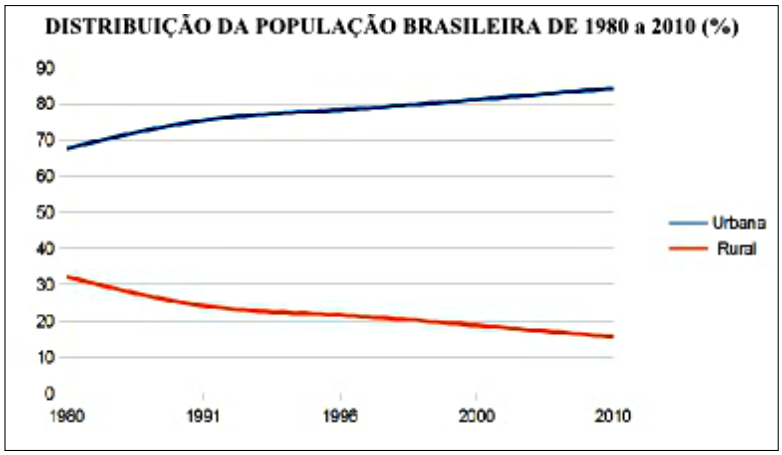
2.5.5 Tipos de Visualização de Dados e Gráficos Relacionados

Os tipos de Visualização de Informações utilizam representações ou metáforas visuais para exibir graficamente dados que geralmente não possuem representação direta, óbvia e natural (Luzzardi, 2003). Nas várias técnicas, frequentemente, os autores buscaram inspiração em objetos do mundo real (ou geométricos), para mapear o conjunto de informações. As técnicas podem utilizar representações visuais 1D, 2D ou 3D, não necessariamente de acordo com a dimensão do espaço de informação.

Os gráficos constituem uma forma clara e objetiva de apresentar dados estatísticos. De acordo com a característica da informação, é necessário escolher o gráfico correto (ALVES, 2014):

a) Gráfico de linhas - conforme se vê na figura 11, esse gráfico exibe uma série como um conjunto de pontos conectados por uma única linha. As linhas são usadas para representar grandes quantidades de dados que ocorrem em um período de tempo contínuo. Um gráfico de linhas requer pelo menos dois pontos para desenhar uma linha. Se o conjunto de dados tiver apenas um ponto de dados, o gráfico de linhas exibirá apenas um marcador de ponto de dados. Tem como objetivo ser esclarecedor e simples (Alves, 2014).

Figura 11 – Exemplo de gráfico de linhas para distribuição da população brasileira.

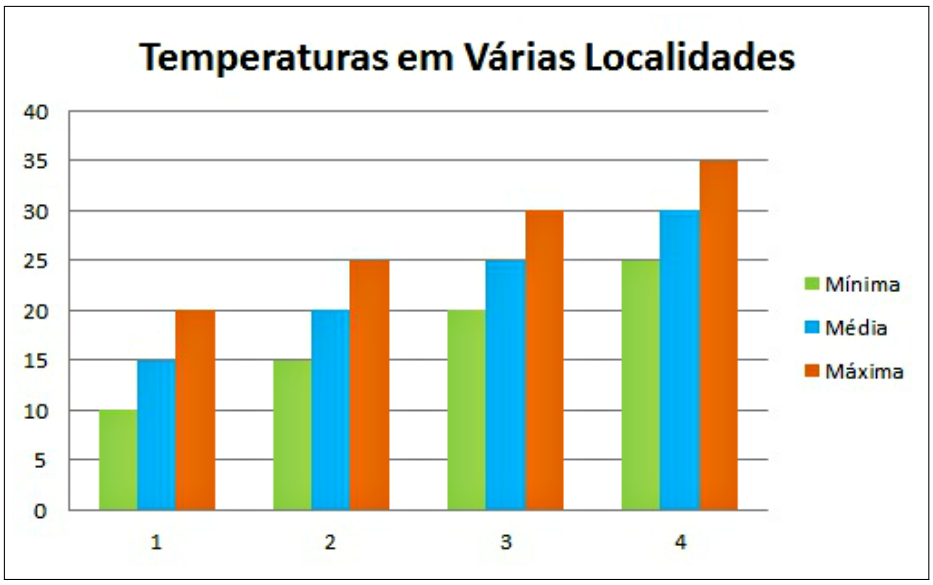


Fonte**:** (Alves, 2014)

b) Gráfico de barras - é uma forma de resumir um conjunto de dados categóricos. Ele mostra os dados utilizando um número de barras de mesma largura, cada uma delas representando uma categoria particular. A altura de cada barra é proporcional a uma agregação específica (por exemplo, a soma dos valores na

categoria que representa). As categorias podem ser algo como um grupo de idade ou uma localização geográfica. Se aplicado quando uma análise foi criada, o gráfico de barras pode exibir uma informação adicional em linhas de referência ou em diferentes tipos de curvas. Estas linhas ou curvas podem, por exemplo, exibir quão bem os seus dados se adaptam a certo ajuste de curva polinomial ou para resumir uma coleção de pontos de dados amostrais ajustando-os a um modelo que descreverá os dados e exibirá uma curva ou uma linha reta no topo da visualização. A curva normalmente modifica a aparência dependendo de quais valores foram filtrados na análise (Alves, 2014).

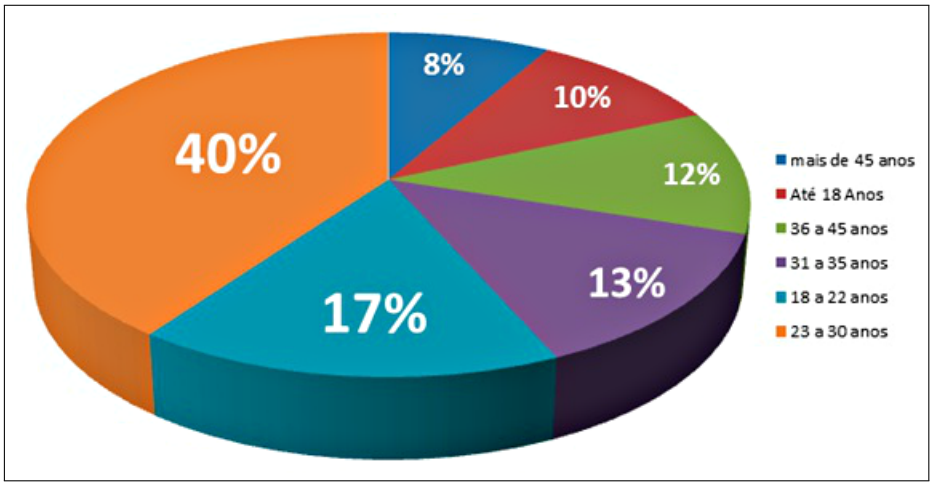
Figura 12 – Exemplo de gráfico de barras para temperaturas em várias localidades.



Fonte**:** (Alves, 2014)

c) Gráfico de pizza - gráfico dividido em setores, onde cada setor da pizza exibe o tamanho de uma parte da informação relacionada (Santos, 2016). Gráficos de pizza normalmente são utilizados para exibir os tamanhos relativos das partes de um todo, conforme exibido na figura 13.

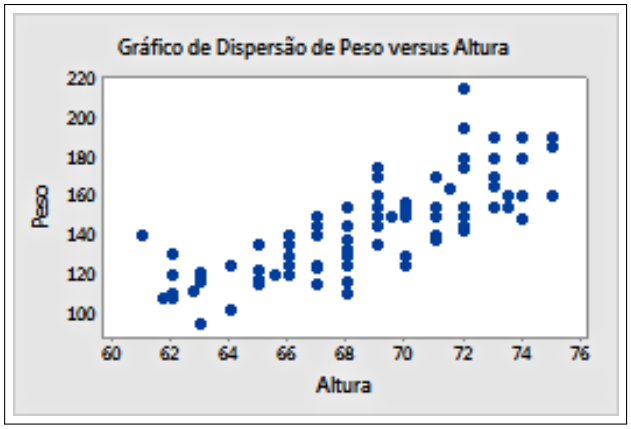
Figura 13 – Exemplo de gráfico de pizza para porcentagem da faixa etária de habitantes em determinada cidade.



Fonte**:** (Santos, 2016)

d) Gráfico de dispersão - quando é preciso visualizar a correlação entre mais de um atributo, ou melhor, identificar se o aumento ou diminuição de alguma variável influencia em outra, pode-se usá-lo, aplicado para correlações de duas ou mais variáveis. Um tipo conhecido desse gráfico é o Scatter Plot, utilizado para visualizar atributos par a par (Pereira, 2014).

Figura 14 – Exemplo de Scatter Plot (gráfico de dispersão).

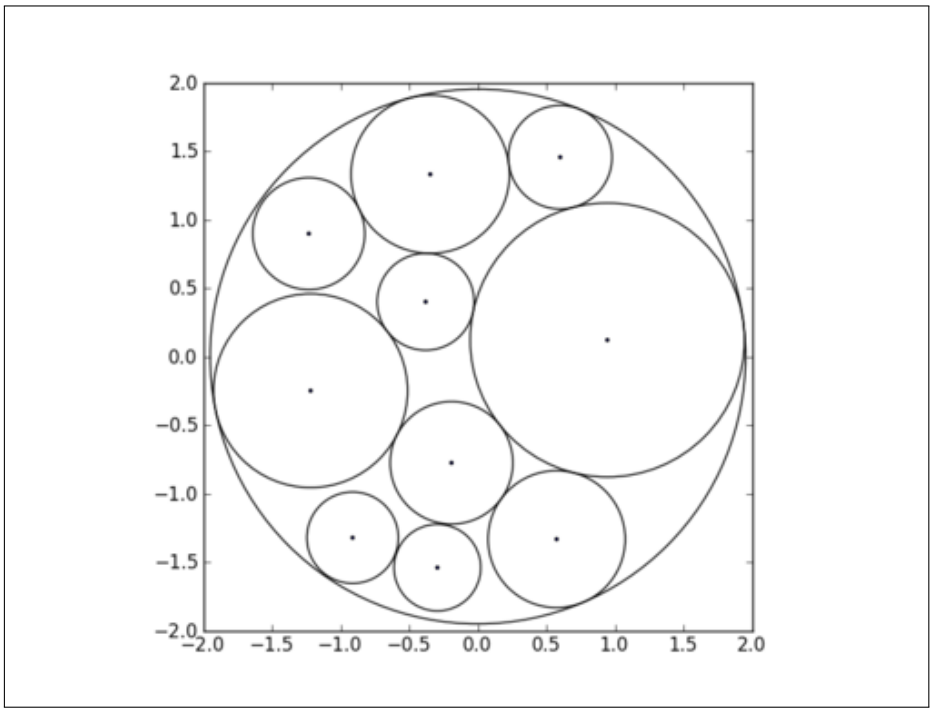


Fonte**:** (Pereira, 2014)

e) Gráfico de Circle Packing - em relação ao âmbito da geometria, também analisado em Visualização de Dados, a embalagem em círculo, tradução feita para os gráficos de circle packing, trata do estudo da disposição de círculos (de tamanhos iguais ou variados) em uma determinada superfície, de modo que não ocorram sobreposições e que todos os círculos se toquem. As generalizações podem ser

feitas para dimensões mais altas, o que é chamado de embalagem de esfera, que geralmente trata apenas de esferas idênticas. Esse tipo de gráfico é mais utilizado no ramo da matemática, o que diz respeito à geometria em si e à combinação de embalagens de círculos de tamanho arbitrário: estes dão origem a análogos discretos de mapeamento conformado, superfícies de Riemann e similares (Weissten, 2010).

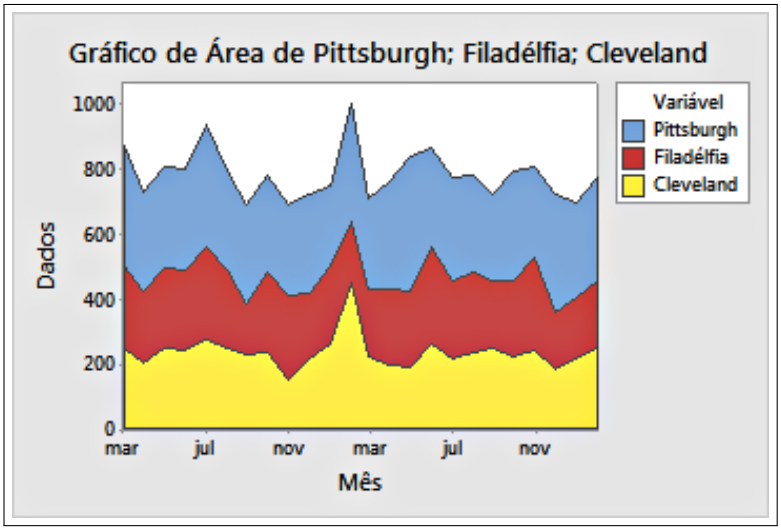
Figura 15 – Exemplo de Circle Packing.



Fonte**:** (Weissten, 2010)

f) Gráfico de área - é útil para enfatizar a magnitude das mudanças ao longo do período. Esses gráficos são também utilizados para mostrar a relação das partes com o todo e são como os gráficos de linhas, mas as áreas abaixo das linhas são preenchidas com cores ou padrões. Em 2008, o gráfico de ondas, que é um tipo de gráfico de área, se popularizou através da publicação de um artigo de Lee Byron, no jornal New York Times, sobre receitas de bilheteria no cinema. O autor afirma que esse tipo de proposta visual gera uma percepção extra sobre áreas, não apenas de um eixo específico. Entretanto, esse tipo de gráfico tem baixa utilização e popularidade nas escolas e faculdades, o que torna seu entendimento mais dificultoso e demorado pelos usuários, porém de extrema perceptividade, quando compreendido (Alves, 2014).

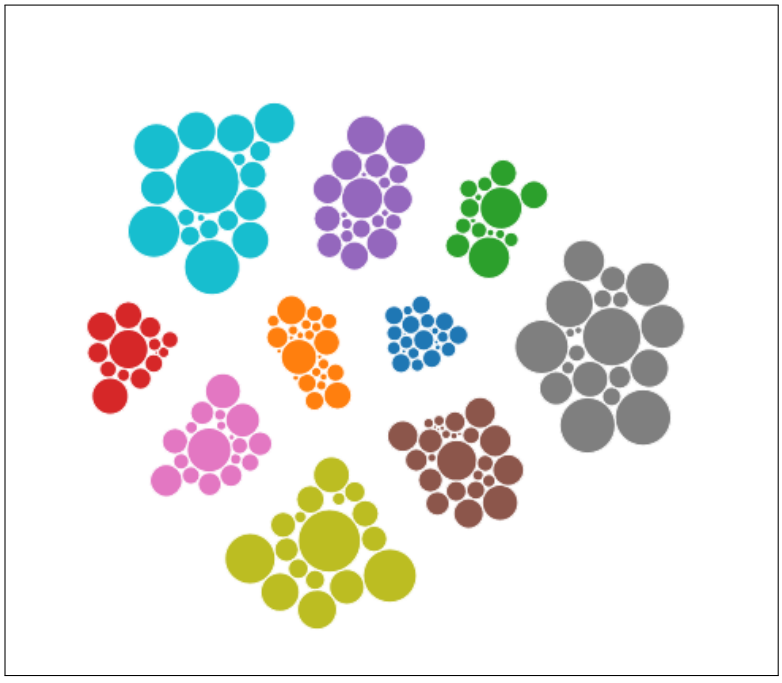
Figura 16 – Exemplo de Gráfico de Área para dados mensais de determinadas cidades.



Fonte**:** (Alves, 2014)

g) Gráfico de Clustered Layout - levando em consideração a mineração de dados, a análise de agrupamentos ou Análise de "Cluster", também denominada classificação não supervisionada, é a classificação de objetos em diferentes grupos, onde cada um deve conter os objetos semelhantes segundo alguma função de distância estatística, utilizada em otimização. Essa análise gera gráficos do tipo "clustered", com enfâse no agrupamento de informações (Jain, 2009).

Figura 17 – Exemplo de Gráfico de Clustered layout.



Fonte**:** (Jain, 2009)

2.5.6 Ferramentas de Visualização de Dados

Atualmente, para auxiliar na análise dos dados a serem visualizados, bem como na disposição dessas informações em gráficos apropriados, tem-se a disponibilidade de diversas ferramentas online, sem custo, que colaboram com o processo de visualização de dados. Esses recursos criam saídas de vetores de visualizações. É possível realizar a inserção de dados a partir de arquivos de texto ou de planilhas e selecionar opções diferentes de visualização, pré-estabelecidas. As ferramentas permitem também que os gráficos gerados sejam salvos em formatos distintos, podendo ser editados e/ou enviados para outros softwares.

2.6 VISUALIZAÇÃO DE SOLUÇÕES EM SBSE

Várias técnicas de visualização tem sido provadas na tomada de decisão múltipla objetiva, onde as fundamentais serão melhor descritas nos parágrafos seguintes. O tratamento desses dados se faz necessário nos problemas de múltiplos objetivos, onde o fator de decisão é convidado a avaliar uma série de alternativas (Levine et al., 2006).

