



DOUGLAS MOREIRA CABRAL PAES

**ANÁLISE DE PROBLEMAS DE
FUNCIONALIDADE E USABILIDADE NO
PROCESSO DE DESENVOLVIMENTO DO
SOFTWARE LEITOR DE TELAS LIVRE NVDA**

LAVRAS – MG

2014

DOUGLAS MOREIRA CABRAL PAES

**ANÁLISE DE PROBLEMAS DE FUNCIONALIDADE E USABILIDADE
NO PROCESSO DE DESENVOLVIMENTO DO SOFTWARE LEITOR
DE TELAS LIVRE NVDA**

Monografia de graduação apresentada ao Departamento de Ciência da Computação da Universidade Federal de Lavras como parte das exigências do curso de Sistemas de Informação para a obtenção do título de Bacharel em Sistemas de Informação.

Orientador

Prof. Dr. André Pimenta Freire

Lavras – MG

2014

DOUGLAS MOREIRA CABRAL PAES

**ANÁLISE DE PROBLEMAS DE
FUNCIONALIDADE E USABILIDADE NO
PROCESSO DE DESENVOLVIMENTO DO
SOFTWARE LEITOR DE TELAS LIVRE NVDA.**

Trabalho de Conclusão de Curso de
Graduação apresentado ao Colegiado do
Curso de Bacharelado em Sistemas de
Informação, para obtenção do título de
Bacharel.

APROVADA em 27 de novembro de 2014.

Dr. José Monserrat Neto

Bel. Thiago Nascimento Oliveira



Dr. André Pimenta Freire (Orientador)

**LAVRAS-MG
Novembro/2014**

Ao meu pai Martinho

À minha mãe Cida

Ao meu irmão Marco Aurélio

Ao meu sobrinho Bento, que está por vir

Aos meus avós (in memoriam)

A todos os meus amigos, tios e primos

DEDICO

AGRADECIMENTOS

Agradeço primeiramente a Deus, pelo dom da vida, pela fé e por sempre guiar e iluminar firmemente meus passos. Sem Ele, nada disso seria possível.

Agradeço também aos meus pais Martinho e Cida, pelo amor incondicional, pelas orações e por me ensinarem os autênticos valores da vida.

Ao meu irmão Marco Aurélio, pelo companheirismo, confiança e exemplo.

Aos meus avós (*in memoriam*), tios e primos por todo o carinho e boas energias transmitidas.

Aos meus amigos de Porto Firme/MG, pelos momentos de alegrias, festejos e conselhos valiosos.

Aos amigos de Januária/MG, pelos inesquecíveis momentos vividos no CEFET.

A todos os amigos de Lavras/MG, especialmente ao pessoal do LEMAF, república e da turma de SI 2009/2 por estarem juntos e me apoiarem sempre.

À Universidade Federal de Lavras (UFLA), ao Departamento de Ciência da Computação (DCC) e aos meus professores, pelos conhecimentos repassados e pela contribuição para que esse trabalho fosse concluído com êxito, em especial ao meu orientador André Pimenta Freire, pela seriedade em seu trabalho e por sua paciência e sabedoria durante a orientação.

Enfim, meus sinceros agradecimentos a todos que contribuíram direta ou indiretamente para que esse trabalho fosse finalizado com sucesso.

*“Mas pra quem tem pensamento forte
O impossível é só questão de opinião”*

(Charlie Brown Jr.)

“Pra quem tem fé, a vida nunca tem fim”

(O Rappa)

RESUMO

Este trabalho apresenta uma análise investigativa do processo de desenvolvimento do *software* leitor de telas livre NVDA a fim de identificar quais problemas de funcionalidade e usabilidade apresentam-se como fatores críticos para o sucesso e crescimento do uso desse *software* pelos usuários com deficiência visual. Através da verificação, classificação e estudo de um conjunto significativo de notificações de erros ou pedidos de melhoria colhidos na plataforma pública de sua comunidade global de usuários, foi possível elencar e apresentar quais são esses fatores críticos de usabilidade primordiais para o sucesso e expansão do uso desse *software*. Questões de usabilidade relacionadas com a flexibilidade de uso, visualização de status do sistema e consistência de ações foram as que mais apresentaram problemas e as que mais tiveram impacto na utilização dos usuários. Apesar da percepção da baixa prioridade de problemas de usabilidade, não foram encontradas diferenças significativas no tratamento de problemas de usabilidade e funcionalidade no projeto analisado. A apresentação destes resultados contribui para compreender os gargalos de usabilidades existentes nesse tipo de *software*. Assim, tais constatações poderão fomentar o desenvolvimento de um *software* mais satisfatório para os usuários desse recurso de Tecnologia Assistiva, permitindo assim alavancar o crescimento dessa comunidade global de usuários.