Cada alternativa é caracterizada usando um vetor objetivo. A partir da perspectiva de visualização, a complexidade do problema de decisão depende de duas dimensões: o número de objetivos e o número de alternativas. A probabilidade pode ser complexa devido a um grande número de alternativas e um número pequeno de objetivos, ou o contrário, embora a natureza da complexidade seja diferente. Assim, diversas técnicas de visualização são necessárias para cada caso, além da avaliação do número de alternativas.

Na estatística descritiva, a computação gráfica é amplamente utilizada para ilustrar informações e produzir representações visuais (gráficos de barras, gráficos de linha, gráficos de pizza, etc). Técnicas de visualização mais avançadas, por exemplo, curvas de Andrews (1972) e faces de Chernoff (1973) também são apresentadas. Especialmente falando das curvas de Andrews e faces de Chernoff, estas foram desenvolvidas para ilustrar dados multivariados. Estas técnicas têm sido elaboradas para problemas em que o principal objetivo é a obtenção de um visão holística dos dados e/ou a identificação de clusters, valores atípicos, etc (Levine et al., 2006).

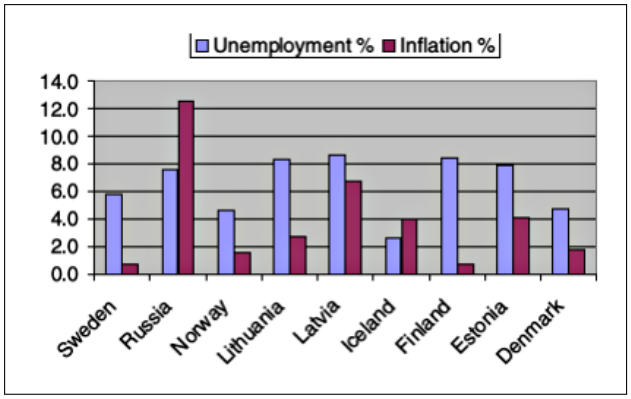
Técnicas gráficas também foram consideradas extremamente úteis em análise de dados. No entanto, elas não tem sido utilizadas pela comunidade para o seu pleno potencial, por exemplo, para abordagens interativas. Os estatísticos têm desenvolvido um certo número de métodos gráficos de análise de dados, tais como gráficos de barras, caminhos de valor, gráficos

de linha, etc. Todos estes tem uma característica comum: eles fornecem uma representação alternativa para dados numéricos e não há uma correspondência de um-para-um entre uma representação gráfica e dados numéricos (Levine et al., 2006). A representação gráfica pode ser transformada em forma numérica com uma certa precisão e vice-versa.

A exemplo disso, temos os gráficos de barras, que usam uma técnica padrão para resumir dados de frequência e que podem ser chamados de histogramas, neste contexto. Analisando a figura 18, um gráfico de barras é usado para representar os valores de desemprego e a taxa de inflação (em porcentagem) para cada um dos nove países. Isto é um exemplo usual de um quadro multiobjetivo, onde as variáveis correspondem aos objetivos. Em outras palavras, pode-se referir o objetivo espaço e não o espaço variável de otimização multiobjetiva.

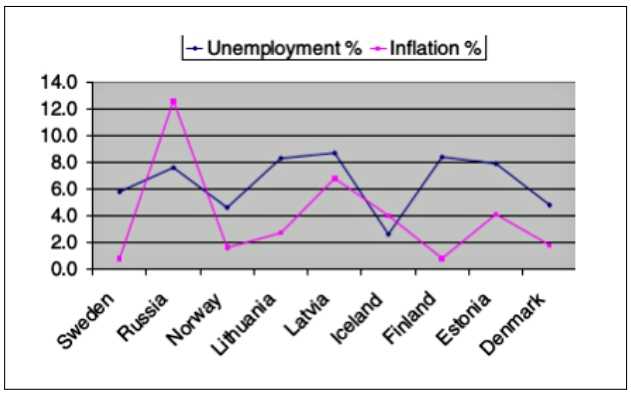
Existem também muitas outras técnicas de visualização utilizadas nas estatísticas, tais como gráficos de pizza, de linha e diagramas de caixa, que podem ser úteis em uma objeção múltipla específica (figuras 19 e 20). Os gráficos circulares são, por exemplo, úteis também para visualizar as probabilidades e percentagens (Bowerman et al., 2004).

Figura 18 – Ilustração das taxas de desemprego e de inflação com um gráfico de barras.



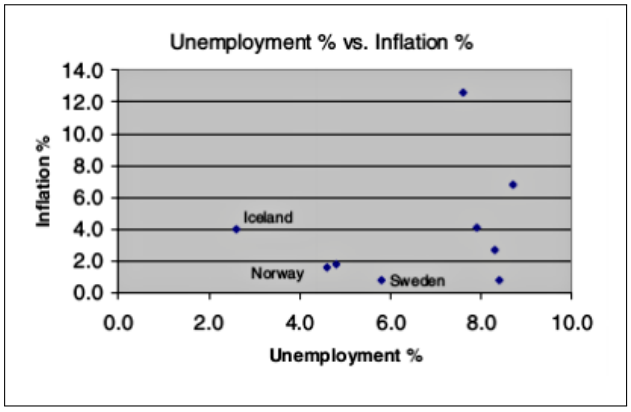
Fonte**:** (retirado de (BRANKE, J et al., 2008))

Figura 19 – Ilustração das taxas de desemprego e de inflação com um gráfico de linhas.



Fonte**:** (retirado de (BRANKE, J et al., 2008))

Figura 20 – Ilustração das taxas de desemprego e de inflação com um diagrama de dispersão.



Fonte**:** (retirado de (BRANKE, J et al., 2008))

É importante lembrar que a representação visual pode ser limitada. Portanto, a probabilidade é que se crie uma representação tridimensional para que seja possível a visualização multivariada de dados, isto é, quando o número de variáveis excede dois. Em estatística, dois princípios são aplicados nesse contexto:

1. Reduzir a dimensionalidade de um problema ou
2. Traçar uma observação multivariada como um objeto (um ícone).

Muitos autores também têm proposto o uso de técnicas gráficas (gráficos de barras, caminhos de valor, gráficos de linha, etc.) para ajudar a avaliar alternativas (Cohon (1978); Geoffrion et al. (1972); Grauer (1983); Grauer et al. (1984); Kok e Lootsma (1985); Korhonen e Laakso (1986); Korhonen e Wal-lenius (1988); Schilling et ai. (1983); Silverman et al. (1985); Steuer (1986)).

Em Miettinen (1999), há um capítulo dedicado à visualização de diferentes técnicas, onde a ideia central revela que uma representação visual melhora a legibilidade das informações. Em estatística, a redução do número de dimensões de um problema é uma técnica amplamente utilizada para comprimir a informação em dados multivariados. Se o número dos objetivos no contexto pode ser reduzido a dois (ou três) sem perder a informação essencial, logo, os dados podem ser examinadas visualmente. A análise de componentes principais e escalonamento multidimensional (MDS) também são interessantes.

Com relação à frente de Pareto, poucos estudos relacionam técnicas para sua abordagem e implementação. (Wallenius et al., 1988), trabalharam com o computador para encontrar os valores preferidos para os objetivos (variáveis de saída). Estes valores foram dispostos em um display em formato numérico e como gráficos de barras. A frente de Pareto representa um conjunto com as melhores soluções possíveis para o problema, considerando todos os objetivos ao mesmo tempo. Em outras palavras, este conjunto é formado por soluções que não são piores que nenhuma outra, considerando todos os objetivos (Souza, 2010).

Assim, podemos considerar o uso de gráficos nas tomadas de decisões mutiobjetivas. A questão principal é analisar o fator de decisão para então ver os vetores-objetivo (alternativas multidimensionais) e para facilitar a preferência nas comparações. As alternativas podem ser apresentadas uma de cada vez, duas de cada vez, ou várias de cada vez, para a consideração deste fator.

2.7 IHC (INTERAÇÃO HUMANO-COMPUTADOR)

A relação humano-computador foi proposta com base no conhecimento empírico relativo ao uso de sistemas interativos. À medida que a comunidade de IHC se consolidou e a pesquisa e aplicação de métodos avançaram, surgiram novas abordagens e teorias para dar conta dos processos de design e interação (Carroll, 2003).

A usabilidade, portanto, é um conceito que aborda a forma como o usuário se comunica com a máquina e como a tecnologia responde à interação do usuário, considerando as seguintes habilidades, de acordo com a norma ISO 9241 (Cybis et al., 2007):

- facilidade de aprendizado: a utilização do sistema requer pouco treinamento;

- facilidade de memorizar: o usuário deve lembrar como utilizar a interface depois de algum tempo;

- maximização da produtividade: a interface deve permitir que o usuário realize a

tarefa de forma rápida e eficiente;

- minimização da taxa de erros: caso aconteçam erros, a interface deve avisar o usuário e permitir a correção de modo facilitado;

- maximização da satisfação do usuário: a interface deve dar-lhe confiança e segurança.

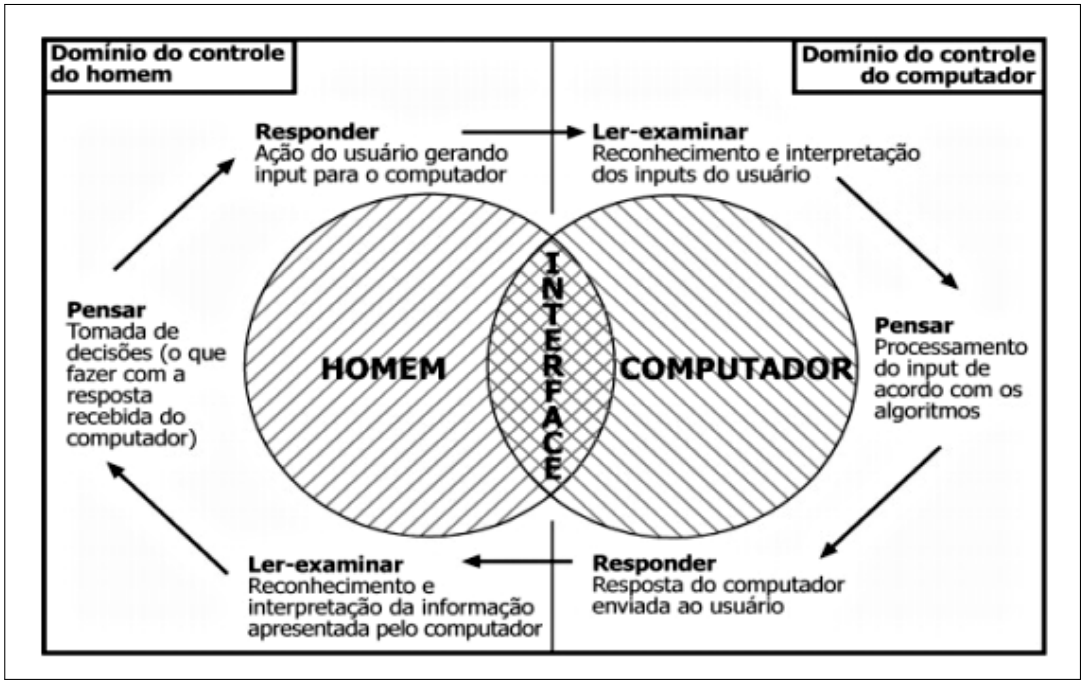
Mesmo que os indivíduos sejam únicos, as pessoas se comportam de forma previsível. Ao longo do tempo, os designers e cientistas da computação foram realizando estudos e observações do usuário, aprendendo como as pessoas operam suas atividades e como elas pensam sobre o que elas fazem. Assim, quando se observa as pessoas utilizando o software, ou fazendo qualquer outra tarefa, pode-se também esperar que elas respondam, de forma padronizada, aos estímulos envolvidos. A partir dessa ideia, percebe-se que uma interface que suporte esses padrões ajudará os usuários a atingir seus objetivos de maneira facilitada, comparando com interfaces que não levam essas ações em consideração. Os padrões não são apenas sobre a interface. Na maioria das vezes, é necessário uma análise de toda a interface do pacote, arquitetura subjacente, escolha de recursos, documentação etc (Carroll, 2003).

Seguindo este princípio, alguns padrões são definidos por O'Reilly (2006), para o estudo de design de interfaces, como por exemplo: exploração segura, gratificação instantânea, satisfação, opções diferidas, construção incremental, habituação, memória espacial e prospectiva, repetição simplificada, recomendações pessoais etc. Todos esses aspectos devem ser considerados no planejamento e na elaboração de propostas de visualização, de maneira que auxiliem os usuários finais.

Winograd (2003) sugere uma abordagem mais humana desse assunto, assumindo que a grande maioria da interação humano-computador é um passo de algum processo de interação humano-humano. O autor defende a ideia de que as possibilidades e mudanças tecnológicas derivam de alternativas, nas quais o computador pode exercer uma função de enriquecer as formas de comunicação humana, ou seja, sugere também algumas implicações para pesquisas futuras, em uma perspectiva de Interação Humano-Computador-Humano.

A figura 21 apresenta o modelo de Interação Humano-Computador de Mayhew (1992), de um sistema interativo baseado em computadores, composto pelo homem, pelo próprio computador e pelos limites do sistema.

Figura 21 – O modelo de três fases da Interação Humano-Computador.

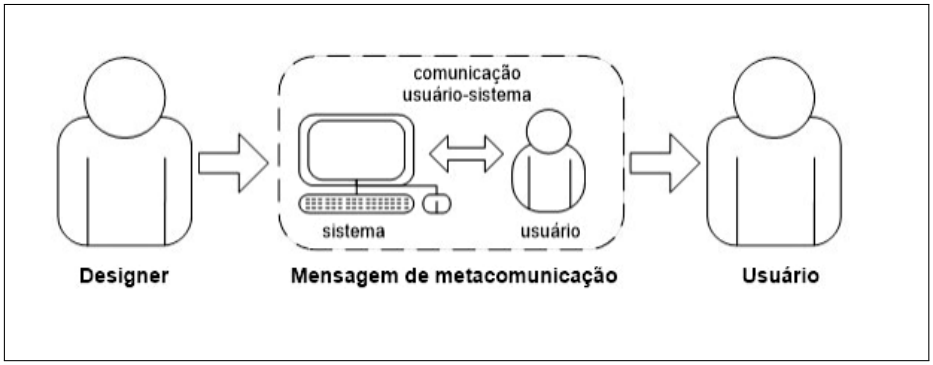


Fonte**:** (MAYHEW, 1992))

Para se entender, portanto, sobre design de interação, que é a área responsável pela criação de artefatos interativos, é necessário entender a ciência desse processo, estudada pelas teorias de IHC. Essas teorias são muito importantes porque delineiam as metodologias, métodos, técnicas e princípios empregados na prática. Nesse sentido abrange-se a engenharia cognitiva, que tem como objetivo aplicar a ciência cognitiva no design e na construção de artefatos computacionais. Esta foi proposta por Don Norman em 1986 e pretende compreender e demonstrar como os processos humano-computador ocorrem, de maneira a prever e evitar possíveis problemas de interação. Logo, os sistemas interativos são artefatos de metacomunicação, conforme exposto na figura 22, em outras palavras, um artefato de comunicação sobre comunicação, visto que o designer cria signos para se comunicar com os usuários por meio das interfaces, ou seja, a comunicação entre designer e usuário é mediada pela interface (Barbosa, 2010).

A partir desse conceito também estuda-se as ideias de distância semântica (distância entre a formulação da mente e a projeção sobre as funcionalidades do software) e distância articulatória (distância entre as funções disponíveis na interface e as ações que o usuário deve realizar).

Figura 22 – Metacomunicação designer-usuário e usuário-sistema .



Fonte**:** (Barbosa, 2010))

2.8 TESTE DE USABILIDADE

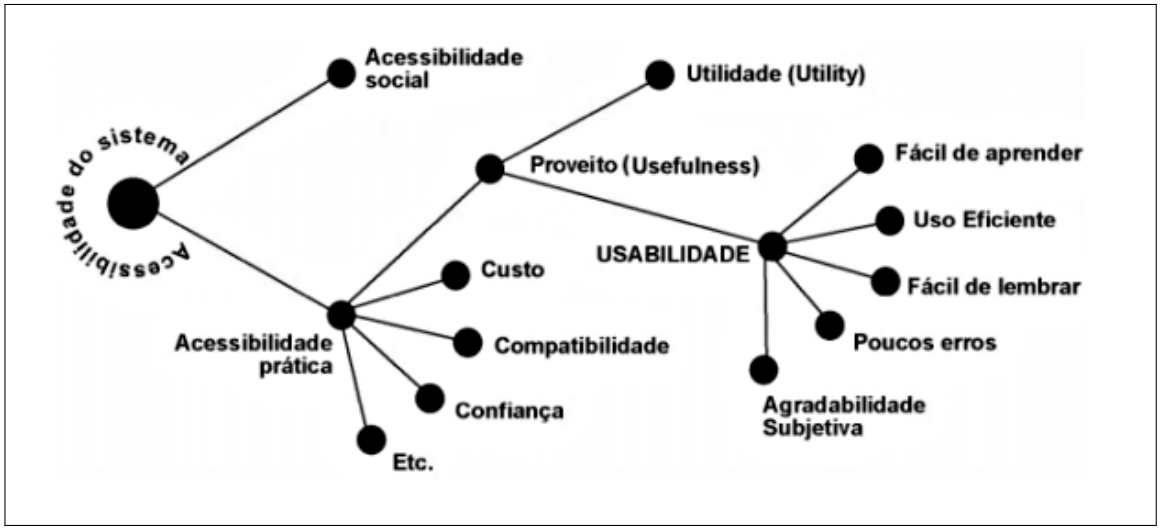
Nielsen (1993) afirma que, com o objetivo de definir o conceito abstrato de "usabilidade" em termos de seus componentes mais precisos e mensuráveis, pode-se chegar a uma disciplina de engenharia, onde a usabilidade não é simplesmente discutida, mas é sistematicamente aproximada, aperfeiçoada e avaliada (possivelmente mensurada).