Palavras-chave: Tecnologia Assistiva. Usabilidade em *Software* Livre. Processo de Desenvolvimento. NVDA.

ABSTRACT

This paper presents an investigative analysis of the software development process of the free screen reader NVDA to identify which functionality and usability problems are presented as critical factors for the success and growth of the use of this software for visually impaired users. By means of the verification, classification and study of a significant number of reports of errors or enhancement requests collected in its global community of users public platform, it was possible to list and display what are the critical factors of usability that are paramount for the success and expansion of the use of this software. Usability issues related to the flexibility of use, viewing system status and consistency of actions were those that presented more problems and that impacted the most the use of the system. Although the perception of low-priority for usability problems, no significant differences were found in the treatment of problems of usability and functionality of the project analyzed. The presentation of these results helps to understand the bottlenecks existing in this type of software usability. Thus, these findings may encourage the development of more satisfactory software for users of Assistive Technology, allowing to leverage the growth of this global community of users.

Keywords: Assistive Technology. FLOSS Usability. Development Process. NVDA.

SUMÁRIO

1.	INTRODUÇÃO.....	14
1.1.	Objetivos.....	16
1.2.	Organização da monografia.....	17
2.	REFERENCIAL TEÓRICO.....	19
2.1.	Processo de Desenvolvimento de <i>Software Livre</i>	19
2.2.	Usabilidade, Acessibilidade, Heurísticas de Jakob Nielsen e Tecnologia Assistiva.....	23
2.2.1.	Usabilidade e Acessibilidade.....	23
2.2.2.	Heurísticas de Jakob Nielsen.....	24
2.2.3.	Tecnologia Assistiva.....	28
2.3.	O Projeto NVDA.....	29
2.3.1.	O surgimento do NVDA.....	29
2.3.2.	Funcionamento do <i>software</i>	31
2.3.3.	Outras características do <i>software</i>	32
2.4.	Ferramentas e técnicas utilizadas no Processo de Desenvolvimento de <i>Software Livre</i>	33
2.4.1.	A plataforma e processo de reporte de <i>bugs</i> do sistema NVDA.....	34
2.4.1.1.	Sobre o sistema de <i>trac</i>	34
2.4.1.2.	Cadastrando um novo <i>ticket</i>	35
2.4.1.3.	Processo de análise e correção de um <i>ticket</i>	41
2.5.	Trabalhos relacionados.....	44
3.	MÉTODO.....	47
3.1.	Caracterização da pesquisa.....	47
3.2.	Procedimentos metodológicos.....	48
3.2.1.	<i>Design</i>	48

3.2.2.	Coleta e organização dos dados	52
3.2.3.	Métodos de análise de dados.....	54
4.	RESULTADOS E DISCUSSÃO	58
4.1.	Dados resultantes da análise	58
4.1.1.	Funcionalidade vs. usabilidade	58
4.1.2.	Usuários que reportaram os <i>tickets</i>	62
4.1.3.	Caracterização dos problemas de usabilidade em relação às Heurísticas de Nielsen.....	66
4.1.4.	Tratamento dos casos pelos desenvolvedores	80
4.2.	Dificuldades encontradas	81
5.	CONCLUSÃO	83
5.1.	Trabalhos futuros	85
	REFERÊNCIAS BIBLIOGRÁFICAS.....	86
	APÊNDICE A: Planilha do <i>data-set</i> de <i>tickets</i> utilizados para análise.....	94

LISTA DE FIGURAS

Figura 2.1 Formulário de cadastro de usuário na plataforma “NVDA Community”	36
Figura 2.2 Tela do menu “Issues” contendo as diretrizes para cadastrar um <i>ticket</i> no sistema de rastreamento de <i>bugs</i>	37
Figura 2.3 Página de busca por <i>tickets</i> já existentes no sistema	38
Figura 2.4 Formulário de preenchimento das informações do <i>ticket</i> a ser cadastrado	39
Figura 2.5 Exemplo de um <i>ticket</i> cadastrado no sistema de rastreamento de bugs/melhorias do NVDA	41
Figura 3.1 Exemplo de um <i>ticket</i> com sugestão de nova funcionalidade reportada por um usuário.	51
Figura 4.1 <i>Ticket</i> reportando um problema de usabilidade	60
Figura 4.2 <i>Ticket</i> reportando um problema de funcionalidade	61
Figura 4.3 Exemplo de um <i>ticket</i> de usabilidade que viola as condições descritas pela heurística 1 de Jakob Nielsen	69
Figura 4.4 Exemplo de um <i>ticket</i> de usabilidade que viola as condições descritas pela heurística 2 de Jakob Nielsen	70
Figura 4.5 Exemplo de um <i>ticket</i> de usabilidade que viola as condições descritas pela heurística 3 de Jakob Nielsen	71
Figura 4.6 Exemplo de um <i>ticket</i> de usabilidade que viola as condições descritas pela heurística 4 de Jakob Nielsen	72

Figura 4.7 Exemplo de um <i>ticket</i> de usabilidade que viola as condições descritas pela heurística 5 de Jakob Nielsen	73
Figura 4.8 Exemplo de um <i>ticket</i> de usabilidade que viola as condições descritas pela heurística 6 de Jakob Nielsen	74
Figura 4.9 Exemplo de um <i>ticket</i> de usabilidade que viola as condições descritas pela heurística 7 de Jakob Nielsen	76
Figura 4.10 Exemplo de um <i>ticket</i> de usabilidade que viola as condições descritas pela heurística 8 de Jakob Nielsen	77
Figura 4.11 Exemplo de um <i>ticket</i> de usabilidade que viola as condições descritas pela heurística 9 de Jakob Nielsen	78
Figura 4.12 Exemplo de um <i>ticket</i> de usabilidade que viola as condições descritas pela heurística 10 de Jakob Nielsen	79

LISTA DE QUADROS

Quadro 1 <i>Status</i> em que um <i>ticket</i> poderá tramitar durante o processo de correção.....	43
Quadro 2 Exemplo de uma a classificação das heurísticas de Nielsen para os <i>tickets</i> de “Usabilidade”	57
Quadro 3 Número de relatores distintos que compuseram o <i>data-set</i> de <i>tickets</i> analisados.....	62
Quadro 4 N° de desenvolvedores que também atuaram como relatores de <i>tickets</i> que formaram o <i>data-set</i>	63

LISTA DE TABELAS

Tabela 1 Categoria dos <i>tickets</i> analisados (em número bruto e %)	59
Tabela 2 Usuários finais que mais reportaram <i>tickets</i> no conjunto do <i>data-set</i> ..	65
Tabela 3 Distribuição das heurísticas quanto aos problemas apresentados nos <i>tickets</i> classificados como de usabilidade no <i>data-set</i>	66
Tabela 4 Respostas para as perguntas efetuadas após a classificação das heurísticas violadas por cada <i>ticket</i>	68
Tabela 5 Relação dos tickets de usabilidade e funcionalidade quanto ao motivo de fechamento dos mesmos	80

1. INTRODUÇÃO

A inclusão de pessoas com deficiência na sociedade em que vivem tem se mostrado um grande desafio em todo o mundo. No Brasil não é diferente. Prover acesso à educação, trabalho, entretenimento, entre outros, é tão importante que seus respectivos acessos, sem barreiras, se tornaram um direito para pessoas com deficiência. Viver de uma forma autônoma, sem barreiras e com as mesmas qualidades de vida de uma pessoa sem deficiência é um dos maiores anseios das pessoas que tem algum tipo de deficiência.

Em particular, uma das formas que podem contribuir para remoção de barreiras e provisão de recursos na busca por uma melhor qualidade de vida de deficientes consiste na utilização de recursos de Tecnologia Assistiva. Cook e Polgar (1995) definem essas tecnologias como uma ampla gama de equipamentos, serviços, estratégias e práticas, concebidas e aplicadas para minorar os problemas encontrados pelos indivíduos com deficiências.

Diferentes recursos de Tecnologia Assistiva podem ser utilizados por pessoas com diferentes tipos de deficiência, e incluem recursos que variam desde simples produtos tecnológicos, como uma bengala, regletes utilizados por cegos para escrita em *Braille*, a produtos tecnológicos computacionais, como sistemas sintetizadores de voz para o conteúdo da tela para cegos, ampliadores de tela para pessoas com visão limitada, teclados e dispositivos alternativos para pessoas com deficiência motora, dentre inúmeros outros recursos.