Para se medir a usabilidade, normalmente, aplica-se o teste com usuários (estes usuários escolhidos devem representar o cenário real da maneira mais próxima possível) que usem o sistema ou aplicação abordada.

De acordo com Nielsen (1993) e Shneiderman (1998), para se chegar ao uso de um sistema de IHC de maneira facilitada, é importante considerar as diferenças individuais e as categorias de usuário. É relevante considerar cada usuário não apenas no seu nível (experiente ou iniciante), mas na intercessão de três dimensões ao longo do qual a experiência dos usuários difere: com o sistema, com os computadores e com o domínio da tarefa.

Logo, um teste de usabilidade deve avaliar se o sistema oferece sua funcionalidade de tal maneira que o usuário, para o qual foi planejado, seja capaz de controlá-lo e utilizá-lo sem constrangimentos sobre suas capacidades e habilidades (MORAES et al., 2000). A figura 23 demonstra o modelo de atributos de acessibilidade do sistema, descrito anteriormente.

Figura 23 – Modelo de atributos de acessibilidade do sistema de NIELSEN (1993).



JOKELA et al. (2003) fez a comparação entre a ISO 9241-11 (que define usabilidade) e a ISO 13407 (que fornece guias para o design da usabilidade), apontando as definições dos termos nelas contidos, como:

• Efetividade: a precisão e integridade (no sentido de completude) com as quais os usuários alcançam objetivos especificados;

• Eficiência: os recursos gastos em relação à precisão e integridade com os quais os usuários alcançam objetivos;

• Satisfação: livre de desconforto e atitude positiva do uso do produto;

• Contexto de uso: características dos usuários, das tarefas e do ambiente físico e organizacional;

• Objetivo: resultado pretendido;

• Tarefa: atividade requerida para se atingir um objetivo.

E ainda, a partir da visão de Nielsen (2003), de que usabilidade é um atributo de qualidade, usado para estimar o quão fácil de usar é uma interface, o autor aponta seus cinco componentes:

• Fácil aprendizado: tarefas básicas devem ser realizadas facilmente logo na primeira vez que os usuários se deparam com o design do sistema;

• Eficiência: uma vez tendo aprendido o design, os usuários devem ser capazes de realizar as tarefas rapidamente;

• Fácil memorização: deve ser fácil de relembrar, de forma que usuários esporádicos ou aqueles que ficaram um período sem usá-lo sejam capazes de retomá-lo sem ter que reaprendê-

lo;

• Poucos erros: o usuário deve poder cometer pouquíssimos erros durante o uso do sistema e, caso ocorram, ele deve ser capaz de corrigí-los;

• Satisfação: o sistema deve ser prazeroso de usar de forma que os usuários fiquem subjetivamente satisfeitos.

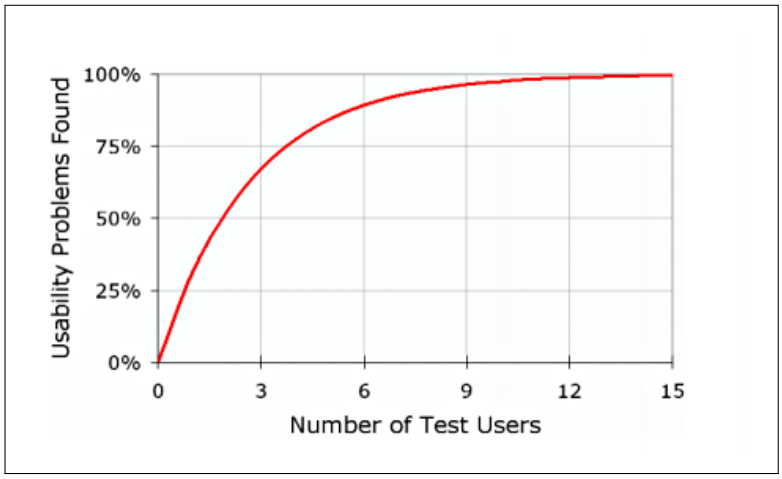
Para Preece (2002), a usabilidade está normalmente relacionada a dar garantias de que um produto interativo é fácil de aprender, possui o uso efetivo e agradável do ponto de vista dos usuários. Isso envolve otimizar a interação que as pessoas desempenham com produtos interativos que permitam a elas realizar suas atividades no trabalho, no estudo e na vida diária.

Para a execução de um teste de usabilidade, portanto, deve-se adotar alguns usuários representativos, como clientes para um site de comércio eletrônico ou funcionários para uma intranet. Esses usuários precisam executar tarefas representativas e devem ser observados no que fazem, onde eles conseguem interagir e onde eles têm dificuldades com a interface. É importante testar os usuários individualmente e deixá-los resolver quaisquer problemas por conta própria. Caso sejam direcionados ou ajudados para qualquer parte específica da tela, os resultados do teste foram contaminados (Nielsen, 2003).

Para identificar os problemas de usabilidade mais importantes de um sistema, o teste de cinco a dez usuários geralmente é suficiente. Em vez de executar um estudo grande e caro, é mais indicado utilizar os recursos para executar pequenos testes e revisar o design entre cada um para que seja possível corrigir as falhas de usabilidade à medida que as mesmas sejam identificadas. O design iterativo é a melhor maneira de aumentar a qualidade da experiência do usuário. Quanto mais versões e idéias de interface forem testadas com os usuários, melhor. É importante observar de perto os usuários individuais à medida que executam tarefas com a interface do usuário. Ouvir o que as pessoas dizem é enganador, é necessário observar o que os usuários realmente fazem (Nielsen, 2012).

Normalmente, entende-se que a usabilidade é muito dispendiosa e complexa e que os testes de usuários devem ser reservados para os complexos projetos de design web, com um enorme orçamento e um horário de tempo pródigo. Mas, na realidade, os resultados já são promissores a partir de testes com cinco usuários, executando pequenas tarefas. O número de problemas de usabilidade encontrados em um teste de usabilidade com n usuários é: N (1- (1- L) n), onde N é o número total de problemas de usabilidade no design e L é a proporção de problemas de usabilidade descobertos ao testar um único usuário (Nielsen, 2000):

Figura 24 – Gráfico para número de usuários versus problemas de usabilidade (NIELSEN, 2000).



À medida que se adiciona mais e mais usuários, aprende-se menos porque os mesmos comportamentos são vistos. Não há carência real de continuar observando o mesmo teste várias vezes (Nielsen, 2000).

2.9 CONSIDERAÇÕES FINAIS

O presente capítulo apresentou uma breve revisão teórica relativa aos principais conceitos de Teste de Software, SBSE, Metaheurísticas, Visualização de Dados, IHC e Usabilidade. De maneira geral, os conceitos apresentados ressaltam a importância da área de teste de software e a emergente Search-Based Software Engineering, bem como o embasamento conceitual de Visualização da Informação. A presente pesquisa é aplicada na fase de planejamento de testes e pretende facilitar a avaliação do profissional envolvido nesse processo. No capítulo seguinte serão apresentados os trabalhos relacionados a esta Pesquisa.

3 TRABALHOS RELACIONADOS

A seguir destacar-se-ão alguns trabalhos e livros que se relacionam à presente pesquisa. Primeiramente, serão abordados estudos que apresentam o contexto geral da área, bem como a dissertação de referência utilizada.

Em seguida, também serão expostos estudos de seleção e priorização de casos de teste, levando em consideração as metaheurísticas aplicadas nos mesmos.

Finalmente, serão discutidos trabalhos na área de SBSE que já tenham aproveitado e empregado às técnicas e os conceitos de visualização de dados.

3.1 CONCEITOS GERAIS

Conforme mencionado no capítulo 2, a área de Teste é, hoje, um dos campos mais estudados no contexto de SBSE. Vários aspectos do processo de validação e verificação de software já foram alvo de análises e trabalhos dos pesquisadores desse ramo. Os estudos são feitos desde a década de 70, relacionando Otimização Matemática com Engenharia de Software. Entretanto, a propagação desses conceitos se deu em 2001, quando o termo Search Based Software Engineering foi criado por Harman e Jones.

Especificamente para a abordagem de seleção e priorização de casos de teste, vários estudos também já foram realizados. A seleção de casos de teste é entendida pela escolha de quais casos de teste serão abordados e/ou executados em uma determinada versão do software. Já a priorização dos casos de teste se dá pela ordenação dos casos a partir de um critério, seja relevância para o cliente, complexidade da implementação etc.

Quanto à geração de casos de teste, temos o processo de reconhecimento das entradas e saídas válidas, levando em consideração as regras do sistema. Ferreira (2003) mostra algumas formas de gerar dados: de forma aleatória (dados são gerados até que uma entrada válida seja encontrada), de forma simbólica (dados simbólicos são atribuídos de forma que se possa caracterizar matematicamente o que o programa faz) e de forma dinâmica (as partes do programa são vistas como funções, onde os dados representarão retornos dessas funções).

Atualmente, por exemplo, já existem ferramentas para geração e seleção de casos de teste, que aceitam como entrada modelos probabilísticos e não probabilísticos.

3.2 PRINCIPAL REFERÊNCIA DA PESQUISA PRESENTE

O principal objeto de pesquisa desse trabalho trata-se de aplicar os tipos de visualização descritos, visando melhorar a saída de dados de uma aplicação obtida na dissertação de mestrado de Maia (2011), cuja finalidade é propor uma abordagem integrada, interativa e multiobjetiva dos problemas de seleção, priorização e alocação de casos de teste. Esses três problemas foram atacados de maneira sequencial (a saída de uma fase é a entrada para a fase seguinte) e implementados em três algoritmos multiobjetivos (NSGA-II, MOCell e SPEA2). O algoritmo SPEA2 apresentou, portanto, as melhores soluções para os três problemas citados. Com base nisso e de acordo com o que foi proposto por Maia (2011), essa metaheurística foi utilizada em dados reais de um software empresarial, o qual encontra-se em produção, apenas nas suas duas primeiras fases, de seleção e priorização (Arruda, 2015).

A partir dessa aplicação, surgiu a necessidade de sofisticar a maneira como as soluções geradas eram exibidas, de forma que o entendimento destas fosse mais esclarecido e também disseminado.

3.3 SELEÇÃO E PRIORIZAÇÃO DE CASOS DE TESTE A PARTIR DE OUTRAS METAHEURÍSTICAS

Um artigo estudado pelo Optimization in Software Engineering Group, em 2010, propôs a utilização de uma outra metaheurística para a solução do problema de priorização de casos de teste, a partir da técnica conhecida como caso de teste de regressão de priorização, também já abordada por Harman et al. (2007), tratando da maximização da qualidade da suíte de teste selecionada, fazendo uso também de uma implementação de algoritmo genético.

De uma forma geral, os artigos de Alander et al. (1997) e de Elbaum et al. (2003), também trazem uma idéia das possibilidades de testar softwares a partir de algoritmos genéticos.

A maior contribuição do trabalho de Yoo et al. (2007) é que o mesmo foi o primeiro a tratar o problema de seleção de casos de teste como um problema multiobjetivo. Os autores propõem duas aborgadens: a primeira considera cobertura de código e custo como funções objetivo, e a segunda considera cobertura de código, custo e histórico de detecção de falhas dos casos de teste como funções objetivo.

O trabalho de Rothermel et al. (2001) foi utilizado como base para muitos dos trabalhos relacionados à priorização de casos de teste. O objetivo deste trabalho é investigar se a priorização dos casos de teste aumenta a probabilidade de detectar erros no início dos testes. Oito

técnicas de priorização de casos de teste foram utilizadas no experimento, sendo implementadas utilizando algoritmos gulosos. As suítes de teste priorizadas por estas técnicas foram comparadas com suítes de teste não priorizadas.

3.4 FRENTE DE PARETO

O Princípio de Pareto também pode ser atribuído ao teste de software. A maioria dos problemas encontrados estão em uma quinta parte dos seus componentes (Pressman, 2001). Esse princípio trata da decisão de qual informação deve receber maior destaque. É a partir desse ponto que as técnicas de teste devem trabalhar, ou seja, é importante que se detecte quais são os casos de teste que tem a maior probabilidade de encontrar esses problemas.

Para o trabalho presente, utilizou-se o conceito de optimalidade de Pareto, que tem auxiliado os projetistas em suas decisões e processos. O participante articula sua preferência em relação aos diferentes objetivos, uma vez que tenha conhecimento da Frente de Pareto. A abordagem da visualização dessa frente tem sido amplamente utilizada na tomada de decisões para dois ou três problemas-objetivo, o que pode ser visualizado usando meios gráficos tradicionais 2D e 3D, como visto nas pesquisas de (Agrawal et al, 2005).

Também existem diferenças fundamentais entre problemas monoobjetivo e problemas que apresentam múltiplos objetivos. Em um problema de otimização monoobjetivo, existe apenas uma função objetivo que se pretende minimizar ou maximizar e cujas soluções estão sujeitas a um conjunto de restrições (de igualdade, desigualdade e condições limite) que devem ser satisfeitas para serem admissíveis para o problema. Já os problemas de otimização multiobjetivo apresentam várias funções objetivo, no mínimo duas, que na maior parte dos casos estão em conflito, e que se pretendem maximizar/minimizar em simultâneo, conforme abordado no trabalho de Ibrahim et al. (2004) e Maia (2011).

3.5 VISUALIZAÇÃO DE DADOS

O trabalho de CAIADO et al. (2001), sobre Visualização de Dados, descreve o design e implementação de um aplicativo para visualização interativa de grandes conjuntos de dados tridimensionais traduzidos na linguagem Java. A pesquisa apontou a mesma responsabilidade sobre o estudo de visualização de informações e dos tipos de gráficos apresentados para que a aplicação fosse desenvolvida.

No trabalho de Lima et al. (2006), uma interface gráfica foi implementada para o

tratamento dos dados sísmicos de reflexão, dentro do estudo de propagação de ondas e a partir dos conceito de visualização de dados, de maneira a facilitar a compreensão da massa de dados.

Já na pesquisa de Rodrigues, A. (2010), o conceito da visualização de dados (Tufte, 2001) é abordado para lançar uma análise sobre o atual estágio da infografia, contemplando não apenas a parte estrutural dos dados no ambiente informacional, mas estabelecendo vínculos estreitos com o conteúdo, interpretação, comunicação visual, design da informação e linguagem, dentro do ambiente de jornalismo.

O trabalho de Lunardi, et al. (2008), por sua vez, foi desenvolvido em duas etapas. A primeira etapa consistiu no desenvolvimento de uma aplicação integrada ao sistema de pesquisa do Yahoo para mostrar os resultados nas nuvens de texto e, a segunda, consistiu na avaliação da visualização de dados para tratamento dos dados do aplicativo, testando também seus benefícios.

No âmbito estudantil, o trabalho de Cavalcanti et al. (2011) propôs uma ferramenta open source de visualização de dados, desenvolvida com envolvimento de usuários do Ambiente Virtual Moodle. Essa ferramenta utiliza técnicas de visualização de informações para mostrar de maneira prática e rápida através de seus diversos tipos de gráficos, os acessos gerais e individuais aos recursos e atividades dos alunos ao ambiente virtual, permitindo a gestores, professores e tutores um acompanhamento prático do nível de interação dos estudantes.

Por fim, na área de Teste de Software, a pesquisa de Silva et al. (2013), desenvolveu uma ferramenta para a realização de processamento e análise de visualização de dados para instrumentos de astrofísica de altas energias. Para a equipe de desenvolvimento e para os cientistas responsáveis pelo instrumento científico, é importante haver uma ferramenta que permita a interação com os instrumentos ao nível de percepção do usuário final, aliada à execução de casos de teste e geração de relatos de testes de forma automática.

3.6 VISUALIZAÇÃO DE SOLUÇÕES EM SBSE

Existem poucas referências das aplicações e dos estudos realizados na área de visualização em SBSE. O livro Multiobjective Optimization (Otimização Multiobjetiva), de Branke et al. (2008) é que apresenta contribuições relevantes nesse sentido.

Basicamente, dois capítulos tratam sobre o assunto. Um deles aborda técnicas para a visualização do conjunto ótimo de Pareto, que são as técnicas mais discutidas e que podem ser usadas se o problema de otimização multiobjetiva considerado tiver mais do que duas funções objetivo. Em primeiro lugar, as lições aprendidas a partir de métodos desenvolvidos para

problemas monoobjetivos são consideradas. Em seguida, técnicas de visualização para converter problemas de otimização multiobjetivo com base em uma aproximação poliédrica da Frente de Pareto são discutidos. E finalmente, algumas técnicas de visualização são consideradas, usando uma aproximação pontual do conjunto ideal de Pareto.

O outro capítulo discute a visualização da Frente de Pareto e relaciona a diferença entre o espaço de decisão e o espaço objetivo. As técnicas abordadas destinam-se principalmente à visualização de vetores objetivos (pontos) que representam uma parte da viável região do objetivo. Tal interesse em visualizar a frente de Pareto é relacionado, principalmente, ao fato de que em problemas de otimização multiobjetiva, o foco está relacionado com os valores objetivos, e não com as variáveis de decisão.

Segundo Harman (2009), as técnicas de modelagem estatísticas de um comportamento futuro fazem parte da área de mineração de dados que se preocupa com probabilidades de previsão e tendências, variáveis também analisadas dentro do contexto de SBSE. Essa modelagem, portanto, segue tipos de visualização que facilitem o entendimento da natureza do problema, além se suas contribuições para que os pesquisadores de SBSE compreendam melhor suas landscapes (paisagens). Landscapes são extensões de paisagens que pode ser vistas em uma visão única.

Este capítulo apresentou algumas das principais contribuições encontradas na literatura para os problemas de seleção e priorização de casos de teste a partir de outras metaheurísticas, Frente de Pareto, Visualização para Tratamento de Dados e Visualização de Soluções em SBSE. Também foi apresentado o objetivo a partir da principal referência da pesquisa presente. Finalmente, não há trabalhos publicados na área de SBSE envolvendo o problema de tratamento de dados de saída de metaheurísticas, a partir dos conceitos de Visualização da Informação.

4 PROPOSTA

Apresenta-se neste capítulo a visão geral da proposta deste trabalho, bem como a configuração e saída da metaheurística, modelagem da proposta, propostas de visualização e informações sobre a aplicação do teste de usabilidade.

4.1 VISÃO GERAL

Como dito no capítulo 2, a teoria de visualização da informação propõe uma atividade de interpretação, a qual atribui um significado ao que é exposto ao usuário, onde essa interpretação é guiada pelos mapeamentos que o próprio tenha feito entre as variáveis (mentais) de interesse e as variáveis (físicas) do sistema.

Quanto às implementações em SBSE que fazem uso de alguma metaheurística, em geral, as mesmas exigem um ou mais arquivos de entrada de configuração, executam seus processamentos e expõem como saída, para o usuário, um arquivo texto com um conjunto de números, que representam conjuntos de boas soluções para o problema inicial.

Quando se trabalha com problemas em que os objetivos se encontram em conflito, não é possível obter uma solução ótima, mas um conjunto de soluções que constituem a Frente ótima de Pareto. Nestas circunstâncias, ter como solução do problema um conjunto de soluções "ótimas" pode ser entendido no sentido de não se poder afirmar que, nesse conjunto, uma solução é melhor do que outra. A comparação entre soluções não dominadas poderá ser operacionalizada à custa da estrutura de preferências do participante e a escolha de uma única solução ficará sempre vinculada às suas prioridades (Souza, 2010).

A questão é que, na maioria das situações, a saída na forma de sequência de números não apresenta um significado sem algum tipo de tratamento ou decodificação, até mesmo para os profissionais que já tem algum conhecimento do assunto.

Explorando essa forma de expor informações para o usuário e conhecendo o problema da grande massa de dados numéricos não significativos gerados por abordagens de SBSE, pensou-se em uma proposta que melhorasse a percepção e o entendimento do usuário das soluções geradas por uma ferramenta de SBSE para priorização de casos de teste.

4.2 CONFIGURAÇÃO E SAÍDA DA METAHEURÍSTICA

No caso do problema de priorização de casos de teste, a entrada da metaheurística é um arquivo que contém um identificador (ID) do caso de teste e informações, como, por exemplo, tempo de execução, complexidade, importância, etc. Já a saída é uma sequência de identificadores selecionados e ordenados de acordo com os objetivos que se desejam maximizar ou minimizar, o que é função da metaheurística.

Dado esse esforço adicional, o trabalho presente propõe que, após a execução da metaheurística, o arquivo gerado como saída, com as possíveis soluções, seja submetido a uma análise, na segunda fase, de priorização dos casos de teste, anteriormente já selecionados. Essa análise busca o tratamento dos dados e os apresentará graficamente, de uma forma acessível e intuitiva. Para a formulação dessa análise, que fará a conversão dos números em um gráfico, torna-se necessário analisar os parâmetros ou requisitos de cada caso de teste, de maneira que estes direcionem para o melhor entendimento da proposta.

Os gráficos que serão gerados, na fase de priorização, deverão mostrar um número específico, escolhido previamente, de casos de teste, para cada solução gerada pelo algoritmo do problema em questão. Através dessa representação será possível visualizar quais casos de teste se apresentam e suas relações, de acordo com um determinado parâmetro. Os casos de teste mais complexos, por exemplo, devem ser denotados no gráfico, sendo assinalados, de maneira a identificar se pertecem ou não à determinada solução.

Essa análise, com os gráficos, torna mais fácil a escolha da solução que mais se adequa à necessidade do usuário, em contrapartida à comparação de números, além de facilitar a comparação entre soluções e parâmetros, a partir do problema de múltiplos objetivos.

Pretende-se também prover a combinação de parâmetros nos gráficos, de modo a facilitar a viualização de mais de uma característica dos casos de teste, de acordo com a necessidade do avaliador.

4.3 MODELAGEM DA PROPOSTA

Na elaboração da abordagem proposta nesse trabalho, estabeleceram-se, inevitavelmente, objetivos específicos de investigação:

Objetivo 1 – Fornecer uma apresentação mais intuitiva e com informações mais acessíveis sobre as soluções de casos de testes propostas por uma metaheurística, que atualmente só fornece como saída um conjunto de identificadores, sem significado no mundo real.

Associado a esse objetivo, está a possibilidade de que o usuário entenderá de forma mais ágil as características das soluções propostas pela metaheurística, podendo escolher uma das propostas e dar sua opinião, já que não haverá mais o custo de verificar quais informações estão associadas a cada caso de teste.

Como forma mais ágil, considera-se que o tempo decorrido entre a apresentação gráfica das soluções e a percepção do usuário por uma solução específica será menor que o tempo para a mesma percepção quando as soluções são apresentadas na forma de uma sequência de identificadores. Desta comparação, chega-se ao primeiro caso.

Caso 1: o entendimento do usuário entre as soluções da metaheurística se dará de forma mais ágil.

Objetivo 2 – Melhorar a satisfação do usuário final na forma como as possíveis soluções geradas pela metaheurística serão apresentadas, facilitando sua escolha e percepção e tornando possível a escolha de uma proposta, que foi mais facilmente entendida.

Caso 2: uma solução gráfica na qual, são exibidas todas as informações do caso de teste, de acordo com um ou mais parâmetros previamente definidos, para cada solução, proporcionando ao usuário uma experiência mais satisfatória do que um texto com sequências numéricas, permitindo a sua participação na fase de priorização dos casos de teste, além do feedback sobre diferentes propostas visuais.

Com vista a atingir os objetivos propostos, foi necessário identificar quais as características que deveriam ser observadas e definir quais as funções e medidas das variáveis. Assim, alguns parâmetros globais foram utilizados para verificar os cenários, com base no trabalho de (MAIA, 2011):

- Complexidade;

- Grau de Importância.

Individualmente, como discorrido anteriormente, pode-se ter o gráfico das soluções dos casos de teste mais complexos e daqueles que apresentam maior importância. Essas características colaboram para a escolha de uma solução, com os casos de teste priorizados.

Através desse tratamento, espera-se que a fusão de características dos casos de teste facilitem ou se adequem à realidade de cada usuário e/ou sistema. Com essa visibilidade, o avaliador poderá participar, de forma mais facilitada e escolher uma das soluções geradas no gráfico. Assim, como saída final, teremos um gráfico com um conjunto de soluções de casos de teste priorizados, que possibilitará novamente a escolha por parte do avaliador.

Esses diferentes gráficos criados e aplicados através do teste de usabilidade englobam

a proposta a ser implementada pelo trabalho presente.

As abordagens propostas, com diferentes representações gráficas, serão implementadas, levando em considerações alguns fatores importantes nesse processo, além da participação do avaliador, os quais são (Nielsen, 2003):

- grau de satisfação, ou seja, o quanto a representação visual é consistente e abrange todas as informações necessárias para a escolha por parte do avaliador;

- grau de adequação, ou seja, o quanto a representação visual pode se adaptar à necessidade do avaliador e à combinação de parâmetros, sem que o entendimento seja afetado;

- grau de comparação, ou seja, o quanto uma proposta é mais confiável e compreensível que a outra.

Uma análise criteriosa envolve procedimentos bem projetados: têm-se cenários que devem ser testados e variáveis de interesse que devem ser controladas. Assim, torna-se necessário o conhecimento estatístico para validação dos resultados. Esses testes são importantes já que o projetista tem uma visão limitada e possui dificuldade em prever o comportamento dos usuários diante do apresentado e da abordagem proposta. Nesse caso, torna-se relevante também a coleta da opinião dos usuários envolvidos.

Alguns paradigmas e técnicas são apontados, como testes de usabilidade, estudos de campo, avaliação preditiva etc. Estes são procedimentos cujo objetivo é verificar o que acontece com os usuários ou se determinada possibilidade se confirma e colabora, de maneira efetiva, na validação dos resultados e na satisfação por parte dos usuários (Nielsen, 2003).

4.4 FERRAMENTA ESCOLHIDA PARA VISUALIZAÇÃO DE DADOS

A ferramenta Raw Graphs é construída em cima da biblioteca d3.js, por Mike Bostock. RAW.js é lançado sob a licença Apache 2 e está aberto à comunidade para melhorias ou para seus projetos pessoais/acadêmicos, que tem como principal objetivo tornar a representação visual de dados complexos fácil para todos. O projeto, liderado e mantido pelo Laboratório de Pesquisa de Densidade (Politecnico di Milano), foi lançado publicamente em 2013 e é considerado uma das ferramentas mais relevantes no campo de visualização de dados.

A figura 25 representa o layout da ferramenta para inserção de dados, por meio de planilha, documentos de texto, xml etc, enquanto que a figura 26 apresenta uma parte dos gráficos disponíveis para tratamento dos dados inseridos, a partir dos conceitos de Visualização de Dados.

Figura 25 – Ferramenta Raw Graph: inserção de dados (exemplo).

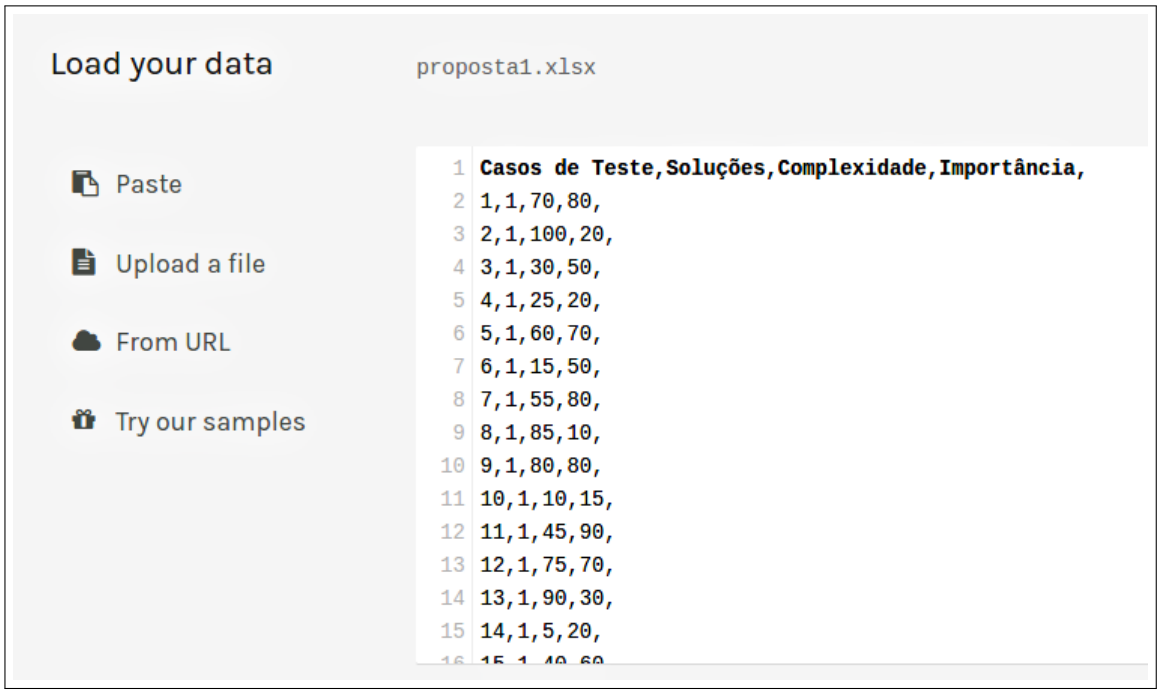
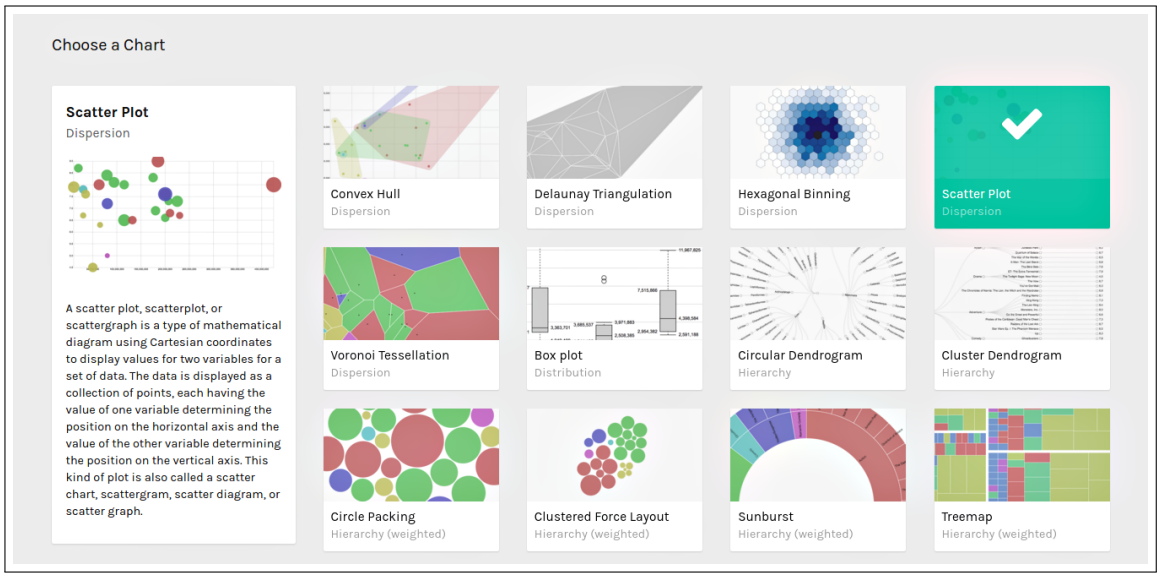


Figura 26 – Ferramenta Raw Graph: escolha da forma de visualização (exemplo).



4.5 PROPOSTAS DE VISUALIZAÇÃO

Um dos critérios relacionados à representação visual trata-se do Volume de dados. Alguns tipos de visualização procuram representar todo o conjunto de dados em uma única representação visual. No entanto, em alguns casos, não é possível representar toda informação de forma adequada devido à oclusão dos dados ou espaço insuficiente ou ainda por desordem visual (Luzzardi, 2003). Segundo Valiati (2008), quando esses problemas ocorrem, o tipo de visualização deve disponibilizar recursos interativos que possibilitem e auxiliem a análise dos dados.

O segundo critério utilizado para avaliar a representação visual é o uso de cores. Para Tufte (2001), o uso de cores em visualizações tem por objetivo rotular, medir, representar ou imitar a realidade, animar ou decorar. Segundo Few (2008), cores diferentes devem ser usadas apenas quando essas cores evidenciarem significados à visualização, auxiliando o usuário na interpretação dos dados.

O uso de rótulos também auxilia o usuário no processo de análise dos dados (Valiati, 2008). Este terceiro critério não está associado apenas ao uso de legendas. Segundo Yau (2010), se o gráfico utilizar os eixos coordenados X e Y, então os mesmos também devem ser rotulados. Além dos eixos, os elementos da visualização também podem ser rotulados.

A partir dessa perspectiva, foram criadas seis propostas, utilizando a ferramenta Rawgraphs, a partir do estudo dos tipos de visualização e gráficos, descritos na seção 2.5.6.

A seguir, as seis abordagens são apresentadas:

Proposta 1: A partir do estudo da disposição de círculos (de tamanhos iguais ou variados) em uma determinada superfície, de modo que não ocorram sobreposições, abordado pelo tipo de gráfico de Circle Packing, apresentado na seção 2.5.6, criou-se a proposta da figura 26, onde os círculos representam os casos de testes (com seus identificadores) e estão agrupados por solução. O raio do círculo indica a complexidade (quanto maior, mais complexo) e a cor indica a importância (quanto mais escuro, mais importante).

Proposta 2 e 3: Para identificar se o aumento ou diminuição de alguma variável influencia em outra, usou-se um gráfico de dispersão, o Scatter Plot. Os círculos representam os casos de testes (com seus identificadores) no eixo x e as soluções estão no eixo y, na proposta 2. Na proposta 3, em outra perspectiva, as variáveis são trocadas nos eixos. O raio do círculo indica a complexidade (quanto maior, mais complexo) e a cor indica a importância (quanto mais escuro, mais importante).

Figura 27 – Proposta 1 – Circle Packing.