Porém, um dos grandes desafios ao acesso de recursos de Tecnologia Assistiva no Brasil deve-se ao seu alto custo de aquisição. Muitas dessas tecnologias são importadas e por este motivo, chegam com um custo que impede a sua aquisição por grande parte das pessoas necessitadas.

Ao mesmo tempo em que estas barreiras impedem a disseminação do uso de tais tecnologias, alternativas possibilitam contornar os problemas citados. Dentre as quais, podemos destacar o crescente desenvolvimento de diversas ferramentas como *software* livre. Para casos de usuários cegos que utilizam *softwares* leitores de telas, podemos citar o desenvolvimento de tecnologias como o *NonVisual Desktop Access – NVDA* (NVDA, 2013) para objeto de estudo.

O NVDA é um leitor de telas gratuito, disponibilizado para computadores que executam o sistema operacional *Windows*. O *software* NVDA surgiu de um projeto *open source* iniciado no ano 2006, em South East Queensland, Austrália, por Michael Curran e que, posteriormente ganhou o reforço de James Teh à equipe de desenvolvimento do projeto. Juntos, esses dois homens (totalmente cegos) fundaram uma organização, sem fins lucrativos, denominada *NV Access*. Através de doações e aportes financeiros de diferentes instituições, foi possível o desenvolvimento do NVDA pela *NV Access* (NVDA, 2013).

O uso de recursos de Tecnologia Assistiva livres no Brasil (como o NVDA) tem contribuído amplamente para impulsionar o uso de computadores por pessoas com deficiência. Em particular, pode ser observado o maior uso de tais tecnologias no Brasil que em outros países com níveis de renda mais elevada. Uma pesquisa realizada pela *Web Accessibility in Mind* (WebAIM, 2012) com 1782 participantes, em sua maioria pessoas com deficiências visuais e residentes na Europa e América do Norte, revelou que 49% dos usuários utilizavam o *software* leitor de telas comercial *JAWS*, da *Freedom Scientific*, como seu principal leitor de telas, enquanto somente 18% responderam que utilizam o NVDA. Resultados preliminares de uma pesquisa do escritório brasileiro do *World Wide Web Consortium* (W3C Brasil, 2013) com 157

usuários com deficiência visual revelou que a proporção de usuários que participaram da pesquisa que utilizavam o NVDA como principal leitor de telas era de 34%, atrás apenas do *JAWS*, utilizado por 41%. Esses números mostram que a utilização do leitor de telas livre no Brasil é significativamente maior que na Europa e América do Norte, demonstrando a relevância de se impulsionar o desenvolvimento de tais sistemas no Brasil.

Focando no contexto brasileiro, existe pouco conhecimento sobre a dinâmica de projetos de recursos de Tecnologia Assistiva livres que tem alcançado elevados números de usuários e uma ampla utilização pelos mesmos. Também há pouco conhecimento sobre os fatores desses projetos que levaram à sua maior utilização, em particular fatores relacionados ao processo de desenvolvimento e à forma de interação entre usuários com deficiência e desenvolvedores de tais tecnologias. Essa dificuldade de conhecimento se dá, de acordo com Crowston et al., (2005) principalmente devido a peculiaridades nos processos de desenvolvimento de *softwares* de código livre, em relação aos processos tradicionais de desenvolvimento.

Levando em consideração os aspectos citados, surge a necessidade de conhecimento de processos e boas práticas, com o intuito de melhorar o desenvolvimento desses recursos de Tecnologia Assistiva livres.

1.1. Objetivos

De acordo com o contexto apresentado, a presente pesquisa tem como objetivo geral realizar uma análise investigativa sobre o processo de desenvolvimento do *software* leitor de telas livre NVDA, a fim de identificar quais problemas de funcionalidade e usabilidade apresentam-se como fatores críticos para o sucesso e crescimento do uso desse *software* pelos usuários com

deficiência, contribuindo assim para o conhecimento em como melhorar e impulsionar o uso de tais recursos de *software* livre como Tecnologia Assistiva.

A escolha pelo NVDA foi pelo fato de ser um *software* livre e também por ser o leitor de telas livre mais utilizado no Brasil (W3C Brasil, 2013).

Para o alcance do objetivo geral, os objetivos específicos que o compõe o estudo são:

- Efetuar um estudo sobre processo de desenvolvimento de *softwares* livres;
- Efetuar um estudo sobre questões de usabilidade em desenvolvimento de *software* livre;
- Efetuar um estudo sobre o processo de desenvolvimento do *software* NVDA;
- Efetuar uma análise de questões de funcionalidade e usabilidade no processo de desenvolvimento do NVDA.

1.2. Organização da monografia

Esta monografia está organizada da seguinte forma:

No Capítulo 2 é apresentado o referencial teórico sobre o tema abordado. Neste capítulo é realizado um estudo do estado atual da arte e uma breve revisão da literatura sobre processos de desenvolvimento de *software* livre, Usabilidade e Acessibilidade, Heurísticas de Jakob Nielsen, Tecnologia Assistiva, além da caracterização do objeto de estudo – o Sistema NVDA. Também são apresentadas algumas das técnicas e ferramentas utilizadas no desenvolvimento de projetos de *software* livre, e ainda os trabalhos relacionados com a temática de usabilidade em processos para esse tipo de *software*.

No Capítulo 3 é descrito a metodologia utilizada nesse estudo, apresentando uma caracterização da pesquisa e os procedimentos metodológicos adotados para realização de tal pesquisa.

No Capítulo 4 são descritos os resultados alcançados após a aplicação do estudo e discussões sobre a execução do trabalho.

Por fim, no Capítulo 5 são apresentadas as conclusões que puderam ser obtidas após a análise dos resultados e finalização do trabalho.

2. REFERENCIAL TEÓRICO

Este capítulo apresenta uma revisão da literatura dos principais conceitos que tangem o tema explorado. Primeiramente são lembrados conceitos e práticas utilizadas no processo de desenvolvimento de *softwares* livres. Em seguida, são definidos conceitos de Usabilidade, Acessibilidade, Heurísticas de Jakob Nielsen e Tecnologia Assistiva. Há também uma breve abordagem acerca do projeto NVDA. Por fim, são apresentados os principais trabalhos relacionados a questões de usabilidade em processos de *software* livre.

2.1. Processo de Desenvolvimento de *Software* Livre

Antes de compreendermos como se dá o processo de desenvolvimento de um *software* livre, devemos primeiramente definir alguns conceitos básicos. Um deles é o Processo de *Software*. Sommerville (2010, p. 28, tradução nossa) define processo de *software* da seguinte forma:

“Um processo de *software* é um conjunto de atividades relacionadas, que conduzem à produção de um produto de *software*. Essas atividades podem envolver o desenvolvimento de *software* a partir do zero em uma linguagem de programação padrão, como *Java* ou *C*. No entanto, aplicações de negócios não são necessariamente desenvolvidas desta forma. Novos *softwares* de negócio hoje em dia são muitas vezes desenvolvidos através da ampliação e modificação de sistemas existentes ou através da configuração e integração de *off-the-shelf software* ou sistema de componentes.”

Pressman (2011) deixa claro que “processo” no contexto de engenharia de *software* não é uma prescrição rígida de como desenvolver um *software*. Segundo ele, este conceito se apresenta como uma abordagem adaptável que permite que pessoas (equipe de *software*) realize o trabalho de selecionar um conjunto apropriado de tarefas e ações para o mesmo.

Ainda nos anos 70, Schwartz (1975) já definia que um processo de *software* é composto das seguintes fases: Especificação de Requisitos, Projeto de Sistema, Programação (Codificação) e Verificação e Integração (*Checkout*).

De uma forma simplista e resumida, podemos dizer que o processo de desenvolvimento de *software* envolve todo o conjunto de etapas ou fases para criação de um produto de *software* em um determinado escopo de projeto num determinado tempo. Essas características são vistas e julgadas como comuns nos mais diversificados processos de desenvolvimento de *software* atualmente.

Porém, algumas peculiaridades podem ser observadas quanto à natureza de desenvolvimento, modo de distribuição e utilização de um *software*. Uma delas é o *Software Livre*. Por “*Software Livre*” (Em inglês, *Free Software*) devemos entender aquele *software* que respeita a liberdade e senso de comunidade dos usuários.