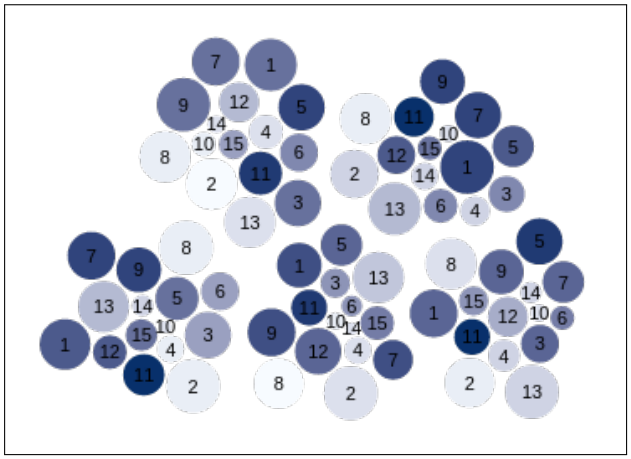


Figura 28 – Proposta 2 – Scatter plot.

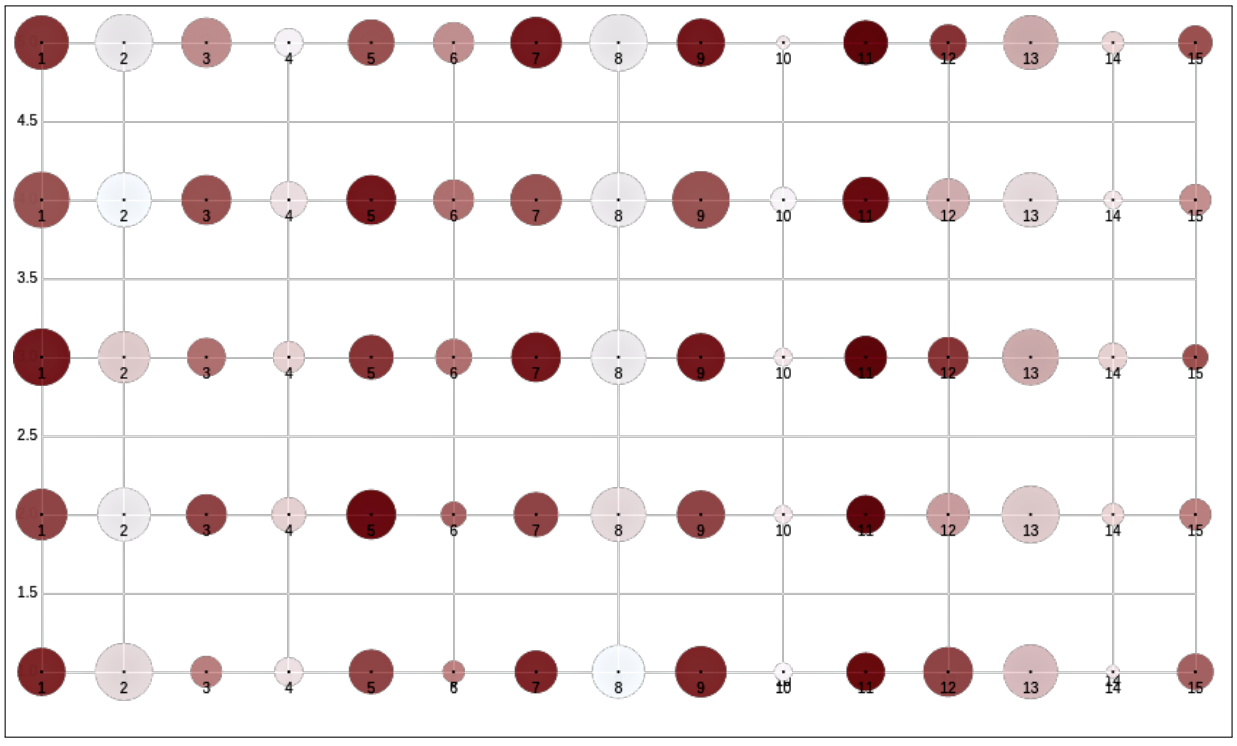
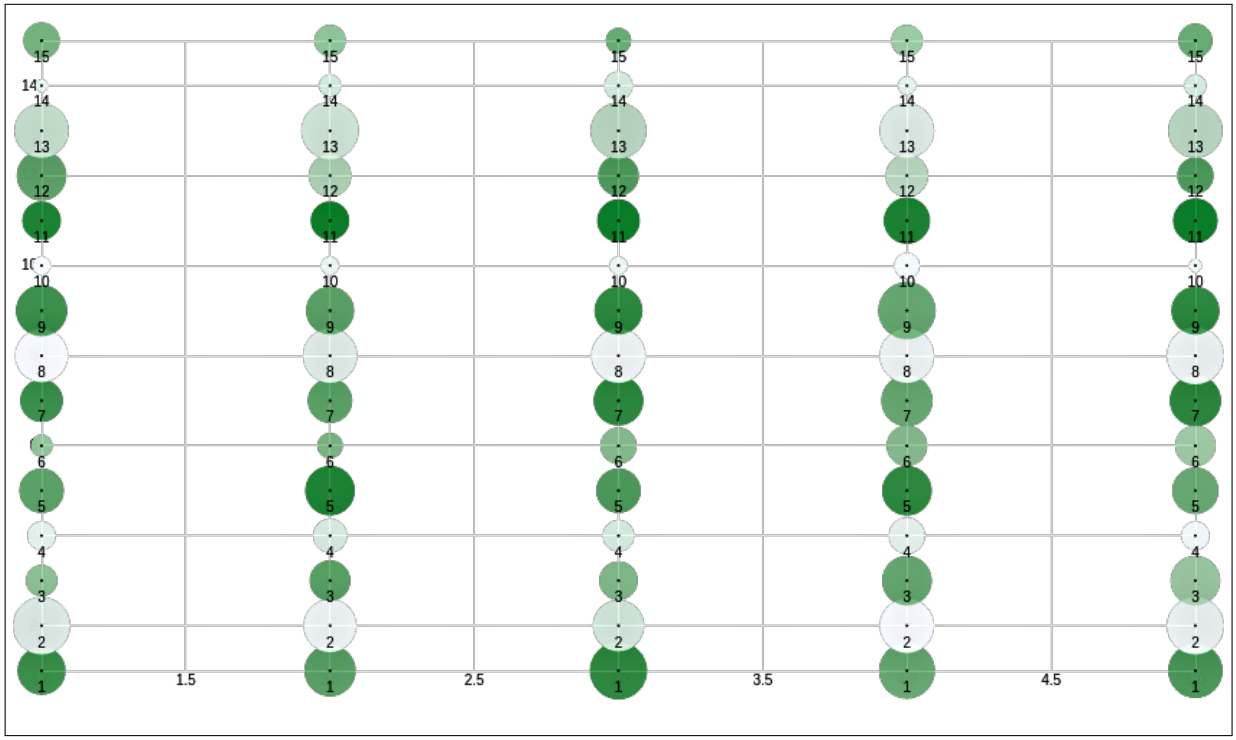
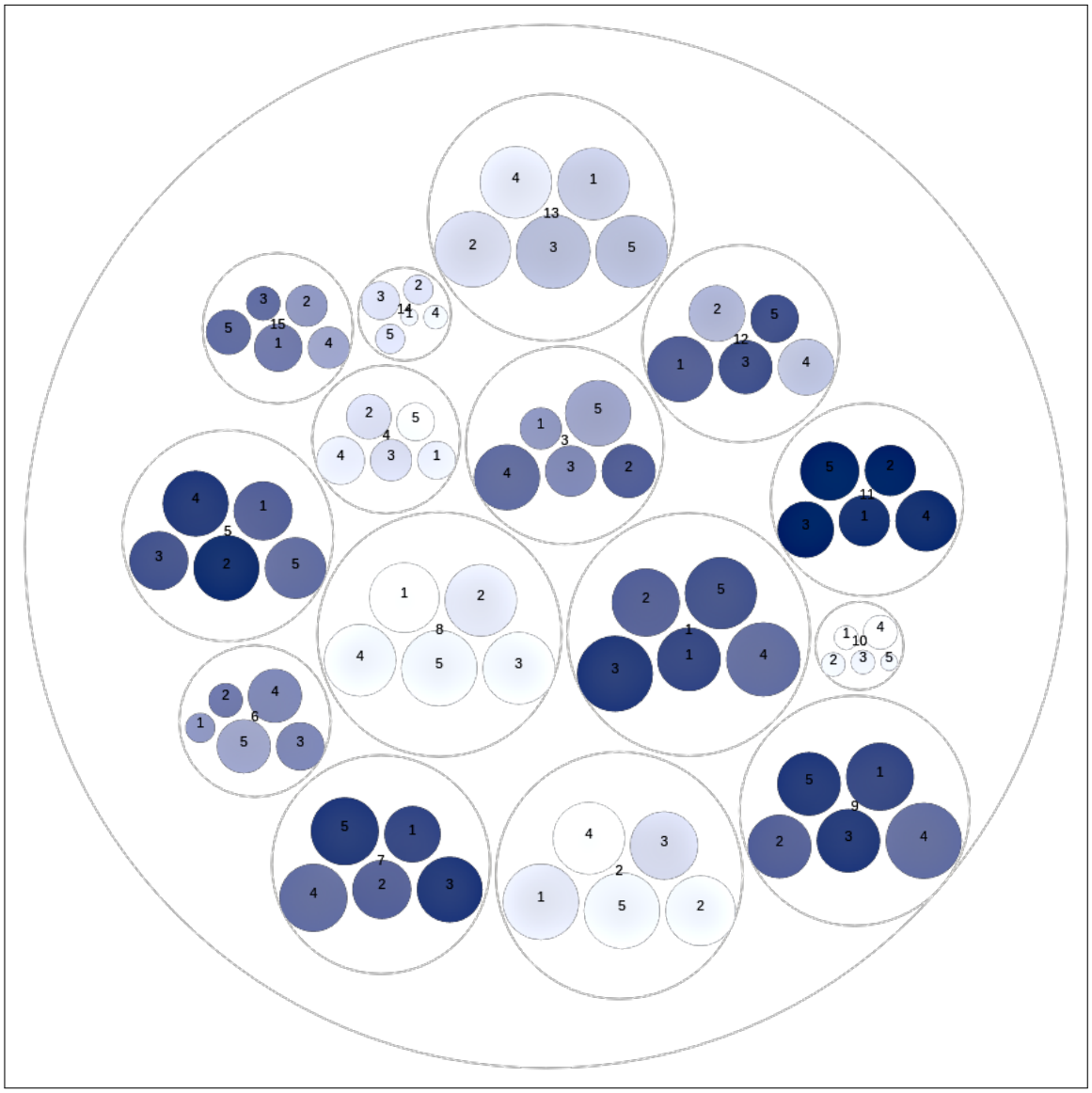


Figura 29 – Proposta 3: Scatter plot de outra perspectiva.



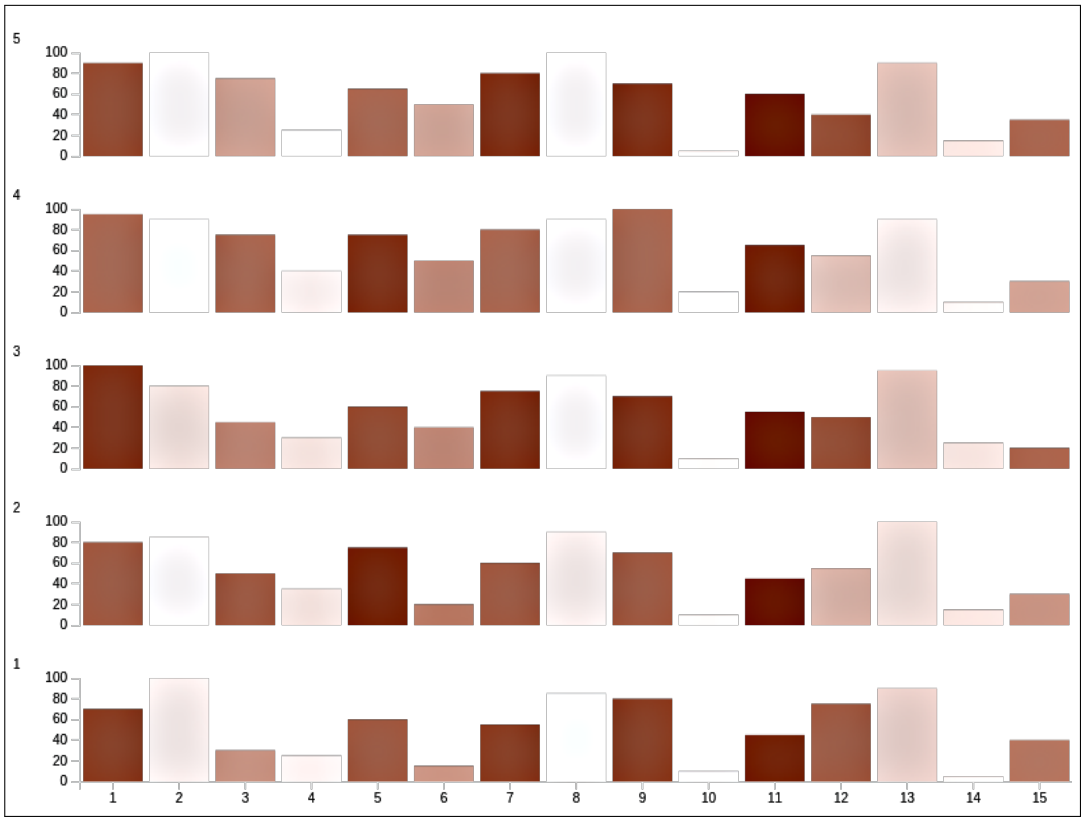
Proposta 4: Para classificação de objetos em diferentes grupos, onde cada um deve conter os objetos semelhantes segundo uma determinada função, empregou-se o gráfico de Clustered layout. Os círculos maiores indicam cada caso de teste (identificador central) e os menores (dentro deles) indicam a complexidade (com o raio do círculo) e a importância (com a cor) daquele caso de teste para cada solução.

Figura 30 – Proposta 4 – Clustered layout.



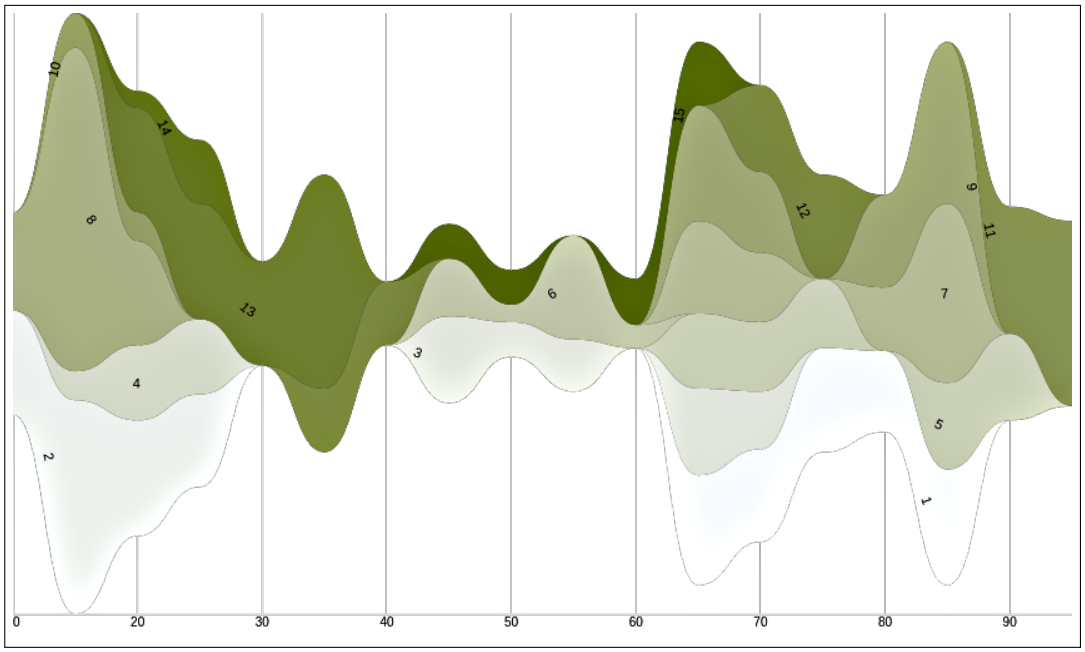
Proposta 5: A partir do conceito de que o gráfico de barras mostra os dados utilizando um número de barras de mesma largura, cada uma delas representando uma categoria particular, empregou-se também a proposta a seguir. Trata-se de um conjunto de 15 barras, representando os casos de teste, para cada solução. O tamanho da barra representa a complexidade (de 0 a 100) e a cor o grau de importância (quanto mais escuro, mais importante).

Figura 31 – Proposta 5 – Gráfico de barras.



Proposta 6: Para enfatizar a magnitude das mudanças ao longo do período, usou-se o gráfico de área (streaming). Esses gráficos são também utilizados para mostrar a relação das partes com o todo. As ondas representam a complexidade dos casos de teste, dispostos no eixo x. Ondas menores assinalam os casos de teste menos complexos, por exemplo. A cor, mais uma vez, representa o grau de importância (quanto mais escuro, mais importante).