A palavra “livre” em “*Software Livre*” refere-se a liberdade, não ao preço. Um *software* livre pode ser ou não comercializado. Porém, uma vez que você tenha este *software*, há quatro liberdades específicas para utilizá-lo: liberdade para executar o *software*; liberdade para copiá-lo e doá-lo a seus amigos; liberdade de modificá-lo da forma que desejar, por ter acesso integral ao seu código-fonte; e liberdade para distribuir versões melhoradas e contribuir com a comunidade de usuários deste *software*. Para essa distribuição, pode ser cobrada alguma taxa pelo ato físico de se transferir a cópia, ou simplesmente fornecer estas cópias gratuitamente (GNU, 2013).

Esse tipo de *software* também é conhecido como *software* de código-fonte aberto, ou simplesmente *software open source* (OSI, 1998).

Reis (2003) cita a definição de *software* livre utilizada pela *The Free Software Foundation* (2013) como sendo um *software* que é distribuído acompanhado de seu código-fonte, e que pode ser livremente modificado e

redistribuído, diferentemente de um *software* proprietário, que possui suas estruturas lógicas de código-fonte fechadas e não legíveis a programadores externos à empresa proprietária. Para um *software* proprietário é disponibilizado apenas o código binário para sua execução, através de compra da licença de uso ou pela distribuição gratuita de seu executável (Taurion, 2004).

Stallman (1999) explica que o *software* de código aberto existe desde os primórdios da informática. Entre os anos 60 e 70 o *software* não tinha valor comercial, pelo fato de que o foco de mercado àquela época era o *hardware*¹. E para cada *hardware* havia um *software* específico. A partir da metade dos anos 70, quando os *softwares* passaram a funcionar em mais de um computador, iniciou-se a comercialização de *software* pelas empresas e, conseqüentemente, o “fechamento” de seu código-fonte, tornando-os *softwares* proprietários.

O termo “*software* livre” começou a ser utilizado após essa época. Em 1983, Richard Stallman lançou o projeto GNU (GNU, 2013). Esse projeto culminou em um manifesto em prol da liberdade de distribuição do *software* juntamente com seu código-fonte (GNU, 2013). A partir desse movimento, o conceito de *software* livre difundiu-se pela comunidade de *software*.

A característica principal de um ambiente de desenvolvimento de *software* livre era a colaboração e, por isso, acesso ao código fonte dos programas era um requisito necessário. Foi criada uma cultura de repassar as melhorias feitas nos códigos aos demais programadores, que, por sua vez, tinham no reconhecimento dos outros a principal motivação (Machado e Pereira, 2006).

¹ Componentes físicos de um computador, como por exemplo, o processador, memória RAM, o disco rígido, etc.

Algumas das características principais observadas em um processo de desenvolvimento de um *software* livre é que o mesmo é desenvolvido de uma forma altamente distribuída, com apoio de ferramentas de listas de discussão, mensageiros eletrônicos (*Skype, IRC, Messenger*, etc.), repositórios de código-fonte e sistema de relatório de erros (*bugs*) utilizados para apoiar a comunicação e sincronização em práticas de trabalho (Gutwin et al., 2004). O texto é o meio primário de comunicação, bem como o objeto primário de interesse (mais especificamente, o código-fonte) (Terry et al., 2010).

Os membros de projetos de *software* livre são geralmente organizados com base em méritos individuais. E esses membros, considerados também como contribuintes, se apresentam muitas vezes de forma voluntária e escolhem tarefas com base em interesses pessoais (Mockus et al., 2000).

O caráter voluntário da comunidade de desenvolvimento levou a pesquisas consideráveis que examinaram as motivações para contribuição nos projetos de *software* livre (Hars e Ou, 2002; Lakhani e von Hippel, 2003). Os resultados mostraram que os membros dos projetos se motivam a participar para desenvolverem habilidades pessoais, encarar desafios intelectuais impostos pelas atividades, construir o capital social, melhorar a sua reputação e fazer parte de uma comunidade (Oreg e Nov, 2008).

A arquitetura nos projetos de desenvolvimento de *software* livre tende a ser bastante modular por natureza, com os desenvolvedores supervisionando todas as modificações para os módulos criados por eles próprios (Mockus et al., 2000). Esta modularidade ajuda a atrair novos desenvolvedores, uma vez que é mais fácil de iniciar qualquer contribuição em pequenas partes, independente do código de aplicação ser extenso ou ser uma aplicação monolítica (Baldwin e Clark, 2006).

2.2. Usabilidade, Acessibilidade, Heurísticas de Jakob Nielsen e Tecnologia Assistiva

Nesta seção são apresentados conceitos envolvendo Usabilidade de sistema, Acessibilidade, as Heurísticas de Jakob Nielsen e Tecnologia Assistiva.

2.2.1. Usabilidade e Acessibilidade

Desenvolvedores devem trabalhar para criar *softwares* que são fáceis e simples para as pessoas usarem. Termos como “familiaridade” e “facilidade no uso” muitas vezes indicam essas características. Porém o termo técnico geral para essas características é “usabilidade” (Petrie e Bevan, 2009).

Pela definição da *International Organization for Standardization (ISO)*, usabilidade é a medida pela qual um produto pode ser usado por usuários específicos para alcançar objetivos específicos com efetividade, eficiência e satisfação em um contexto de uso específico (ISO, 1998).

Neste mesmo padrão, o termo “efetividade” é definido como “a precisão e o nível de completude com os quais os usuários podem atingir objetivos específicos”; “eficiência” é definida como a “quantidade de recursos utilizados para cumprir os objetivos em relação à precisão e completude”; e “satisfação” é definida como “ausência de desconforto e impressões positivas pelo uso do produto”.

Petrie and Bevan (2009) citam que a usabilidade não possui termos absolutos que a define. É relativa aos usuários, objetivos e contextos de uso, que são apropriadas para o conjunto específico de circunstâncias.

Nielsen (1993) define que a usabilidade é um atributo de qualidade que avalia quão fáceis interfaces de usuário são para usar. O termo "usabilidade"

também se refere a métodos para melhorar o processo de *design* de interface. Ele cita que a usabilidade está apoiada por cinco critérios (componentes) de qualidade, descritos a seguir:

- **Facilidade de Aprendizado:** diz respeito ao quanto é fácil para os usuários a realização tarefas básicas na primeira vez que eles utilizam a interface.
- **Eficiência no Uso:** uma vez que os usuários aprenderam a interface (*design*), há rapidez na realização das tarefas?
- **Memorização:** quando os usuários retornam ao projeto após um período sem usá-lo, como facilmente podem eles restabelecer a proficiência?
- **Segurança no Uso (erros):** Quantos erros os usuários encontram, quão grave são esses erros e como facilmente podem eles se recuperarem desses erros?
- **Satisfação do Usuário:** Quanto agradável é usar o *design*?

Seguindo a mesma linha, a Parte 171 da ISO 9241 (ISO, 2008) define acessibilidade de *software* como sendo “a usabilidade de um produto, serviço, ambiente ou funcionalidade por pessoas com os mais variados tipos de habilidades e capacidades”. Esse conceito estende a definição de usabilidade para pessoas com os mais variados tipos de capacidades, em particular aquelas que têm alguma deficiência.

2.2.2. Heurísticas de Jakob Nielsen

Um dos métodos de inspeção de usabilidade que pode ser aplicado ao processo de desenvolvimento de *software*, foi definido e disponibilizado por Nielsen (1994b). Jakob Nielsen é tido como uma das pessoas mais influentes e engajadas no ramo da interação humano-computador (IHC) e seus trabalhos

possuem ampla aceitação pela comunidade acadêmica. Esse método, proposto por ele, consiste em um pequeno conjunto de regras práticas (10 heurísticas) que visam auxiliar/inspecionar o desenvolvimento de uma interface de usuário que se enquadre na sua definição de um sistema com alta “usabilidade”. Em outras palavras, essas heurísticas são regras gerais que visam auxiliar no desenvolvimento de uma interface de usuário que seja amigável e de fácil uso.

Tais heurísticas começaram a ser formuladas por Jakob Nielsen em colaboração com Rolf Molich em 1990 (Molich e Nielsen, 1990; Nielsen e Molich, 1990). Posteriormente foram refinadas e aprimoradas baseando-se em um estudo complementar sobre conjunto problemas de usabilidade de interfaces levantados por Nielsen (ver Nielsen, 1994a e Nielsen, 1994b).

Estas heurísticas estão alinhadas com os cinco critérios (componentes) de qualidade de uma interface estabelecidos por ele, já citado anteriormente neste trabalho.

A seguir a lista e uma breve descrição das 10 (dez) heurísticas definidas por Nielsen (1994b):

H1. Visibilidade do status do sistema:

O sistema deve sempre manter os usuários informados sobre o que está acontecendo, através de *feedback* apropriado em tempo razoável.