Figura 32 – Proposta 6 – Gráfico de stream.



Este capítulo apresentou a abordagem proposta nesta dissertação. Os resultados do teste de usabilidade serão mostrados no Capítulo 5, a seguir.

5 AVALIAÇÃO DAS PROPOSTAS

Este capítulo detalha os resultados a partir do teste de usabilidade realizado para avaliar o desempenho da proposta descrita no capítulo anterior, visando a resolução dos problemas modelados nesta dissertação.

5.1 INFORMAÇÕES DO TESTE DE USABILIDADE

Para aplicação do teste, foi utilizado o software Camtasia, de gravação de vídeos, com o qual foi feita a avaliação do tempo gasto nas tarefas e das percepções dos participantes envolvidos.

O teste foi realizado no escritório de uma empresa de computação de Fortaleza, o Instituto Atlântico, com oito participantes que fazem parte da mesma e remotamente com os outros dois participantes. A empresa foi escolhida de maneira a facilitar a execução dos testes, levando em consideração que a autora do trabalho presente está profissionalmente alocada na mesma. A escolha dos participantes se deu através das redes profissionais e pessoais da pesquisadora e o número foi estabelecido com base em NIELSEN (2001), que recomenda a execução do teste de usabilidade com uma amostra de 5 participantes, sendo suficiente para encontrar a maioria dos problemas de uma interface.

O número 5, portanto, é um guia, mas em muitos casos é interessante testar com mais pessoas: quando a tarefa testada é um pouco mais complexa, com várias subtarefas e quando há mais de um perfil de usuário dentro do público-alvo e essa diferença afeta o uso do produto a ser testado. Mais importante do que a quantidade é a representatividade, que se trata do ato de testar com as pessoas certas e ter todos os perfis de participantes representados (Volpato, 2005). Logo, é viável, para a pesquisa presente, atingir os objetivos com uma amostra de dez participantes, que necessitam apresentar o perfil de Analista de Teste de Software.

A partir desse contexto, foram apresentadas as propostas gráficas, em ordem randômica, de um conjunto de cinco soluções, contendo uma suíte de quinze casos de teste cada uma. Os usuários tiveram a tarefa de escolher uma solução em cada proposta, identificando os fatores que apontavam a complexidade e o grau de importância dos casos de teste, justificando a sua escolha e explicando a sua opinião quanto à proposta que gerou o entendimento mais fácil e o mais difícil.

O teste foi realizado, primeiramente, apresentando o problema e o objetivo final para cada um dos participantes, através de um arquivo power point. As regras foram informadas,

sobre a filmagem realizada, o direito de imagem, os softwares utilizados e a orientação de que cada usuário deveria falar em tom de voz alto para que todas as percepções fossem atingidas com sucesso. A gravação identifica, portanto, a avaliação de cada proposta por parte do usuário, sendo possível registrar a variável de tempo, o comportamento de cada participante e sua respectiva opinião.

5.2 VISÃO GERAL DA AVALIAÇÃO

A partir da proposta lançada para tratamento e apresentação gráfica dos dados gerados pela metaheurística, obteve-se os dados do tempo de decisão do usuário, bem como informações sobre seu comportamento e entendimento dos gráficos. Com isso, foi possível verificar a melhor forma de visualização dentre as seis propostas. Outro objetivo alcançado foi a elevação do grau de satisfação do usuário ao manipular as informações graficamente, de forma mais intuitiva e eficiente, do que apenas ver os números dos casos de teste retornados pelo algoritmo.

De acordo com as evoluções já abordadas até hoje no que diz respeito à interação humano-computador, todas as propostas para melhoria da visualização gráfica dos dados, por parte do usuário, são mais facilmente compreendidas, do que representações númericas e/ou textuais (Nielsen, 2003). A elaboração do estudo de caso, com as diferentes propostas gráficas, tornaram possível a validação dos objetivos. Foi necessário a parametrização das características utilizadas para que fossem realizadas abordagens mais eficientes nesse processo e também para que fossem criadas possíveis ferramentas para esse tratamento de dados.

Seis propostas foram apresentadas e satisfeitas na execução do teste de usabilidade. A escolha por parte dos usuários, entre as soluções dispostas, foi ágil e precisa. A solução gráfica também colaborou com a satisfação dos usuários, visto que todas as informações do caso de teste foram exibidas.

Com as ideias exibidas em cada gráfico foi também possível manipular todos esses dados e parâmetros, de acordo com a necessidade do analista avaliador. Por exemplo, se o mesmo tiver a necessidade de avaliar casos de teste que sejam mais complexos e que, ao mesmo tempo, sejam mais importantes para o cliente, essa combinação de características poderá ser manipulada no gráfico, facilitando a visualização imediata.

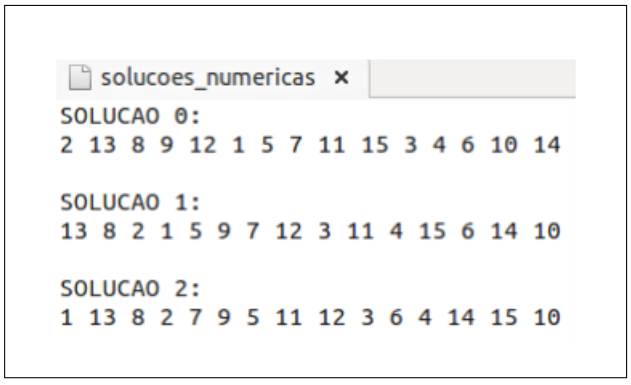
Para descrever melhor os insumos que foram utilizados, foi feita uma análise dos perfis de cada indivíduo envolvido, como propõe a otimização interativa. Vários problemas

reais de otimização não podem ser satisfatoriamente resolvidos utilizando-se somente processos automáticos. Exemplos desses problemas são o corte-bidimensional de tecido efetuado em confecções e a definição de horários de aula para uma universidade (onde não pode haver conflitos na alocação das salas de aula, e as preferências de horário propostas pelos professores devem ser atendidas o tanto quanto possível). Para tais problemas, o elemento humano é ainda necessário. O usuário é útil, por exemplo, para incorporar novas condições ao modelo computacional, para ajustar ou validar uma solução construída pelo computador ou para gerenciar os recursos computacionais disponíveis (Nascimento, 2003).

Alguns fatores da interação humano-computador foram estudados mais profundamente para otimizar os elementos visuais, a diagramação e a distribuição de informações, de forma que proporcionassem uma melhor interação com o indivíduo.

A figura 33 exibe o arquivo de texto que apresenta um conjunto de soluções adquirido a partir da execução do algoritmo para priorização de casos de teste. Como já demonstrado anteriormente, o arquivo não assinala facilmente as informações. Tem-se uma linha de números, onde cada número representa um determinado caso de teste e cada linha uma solução gerada. Exemplificando, tem-se o número "treze" aparecendo nas primeiras posições das soluções expostas, o que nos garante que esse caso de teste específico deve ter prioridade alta, de acordo com os parâmetros analisados. Esse comportamento é representado mais claramente, portanto, pela forma gráfica, como visto nas propostas.

Figura 33 – Exemplo de soluções numéricas utilizadas.



Os gráficos da seção a seguir apresentam os resultados obtidos com a execução do teste de usabilidade. Esses insumos foram, portanto, dados relevantes, a nível de estudos de comparação, na construção de outros gráficos, bem como na evolução do trabalho e na possível criação de uma ferramenta que englobe todos esses benefícios para a sociedade de Teste de Software e SBSE.

Os dados foram armazenados em uma planilha de teste de usabilidade (Rosemberg, 2015), que dispõe de parâmetros e campos para registro das informações.

5.3 INDICADORES UTILIZADOS

Segundo JORDAN (1998), cada método para a avaliação de interfaces gráficas digitais possui uma série de propriedades que fornecem certas vantagens ou desvantagens. Isto inclui, por exemplo, o tempo, o esforço e o nível de habilidade e conhecimento para a utilização do método, facilidades e equipamentos necessários para a condução eficaz do método, além do número mínimo de participantes para reunir informaçãoes úteis.

Com base nas propriedades mencionadas, atribuiu-se como indicadores numéricos, para o trabalho presente, as seguintes variáveis:

a) Tempo: período utilizado pelo participante no entendimento de cada proposta gráfica e na escolha das respectivas soluções;
b) Completude: percentual de plenitude na escolha das soluções, ou seja, na execução da atividade;
c) Habilidade: experiência do usuário com teste de software.

Conforme destacado por NIELSEN (2003) torna-se importante também, na aplicação do teste, a análise dos indicadores subjetivos, como a opinião dos participantes, sobre cada uma das propostas gráficas, bem como informações sobre as facilidades e dificuldades encontradas. Além disso, as expressões faciais captadas pelos vídeos gravados colaboraram para o reconhecimento e estudo no propósito do método.

JORDAN (1998) também afirma que não há algo capaz de substituir a observação de indivíduos tentando utilizar um determinado produto. Apesar dos princípios ergonômicos aplicados a projetos serem responsáveis por trazer grandes benefícios para os usuários, ou seja, no aprendizado das relações entre homem e máquina, em alguns casos estes usuários irão deparar-se com esforços não previstos. Neste momento, os mértodos que envolvem participantes terão um valor adicional, promovendo a descoberta de problemas de usabilidade até então desconhecidos. Similarmente, usuários podem estar aptos para lidar facilmente com alguns aspectos do produto que, de acordo com os princípios e convenções da ergonomia, esperava-se um certo esforço durante a sua utilização. É imporante ressaltar que estes fatos só podem ser descobertos através do envolvimento de participantes (usuários em potencial de determinada interface) durante a avaliação do teste de usabilidade.

5.4 ANÁLISE DOS DADOS

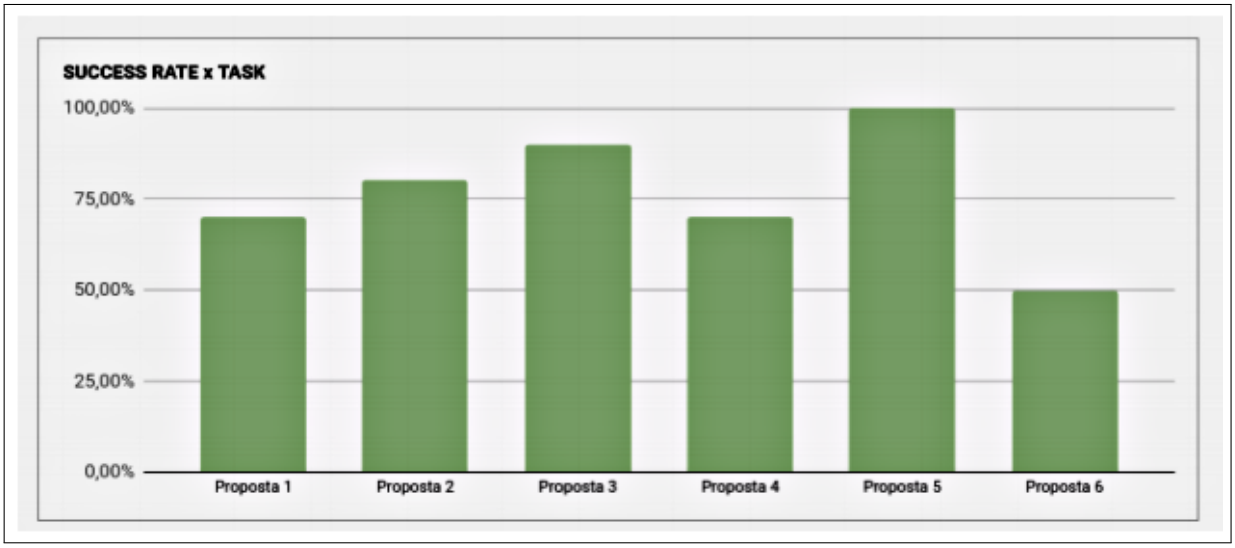
Primeiramente, a figura 34 apresenta uma tabela com os dados relacionados ao tempo, em segundos, que cada participante usou para entender e escolher uma solução, diante da proposta em análise. Na última coluna, a média de tempo de análise de cada proposta é apontada. É possível perceber, a partir dessa notação, as propostas que levaram mais tempo para serem analisadas e entendidas e as que levaram menos tempo. Esse indicativo é importante para considerar o grau de adequação, ou seja, a adaptação do usuário à proposta, conforme descrito na seção 4.3 e definido por (NIELSEN, 2003).

Figura 34 – Registro de tempo utilizado por cada participante na análise das propostas.

TASKS		HOW LONG DID THE TASK TAKE TO COMPLETE? (in linear seconds, for instance: 65 instead of 1 min 05 seconds)										
ID	Task	t1	t2	t3	t4	t5	t6	t7	t8	t9	t10	Average
1	Proposta 1	38	19	35	26	30	22	34	36	41	45	32,60
2	Proposta 2	21	25	13	23	38	24	28	20	27	20	23,90
3	Proposta 3	28	19	22	21	24	19	13	19	21	20	20,60
4	Proposta 4	42	53	34	46	33	39	41	40	38	42	40,80
5	Proposta 5	11	15	8	13	12	9	10	14	15	12	11,90
6	Proposta 6	45	51	35	44	38	41	10	32	9	28	33,30

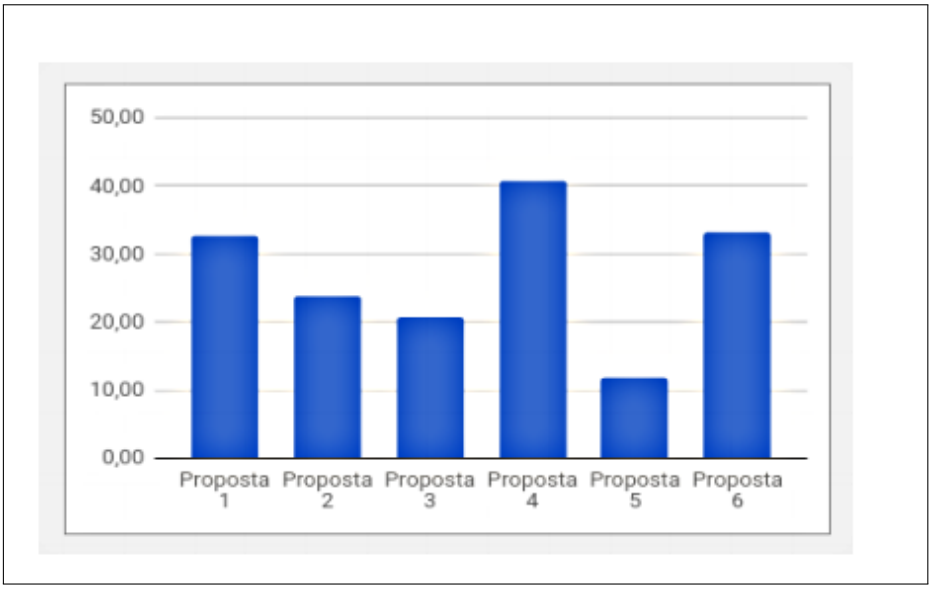
Conforme mencionado anteriormente também, a tarefa do usuário, na avaliação de cada proposta, foi de, não só entender o gráfico apresentado, como também optar por uma das soluções. A figura 35 exibe um gráfico que expõe, para cada proposta, o percentual de sucesso, ou seja, de completude da tarefa. Pode-se perceber, portanto, que a tarefa que corresponde à proposta 6 foi completa apenas por metade dos participantes, enquanto que a tarefa referente à proposta 5 foi completa por todos. Esse indicador, por sua vez, colabora com informações a nível de grau de comparação, destacando propostas visuais mais compreensíveis (NIELSEN, 2003).

Figura 35 – Percentual de completude das tarefas para cada uma das propostas.



Partindo de uma outra perspectiva e de acordo com a tabela da figura 34, temos o gráfico de comparação do tempo, em segundos, levado pelos participantes para completar as tarefas em cada uma das propostas, conforme apresentado na figura 36. Pode-se parametrizar, por exemplo, que as tarefas que demoraram mais de 30 segundos para serem compreendidas foram, portanto, eleitas como as mais complicadas de entender, como é o caso da proposta 1, 4 e 6.

Figura 36 – Médias dos tempos usados para execução das tarefas, em cada proposta.

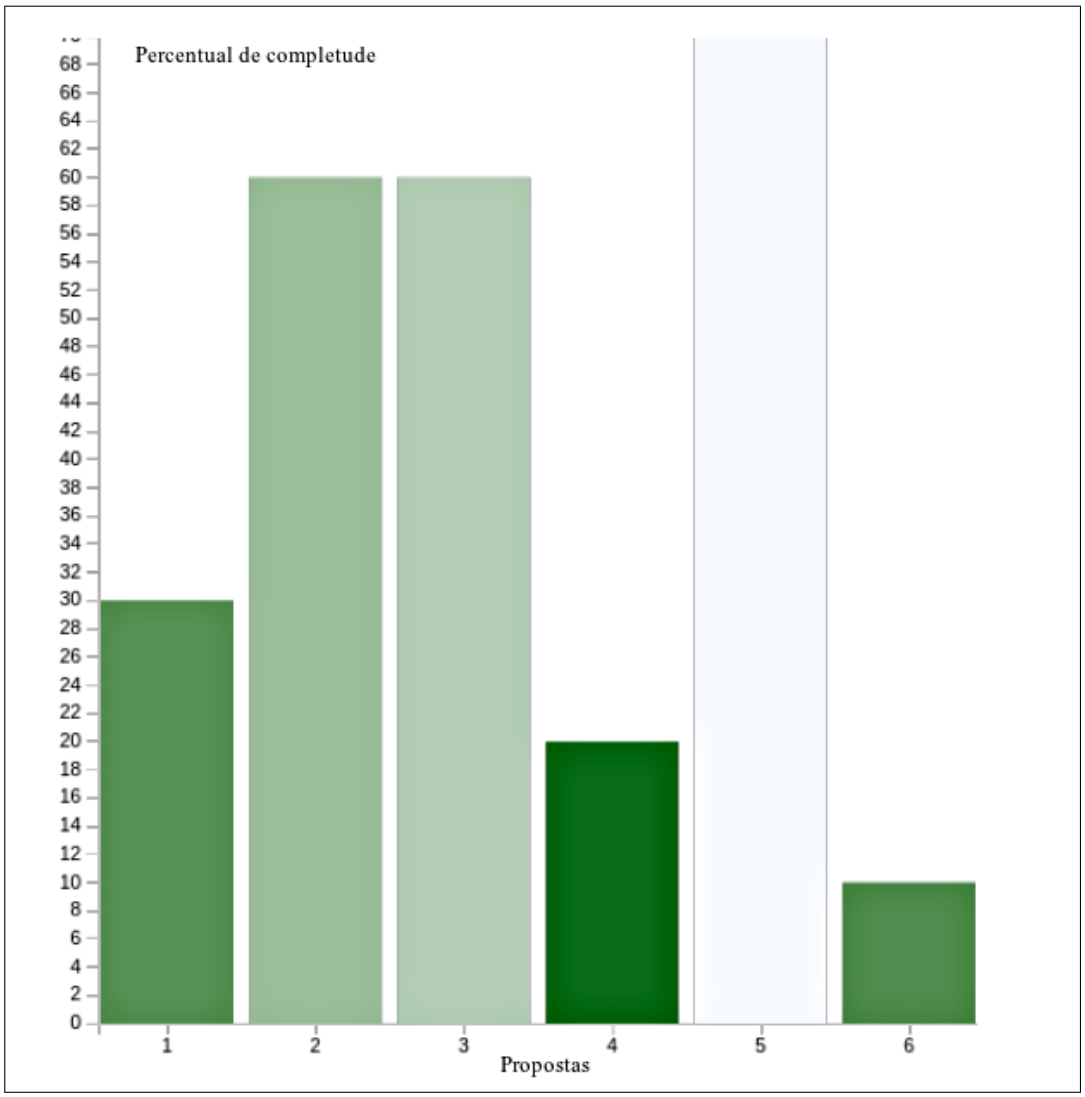


Também para efeito de comparação e medição do grau de satisfação, proposto por NIELSEN (2003), entre o tempo e a compreensão dos participantes em relação às propostas,

temos o gráfico da figura 37. O eixo x contém os seis gráficos utilizados no teste, enquanto o eixo y representa a porcentagem de compreensão dos mesmos, ou seja, de conclusão das tarefas relacionadas a cada proposta, de 0 a 100. A intensidade da cor verde utilizada denota a média dos tempos que os participantes levaram na análise de cada proposta, ou seja, quanto mais escuro, maior foi o tempo gasto na avaliação. Quanto maior a altura da barra, maior a facilidade de compreensão.

Partindo desse princípio, percebe-se que as propostas 1, 4 e 6 foram as que geraram maior desconforto aos participantes, tanto em sua percepção quanto ao tempo despendido na observação. Os participantes demoraram mais tentando compreender essas propostas e em alguns casos houve, inclusive, desistência.

Figura 37 – Comparação entre o tempo e a compreensão dos participantes (indicado pela intensidade da cor - quanto mais escuro, mais difícil), nas seis propostas.



Conforme mencionado no capítulo anterior, o teste foi aplicado à amostra de dez participantes. A figura 38 apresenta o modelo da tabela encontrada em (Rosemberg, 2015), com a descrição da função profissional de cada participante, onde (localização) o teste foi executado com cada um deles e os recursos utilizados, preenchida com as informações do teste da pesquisa presente.

Figura 38 – Descrição dos participantes.

PARTICIPANTS		Details about the participants.			
ID *	Profile / Persona *	Description	Test location	Device	Equipment / tool
1	Paulo Abner	Test Analyst	Office	Notebook	Camtasia
2	Ugo Castro	Test Analyst	Office	Notebook	Camtasia
3	Felipe Mota	Test Analyst	Office	Notebook	Camtasia
4	Amanda Chaves	Test Analyst	Office	Notebook	Camtasia
5	Patrícia Beserra	Test Analyst	Office	Notebook	Camtasia
6	Aline Chaves	Test Analyst	Remote	Notebook	Camtasia
7	Alano Pinto	Test Analyst	Remote	Notebook	Camtasia
8	Rodrigo Araújo	Test Analyst	Office	Notebook	Camtasia
9	Henrique Luz	Test Analyst	Office	Notebook	Camtasia
10	Rubens Brilhante	Test Analyst	Office	Notebook	Camtasia

Cada participante recebeu as informações sobre o problema a partir de um power point e teve conhecimento do objetivo e das regras do teste, sobre filmagem e softwares utilizados. O fluxograma de atividades seguido por cada um deles é detalhado da seguinte forma:

a) O participante analisou a proposta 1 e escolheu uma das cinco soluções apresentadas;
b) O participante analisou a proposta 2 e escolheu uma das cinco soluções apresentadas, sem retornar para a proposta anterior;
c) O participante analisou a proposta 3 e escolheu uma das cinco soluções apresentadas, sem retornar para as propostas anteriores;
d) O participante analisou a proposta 4 e escolheu uma das cinco soluções apresentadas, sem retornar para as propostas anteriores;
e) O participante analisou a proposta 5 e escolheu uma das cinco soluções apresentadas, sem retornar para as propostas anteriores;
f) O participante analisou a proposta 6 e escolheu uma das cinco soluções apresen-

tadas, sem retornar para as propostas anteriores;

g) O participante apontou, dentre os seis gráficos analisados, qual a proposta que o mesmo julgou como melhor, justificando sua escolha;

h) O participante apontou, dentre os seis gráficos analisados, qual a proposta que o mesmo julgou como pior, justificando sua escolha.

As opiniões dos participantes também foram observadas, tanto pelo comportamento, como pelas palavras ditas no momento de análise das propostas, por cada um deles, através de um estudo criterioso dos vídeos produzidos. Alguns pontos relevantes foram observados, como, por exemplo, o fato da proposta do gráfico de stream (proposta 6) ter sido a menos entendida:

"Esse gráfico eu não tive vontade de analisar. Está muito confuso. Desisti logo." - Participante 9

"Não consigo discernir uma solução da outra e nem tão pouco identificar onde estão os casos de teste" - Participante 7

Outro ponto importante, que foi percebido, foram as distintas formas de entendimento de uma mesma proposta:

"O gráfico de barras, apesar de simples, me deixou confuso com essas cores" - Participante 2

"O entendimento mais rápido, sem dúvida, é atráves desse de barras, vejo claramente as diferentes soluções e os parâmetros que estão sendo analisados" - Participante 3.

5.5 DISCUSSÃO DA AVALIAÇÃO

Conforme demonstrado nos gráficos da seção anterior, chegou-se a conclusão que a proposta que obteve o melhor entendimento foi a proposta do gráfico de barras (proposta 5). Este gráfico resume um conjunto de dados categóricos. Ele mostra os dados utilizando um número de barras de mesma largura, cada uma delas representando uma categoria particular de cada caso de teste. A altura de cada barra é proporcional a uma agregação específica, enquanto a cor pode representar uma outra categoria ou característica. Comparando o tempo utilizado para análise dessa proposta e o comportamento do usuário durante essa tarefa, bem como os comentários feitos, é possível perceber a maior facilidade e interação de cada participante com esse tipo de gráfico.

Segundo Kelley et al. (2009), o gráfico de barras define proporcionalidade às variáveis que representa, podendo ser desenhado tanto da forma horizontal como vertical. Esse

tipo de representação, portanto, ilustra claramente as comparações e facilita o entendimento por parte do usuário. Um eixo do gráfico mostra especificamente o que está sendo comparado enquanto o outro eixo representa valores discretos.

Em contrapartida, o gráfico que gerou o maior número de questionamentos, levando em consideração às opiniões faladas e as expressões colhidas pelos vídeos gravados, foi a proposta do gráfico de stream (proposta 6). A maioria dos participantes apresentou dificuldades em, basicamente, diferenciar uma solução da outra e analisar a ordem de cada caso de teste. Um gráfico de stream, também conhecido como gráfico de fluxo é um tipo de gráfico de área empilhada, que é deslocado em torno de um eixo central, resultando em uma forma orgânica fluente (Kelley et al., 2009).

A figura a seguir apresenta uma tabela simples, também encontrada em ROSEMBERG (2015), com o registro das tarefas realizadas pelos participantes no entendimento de cada proposta. As tarefas que foram bem-sucedidas recebem "1" e representam os usuários que entenderam a proposta e que conseguiram escolher uma solução, finalizando sua atividade. Já as tarefas que não foram bem-sucedidas são representadas pelo espaço vazio e denotam os participantes que não conseguiram compreender o gráfico e que, portanto, não concluíram sua atividade.

Figura 39 – Resultados das percepções dos participantes.

TASKS		WHAT PARTICIPANTS FINISHED THE TASK?									
ID	Task	p1	p2	p3	p4	p5	p6	p7	p8	p9	p10
1	Proposta 1		1	1		1	1		1	1	1
2	Proposta 2	1		1	1	1		1	1	1	1
3	Proposta 3	1	1	1	1		1	1	1	1	1
4	Proposta 4	1	1	1	1	1	1	1			
5	Proposta 5	1	1	1	1	1	1	1	1	1	1
6	Proposta 6	1	1					1	1		1

Analisando a tabela, pode-se perceber que a proposta 6, que é a do gráfico de stream, apresenta a maior quantidade de espaços vazios, ou seja, foi a proposta que gerou mais questionamentos e dúvidas por parte dos participantes que, em sua maioria, não alcançaram o entendimento do gráfico.

A compreensão dos dados obtidos atráves do teste de usabilidade é de grande importância para construção de novas propostas e/ou para melhoria dos gráficos já utilizados. A partir das orientações sobre os trabalhos futuros, descritas no próximo capítulo, pode-se utilizar os resultados da pesquisa presente de maneira a incorporar e melhorar a saída dos dados de metaheurísticas, no campo de SBSE.

Este capítulo apresentou os resultados do teste, de modo a validar a abordagem proposta. Os seis gráficos criados foram detalhados a partir da análise por meio dos participantes envolvidos. O capítulo seguinte discutirá as conclusões e apresentará idéias e limitações para trabalhos futuros com a abordagem proposta nesta dissertação.

6 CONCLUSÕES E TRABALHOS FUTUROS

A seleção e priorização de casos de teste é uma atividade rotineira do ciclo de desenvolvimento de testes e, por envolver diversos aspectos, muitas vezes conflitantes, a definição de quais casos de teste devem ser implementados em cada liberação é uma tarefa complexa cognitiva e computacionalmente.

O problema de priorizar a sequência de funcionalidades a serem testadas para os clientes, é conhecido como um problema para a área de teste de software e tem sido bastante abordado em SBSE. Porém, não existem trabalhos com foco nas experiências visuais que o analista de teste, nesse caso, pode possuir durante o processo de geração das soluções.

Nesse contexto, o presente trabalho teve como principal objetivo propor uma abordagem interativa usando as teorias de visualização de dados propostas, de modo a aliar a expertise humana ao poder computacional dessas técnicas.

6.1 CONTRIBUIÇÕES DO TRABALHO

Os resultados obtidos demonstraram que além de utilizar as preferências do usuário para guiar esta priorização, a abordagem foi capaz de satisfazer objetivos específicos da área, priorizando aqueles casos mais importantes e com pouco comprometimento de outros fatores como o valor de negócio e o risco de projeto. A partir da proposta que foi lançada para tratamento e apresentação gráfica dos dados gerados pela metaheurística, o grau de satisfação e o tempo despendido na análise foram avaliados. O objetivo foi alcançado ao manipular as informações graficamente, de forma mais intuitiva e eficiente. Foi identificada, portanto, a melhor proposta visual dentre algumas selecionadas para a análise, o que provê insumos e informações relevantes para futuros trabalhos.

6.2 LIMITAÇÕES

Pode-se citar três limitações para a abordagem proposta. A primeira limitação é a quantidade de participantes envolvidos no teste de usabilidade. Uma amostra maior, com diferentes experiências, enriqueceria mais ainda o trabalho presente. A segunda limitação é o tempo para levantar as variáveis comportamentais de cada parâmetro, a partir dos participantes. Percepções relacionadas às respostas e ao comportamento de cada profissional envolvido requer conhecimento e tempo. A terceira limitação se refere ao uso de dados gerados na abordagem.

A suíte de caso de teste escolhida possui apenas dez casos e as soluções geradas foram apenas cinco. Os resultados seriam mais realistas se os dados utilizados fossem aplicados à uma massa maior de dados.

6.3 TRABALHOS FUTUROS

Motivado pelo fato das constatações iniciais apontarem resultados promissores, podem ser estudadas e implementadas complementações que deixarão a proposta ainda mais realista e aplicável. Dentre essas melhorias estão, a melhoria dos gráficos de análise, bem como a aplicação dos resultados obtidos no futuro desenvolvimento de uma ferramenta, que também possibilite a melhoria da entrada dos dados e a priorização de casos de teste, por meio das soluções geradas e apresentadas graficamente.

REFERÊNCIAS

ABREU, B. T. Uma abordagem evolutiva para a geração automática de dados de teste. Dissertação de mestrado, IC/Unicamp, Campinas - SP, 2006.

ANTONIOL, G., KPODJEDO, S., RICCA, F., GALINIER, P. Evolution and Search Based Metrics to Improve Defects Prediction. In: Proceedings of the 1st International Symposium on Search Based Software Engineering, p. 23-32, 2009.

AGNER, Luiz. Ergodesign e Arquitetura da Informação: trabalhando com o usuário. Editora Quartet, Rio de Janeiro: 1959.

AGRAWAL, G. et al. Intuitive Visualization of Pareto Frontier for MultiObjective Optimization in n-Dimensional Performance Space. University at Buffalo, New York, 2008.

ALVES, R. Desenho e Implementação de um Ambiente de Programação Visual para Robótica Educacional. Universidade Federal do Rio de Janeiro. PPG, 2014.

ANDRADE, T.C.; FREITAS, F.G.; COUTINHO, D.P.; SOUZA, J.F. Uma Abordagem de Otimização Multiobjetiva para o Problema da Priorização da Correção de Defeitos. I Workshop Brasileiro de Otimização em Engenharia de Software, 2010.

ARRUDA, L.V. Uma abordagem de otimização para seleção e priorização de casos de teste a partir de dados reais. Universidade Estadual do Ceará – Centro de Ciência e Tecnologia. Monografia da Especialização em Engenharia de Software com Ênfase em Padrões. Fortaleza, 2015.

BARBOSA, S.D.J. Introdução à Teoria e Prática da Interação Humano Computador fundamentada na Engenharia Semiótica. In Tomasz Kowaltowski and Karin Breitman (orgs.) Atualizações em informática, 2010.

BASTOS, R. Gerência de Projetos e Processos de Desenvolvimento de Software: uma proposta de integração. Faculdade de Informática - Pontificia Universidade Católica do Rio Grande do

Sul, 2007.

BATISTA, J. et al. Desenvolvimento de programas educativos por prototipagem continuadamente evolutiva. Simpósio de Investigação e Desenvolvimento de Software Educativo, Coimbra, 1999.

BINDER, R.V. Testing object-oriented systems: models, patterns, and tool; Addison-Wesley, 2000.

BOWERMAN, S., Lara, J. C., Lidstrom, M. E. Chistoserdova, L., Methylotenera Mobilis gen. nov., sp. nov, an Obligately Methylamine-Utilizing Bacterium Within the Family Methylophilaceae. International Journal of Systematic and Evolutionary Microbiology, 56, 2819-2823 (2004).

BRANKE, J., Deb, K., Miettinen, K., Slowin,ski, R. (Eds.). Multiobjective Optimization. Interactive and Evolutionary Approaches. Springer-Verlag Berlin Heidelberg 2008.

BUENO, P., JINO, M. Automatic test data generation for program paths using genetic algorithms. International Journal of Software Engineering and Knowledge Engineering, 2002; 12(6):691-709.

CAIADO et al., NISVAS, Three-Dimensional Interactive Visualization in Java3D, Conference: 14th Brazilian Symposium on Computer Graphics and Image Processing (SIBGRAPI, 2001).

CARROLL, J. M. (2003) HCI Models, Theories, and Frameworks: Toward a multidisciplinary science. San Francisco Morgan Kaufmann Publishers. p.576.

CAVA, R.A; FREITAS, C.M.D.S. Node-Edge Diagram Layout for Displayng Hierarchies. Simpósio Brasileiro de Computação Gráfica e Processamento de Imagens. Santa Catarina, 2001.

CAVALCANTI, et al., Visualização de dados aplicados em educação à distância no processo de avaliação ao aluno. Universidade Federal de Pernambuco, 2011.

CEMIN, C. Visualização de Informações Aplicada à Gerência de Software (Dissertação) — Instituto de Informática, UFRGS, Porto Alegre, Rio Grande do Sul, Brasil, 2011.

COHON, J.L.: Multiobjective Programming and Planning. Academic Press, New York (1978).

CYBIS, W. et al. (1999). “Ergonomia em software educacional: A possível integração entre usabilidade e aprendizagem”, In II Workshop sobre Fatores Humanos em Sistemas Computacionais, Florianópolis - SC.

DEB, K. Multi-Objective Optimization using Evolutionary Algorithms. John Wiley Sons, 1st Edition, 2008. COLLETTE, Y., SIARRY, P. Multi-Objective Optimization: Principles and Case Studies. Springer, 2003.

DIX, Alan; FINLAY, Janet; ABOWD, Gregory D.; BEALE, Russel. Human-Computer Interaction. New York, Prentice Hall: 1998.

DREO, J.; PETROESKI, A.; SIARRY, P.; TAILLARD, E. Metaheuristics for Hard Optimization: Methods and Case Studies, 1 ed., Springer, 2005.

DWYER, D. et al. “Ensinando com tecnologia; criando salas de aula centrada nos alunos”, Porto Alegre, Artes Médicas, 2002.

ELBAUM, S., ROTHERMEL, G., KANDURI, S., MALISHEVSKY, A. G. Selecting a CostEffective Test Case Prioritization Technique. Software Quality, v. 12, n. 3, P. 185-210, 2004.

DEB, K. Multi-Objective Optimization using Evolutionary Algorithms. John Wiley Sons, 1st Edition, 2003.

FAREBROTHER, R. W. Fitting Linear Relationships: A History of ther Calculus of Observations 1750-1900. New York: Springer. 1999.

FEI LI; J. Nieh (2010). Low-complexity interpolation coding for server-based computing.

FERREIRA, Luciano P. Tdsgen - Uma Ferramenta de Geração de Dados de Teste Baseada em Algoritmos Genéticos. Universidade Federal do Paraná - Setor de Ciências Exatas. Dissertação de Mestrado. Curitiba, 2003.

FERREIRA, A. Coesão em Equipes de Engenharia de Software: Um Mapeamento Sistemático. Universidade Federal de Pernambuco. Centro de Informática. 2012.

FREED, Larry, (2008). Freed Your Mind, selected articles on successful application of 'Voice of Customer' metrics.

FREITAS, F. G.; MAIA, C. L. B.; CAMPOS, G. A. L.; SOUZA, J. T. Otimização em Teste de Software com Aplicação de Metaheurísticas. Revista de Sistemas de Informação da FSMA,(2010), pp. 3-13.

FREITAS, C., CHUBACHI, O; LUZZARDI, P.; CAVA, R. Introdução à Visualização de Informações. Revista de Informática Teórica e Aplicada (RITA) 8(2):143-158, 2001.

FEWSTER, Mark; GRAHAM, Dorothy. Software Test Automation. Addison-Wesley Professional; 1st edition, 1999.

FILHO, João Gomes. Gestalt do Objeto: sistema de leitura visual da forma. Editora Escrituras, São Paulo: 2001.

FISHER, A. Porter. "The Past, Present, and Future of MOOCs and their Relevance to Software Engineering". In Proceedings of the on Future of Software Engineering, pp. 212-224, 2013.

FUNKHOUSER, H. G. (1936). A note on a tenth century graph. Osiris, 1, 260-262.

GLOVER, F., KOCHENBERGER, G. Handbook of Metaheuristics, Springer, 1st edition, 2002.

GLOVER, F., Future paths for integer programming and links to artificial intelligence, Computer Operational Research 13, pp. 533-549., 1986. HARMAN, M. The Current State and Future of Search Based Software Engineering, International Conference on Software Engineering - Future of Software Engineering, IEEE Computer Society Washington, DC, USA, 2007, pp. 342-357. Computation , 6, 2, Abril.

GOODWIN, K. (2009). Designing for the digital age: How to create human-centered products and services. Indianapolis, IN: Wiley Publishing.

GRAUER, M., Lewandowski, A., Wierzbicki, A.: DIDASS – theory, implementation and experiences. In: Grauer, M., Wierzbicki, A. (eds.) Interactive Decision Analysis, vol. 229, pp. 22–30. Springer, Berlin (1984).

GREENBERG, S., Carpendale, S., Marquardt, N., Buxton, B. (2012). The Narrative Storyboard: Telling a story about use and context over time. Interactions, January+February, 64-69.

HARMAN, M.; JONES, B. F. Search-based software engineering. Information and Software Technology, pp. 833-839. 2001.

HARMAN, M. LI, Z.; HIERONS, M. (2007). Search Algorithms for Regression Test Case Priorization.

HARMAN, M. "Why the virtual nature of software makes it ideal for search based optimization", 13th International Conference on Fundamental Approaches to Soft. Engineering (FASE 2009), vol. 6013, pp. 1-12, 2009.

HEALEY, C.G. Building a Perceptual Visualisation. Architecture, Behaviour Information Technology 19(5):349-366, 2000.

HERMAN, I; MELANÇON, G; MARSHALL, M.S. Graph Visualization and Navigation in Information Visualization: A Survey. IEEE Transactions on Visualization and Computer Graphics, 6(1):24-42, 2000.

HUGO, A. D. do Nascimento and Peter Eades. “User Hints for Directed Graph Drawing”,9th Graph Drawing Conference (GD 2001), Lectures Notes on Computer Science, vol. 2265, 2002, pp. 205-219.

IBRAHIM, A. et al. 3D-RadVis: Visualization of Pareto Front in Many-Objective Optimization. Software Engineering University of Ontario Institute of Technology Oshawa, Canada, 2016.

JAIN, A. “SimVBSE: Developing a Game for Value-Based Software Engineering. Proceedings 19th Conference on Software Engineering Education and Training”. Paginas 103 -114, 2009.

JOHNSON, B.; SHNEIDERMAN, B. TreeMaps: A space-filling approach to the visualization of hierarchical information structures., Proceedings of IEEE Visualization, San Diego, pp. 284-291, 1991.

JOKELA, T. (2003). When good things happen to bad products: where are the benefits of usability in the consumer appliance market? Interactions Vol. 11, Issue 6- Nov/ Dez.

JORDAN, P. W. An Introduction to Usability. Londres: Taylor Francis Ltda., 1998.

KAPFHAMMER, G. M. Using coverage effectiveness to evaluate test suite prioritizations. In: Proceedings of the 1st ACM International Workshop on Empirical Assessment of Software Engineering Languages and Technologies: Held in Conjunction with the 22Nd IEEE/ACM International Conference on Automated Software Engineering (ASE), 2016.

KJELDSKOG, J., Stage, J. (2004). New techniques for usability testing of mobile systems. Internati- onal Journal of Human-Computer Studies, 60, 599-620.

KLEIN, G. et al. “Improvements of skills for solving ill defined problems”. Educational Psychologist 13:13-41, 1997.

KORHONEN, P.: A visual reference direction approach to solving discrete multiple criteria problems. European Journal of Operational Research 34, 152–159 (1988).

KRUG, S. (2000). Don’t Make Me Think! A Common Sence Approach to Web Usability. Indiana- polis, IN: New Riders Publishing.

LAMPING J.; RAO, R. The hyperbolic browser: a focus + context technique for visualizing large hierarchies. Journal of Visual Languages and Computing 7(1): 33-55, 1996.

LEVINE, L. et al. Software Process Automation: Interviews, Survey and Workshop Results. (CMU/SEI-97-TR-008) Pittsburgh: PA: Software Engineering Inst., Carnegie Mellon Univ., October 2006.

LIMA et al., Uma Interface Gráfica Para Visualização De Dados No Formato Su, II Simpósio Brasileiro de Geofísica, 2006.

LINTERN, R., MICHAUD, J., STOREY, M., et al. (2003) "Plugging-in Visualization: Experiences Integrating a Visualization Tool with Eclipse". In: ACM Symposium on Software Visualization (SoftVis), p. 47-ff, San Diego, California, June.

LUNARDI, et al. Visualização dos resultados do Yahoo em nuvens de texto: uma aplicação construída a partir de web services. InfoDesign: Revista Brasileira de Design da Informação, 2008.

LUZZARDI, P. R. G. Critérios de Avaliação de Técnicas de Visualização de Informações Hierárquicas. Tese (Doutorado em Ciência da Computação) - Programa de Pós-Graduação em Computação - Universidade Federal do Rio Grande do Sul, Porto Alegre, 2003.

MACKINLAY, J.D.; ROBERTSON, G.G.; CARD, S K. The Perspective Wall: Detail and Context Smoothly Integrated. Proceedings of the ACM CHI'91 Human Factors in Computing Systems Conference, pp. 173-179, 1991.

MAIA, C.L.B. Uma abordagem integrada, interativa e multi-objetiva para os problemas de seleção, priorização e alocação de casos de teste. Universidade Estadual do Ceará - Centro de Ciência e Tecnologia. Dissertação do Mestrado Acadêmico em Ciência da Computação. Fortaleza, 2011.

MAYHEW, D., Principles and guidelines in software user interface design. New Jersey: Prentice Hall, 1992.

MIETTINEN, K.: Nonlinear Multiobjective Optimization. Kluwer Academic Publishers, Boston (1999).

MOLICH, R., and Nielsen, J. (1990). Improving a human-computer dialogue, Communications of the ACM 33, 3 (March), 338-348.

MOLINARI, Leonardo, Testes Funcionais de Software. Florianópolis: Visual Books, 2008.

MOLINARI, Leonardo, Testes de Software. Florianópolis: Visual Books, 2014.

MORAES, R. Recursos tecnológicos e intervenções pedagógicas significativas no ensino da visualização de dados. Florianópolis: Universidade Federal de Santa Catarina. Dissertação (Engenharia de produção), 2000.

MOREIRA, R. Plataforma em hardware reconfigurável para o ensino e pesquisa em laboratório de sistemas digitais à distância. Mestrado, Departamento de Comunicações, Faculdade de Engenharia Elétrica e de Computação, Universidade Estadual de Campinas (UNICAMP), 2013.

MYERS, A. et al. A framework and methodology for studying the causes of software errors in programming systems. Journal of Visual Languages and Computing, v.16 n.1-2, p.41-84, 2004.

NASCIMENTO, J. Modelagem e Simulação Dinâmica AMESim - Ambiente para Protótipos Virtuais. Unicamp, 2014.

NASCIMENTO, V. Modelagem matemática da secage convectiva. Universidade Estadual de Campinas. 2015.

NIELSEN, Jakob. Ten Usability Heuristics. 2001.

NIELSEN, J. (2000). Why You Only Need to Test with 5 Users. Jakob Nielsen's Alertbox. Retrie- ved September 3, 2011, from http://www.useit.com/alertbox/20000319.html.

NIELSEN, J. (2003). Usability 101: Introduction to Usability. Jakob Nielsen's Alertbox. Retrieved May 21, 2011, from http://www.useit.com/alertbox/20030825.html.

NIELSEN, J. (2012). Mobile Usability. Jakob Nielsen's Alertbox. Retrieved May 21, 2011, from http://www.useit.com/alertbox/mobile-usability.html.

O'REILLY, T., What is Web 2.0, (2006). http://www.oreilly.com/go/web2.

Pearrow M.Web Site Usability (2000) (Charles River Media, Rockland, MA).

PEARSON, W. Redes de Computadores e a Internet, 3a Ed. São Paulo, 2006.

PIERRI, M. Gerência de Riscos em Projetos de Software: Baseada nos Modelos de Processos de Referência PMBOK, CMMI, MPS.BR, TenStep e ISSO 12207. Rio de Janeiro: Ciência Moderna, 2013.

PREECE, J., Rogers, Y., Sharp, H. Design de Interação. Além da Interação HomemComputador. Bookman, 2013.

PRESSMAN, S. Engenharia de Software. Mc.Gaw-Hill. 1997.

PRESSMAN, S. Engenharia de Software. Mc.Gaw-Hill. 2001.

PRESSMAN, S. et al. Engenharia de Software. Oitava edição. Bookman, 2016.

RIDDEL, C. Studies on the pathogenesis of tibial dyschondroplasia. II. Growth rate of long bones. 1980.

ROBINSON, Pauline. ESP today: a practitioner's guide. New York: Prentice Hall, 1982.

RODRIGUES, A. Visualização de dados na construção infográfica: abordagem sobre um objeto em mutação. Pós-Graduação em Jornalismo e Convergência Midiática da Faculdade Social da Bahia (FSBA), 2010.

ROSEMBERG, C. Turning Usability Testing Data into Action without Going Insane. Designer Toptal, 2015.

ROTHERMEL, Kurt. (2001) “Architecture and Algorithms for a Distributed Reputation System”. Springer-Verlag, Berlin, Heidelberg.

SCHILLING, D., Revelle, C., Cohon, J.: An approach to the display and analysis of multiobjective problems. Socio-Economic Planning Sciences 17, 57–63 (1983).

SETZER, V. W. Data, Information, Knowledge and Competency, 1999.

SHARMA, C. et al. A Survey on Software Testing techniques using Genetic Algorithm, IJCSI (International Journal of Computer Science Issues), 2013. J. C. Maldonado. Critérios Potenciais Usos: Uma Contribuição ao Teste Estrutural de Software. PhD thesis, DCA/FEE/UNICAMP, Campinas, SP, July 1991.

SHNEIDERMAN, B. “Designing the User Interface: Strategies for Effective Human-Computer Interaction”, 2a Ed., Addison-Weslley, 1994.

SILVA et al., SPAC: Ferramenta para Processamento e Análise de Dados Científicos no Processo de Validação de Software em Aplicações Espaciais. Divisão de Sistemas de Solo Instituto Nacional de Pesquisas Espaciais (INPE), 2013.

SILVERMAN, J., Steuer, R.E., Whisman, A.: Computer graphics at the multicriterion computer/user interface. In: Haimes, Y., Chankong, V. (eds.) Decision Making with Multiple Objectives, vol. 242, pp. 201–213. Springer, Berlin (1985).

S. OHARA; F. Tsunoda; H. Maezawa; M. Hui; T. Wang; P.C.Y. Sheu; R. Paul (2014). Object testing in ITEE.

SOMMERVILLE, Ian. Engenharia de Software, 9a Edição. Pearson Education, 2011.

SOUZA, M.J.F. A hybrid heuristic algoritm for the open-pit-mining operational planinig problem. EJOR, 2010.

SUCUPIRA, G. PET-SI: Em Busca de Formação Diferenciada Para Discentes de SI através da Fábrica de Software Baseada em Métodos Ágeis. Universidade Federal Rural do Rio de Janeiro. Programa PET Sistemas de Informação, 2010.

TAKAHASHI, R.H.C (et al.) (Eds.). Evolutionary Multi- Criterion Optimization. 6th International Conference, EMO 2011, Ouro Preto, Brazil, April 5-8, 2011, Proceedings.

TALLAM, S., GUPTA, N. A Concept Analysis Inspired Greedy Algorithm for Test Suite Minimization. In: Proceedings of 6th ACM SIGPLAN-SIGSOFT workshop on Program analysis for software tools and engineering, p. 35-42, 2006.

TEUNISSEN, W. A preliminary investigation of the impact of open source software on telecommunication software development. In: Proceedings of the Southern African Telecommunication Networks and Applications Conference. 2002.

TIERNEY P.J. The FOOM Method - Modeling Software Product Line in Industrial Settings. Proceedings of the International Conference on Software Engineering Research and Practice, Las Vegas, USA. 1990.

TUFTE, Edward. Envisioning Information. Graphics Press. Cheshire, Connecticut, 2001.

TUSAR, T. et al. Visualization of Pareto Front Approximations in Evolutionary Multiobjective Optimization: A Critical Review and the Prosection Method. IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION, VOL. 19, 2015.

VOLPATO, Elisa. Quantos participantes chamar para um teste de usabilidade?. TESTR, São Paulo, 2005.

WAINER, J. et al. Um sistema crítico e coletor de design rationale integrados em um ambiente para análise e projeto orientados a objetos. In: Anais do Concurso de Trabalhos de Iniciação Científica, Sociedade Brasileira de Computação, Curitiba, Paraná, Julho, 2001.

WALLENIUS (1988). A Pareto race. Naval Research Logistics, vol. 35, 615-623. Steuer, R. An interactive multiple objective linear programming procedure. TIMS Studies in the Management Sciences, vol. 6, 225-239.

WALLIS, K.M. "Engineering Design Research?"in Jacques, R. and J.A. Powell (Eds.), Design: Science: Method, Proceedings of the 1980 Design Research Society, Westbury House, 1987.

WARE, M. Measuring the Effects ofSoftware Aspectization', CD-Rom Proceedings of the 1stWorkshop on Aspect Reverse Engineering (WARE 2004).. Delft, The Netherlands. Novembro, 2004.

WEISSTEIN, E. W. Root-Mean-Square. Wolfram Math World, 2010.

WINOGRAD, T. Understanding Computers and Cognition. Norwood, Ablex Publishers Company, 2003.

YOO, S., HARMAN, M. Pareto Efficient Multi-Objective Test Case Selection. In: Proceedings of International Symposium on Software Testing and Analysis (ISSTA'07), p. 140-150, 2007.