H2. Correspondência entre o sistema e o mundo real:

O sistema deve falar a linguagem dos usuários, com palavras, frases e conceitos familiares para o usuário, em vez de termos orientados ao sistema. Siga as convenções do mundo real, fazendo com que as informações apareçam em ordem natural e lógica.

H3. Controle e liberdade do usuário:

Usuários frequentemente escolhem funções do sistema por engano e irão precisar de uma “saída de emergência” claramente demarcada, auxiliando a mudar-se do estado indesejado sem ter que passar por um diálogo extenso. Forneça possibilidades de desfazer e refazer.

H4. Consistência e padrões:

Os usuários não devem ter que se perguntar se diferentes palavras, situações ou ações significam a mesma coisa. Siga as convenções de plataforma.

H5. Prevenção de erros:

Melhor do que boas mensagens de erro é um *design* cuidadoso que impeça que um problema ocorra. Eliminar as condições passíveis de erros ou verificá-los e apresentar aos usuários uma opção de confirmação antes de se executar a ação.

H6. Reconhecimento ao invés de recordação:

Minimizar a carga de memória do usuário, fazendo objetos, ações e opções sempre visíveis. O usuário não deve ter que lembrar informações de uma parte do diálogo para outra. Instruções de utilização do sistema devem estar visíveis ou facilmente recuperáveis sempre que apropriado.

H7. Flexibilidade e eficiência no uso:

Aceleradores – invisíveis aos usuários novatos – podem frequentemente tornar mais eficiente a interação para o usuário experiente de tal forma que o sistema atenda a ambos os usuários, inexperientes e experientes. Permita aos usuários adaptações para ações frequentes.

H8. Estética e design minimalista:

Os diálogos não devem conter informações que são irrelevantes ou raramente necessárias. Cada unidade extra de informação em um diálogo compete com as unidades relevantes de informação e diminui sua visibilidade relativa.

H9. Ajudar os usuários a reconhecer, diagnosticar e recuperar-se de erros:

As mensagens de erro devem ser expressas em linguagem simples (sem códigos), indicando com precisão o problema e construtivamente sugerindo uma solução.

H10. Ajuda e documentação:

Ainda que o sistema seja tão bom que possa parecer ser usado sem auxílio de documentação, pode ser necessário fornecer ajuda e documentação ao usuário. Qualquer informação deve ser fácil de pesquisar, focada na tarefa do usuário, listar passos concretos a serem realizados e não ser muito grande.

Neste trabalho, o conjunto das 10 heurísticas de usabilidade propostas por Nielsen foi utilizado como um dos critérios para levantamento e classificação dos principais fatores de usabilidade que envolvem o processo de desenvolvimento do NVDA. Mais detalhes serão apresentados no Capítulo 3 “Método”.

2.2.3. Tecnologia Assistiva

Para os autores Cook e Polgar (1995), Tecnologia Assistiva é definida como sendo uma ampla gama de equipamentos, serviços, estratégias e práticas concebidas e aplicadas para minorar os problemas encontrados pelos indivíduos com deficiências.

Recursos de Tecnologia Assistiva fornecem uma ajuda para que pessoas com deficiência possam superar barreiras de acesso a diversas tarefas do cotidiano, como também o acesso à educação, trabalho e entretenimento.

Bersch (2008) acrescenta que tais recursos propiciam à pessoa com deficiência uma maior independência, qualidade de vida e inclusão social por meio da ampliação de sua comunicação, mobilidade, controle de seu ambiente, habilidades de seu aprendizado e trabalho.

Já o Comitê de Ajudas Técnicas do Brasil (2009) define Tecnologia Assistiva (TA) como uma área do conhecimento, de característica interdisciplinar, que engloba produtos, recursos, metodologias, estratégias, práticas e serviços que objetivam promover a funcionalidade, relacionada à atividade e participação, de pessoas com deficiência, incapacidades ou mobilidade reduzida, visando sua autonomia, independência, qualidade de vida e inclusão social.

De um ponto de vista computacional, os recursos de Tecnologia Assistiva permitem que usuário com os mais diversos tipos de deficiência contornem os problemas ou barreiras encontradas no uso de computadores, como por exemplo, o acesso a conteúdos *web*.

Uma das categorias de recursos de Tecnologia Assistiva que permitem que usuários cegos tenham acesso aos computadores são os *softwares* leitores de

telas. Esses *softwares* “leem” o conteúdo exibido no monitor em voz alta, permitindo assim a compreensão por parte do usuário com deficiência visual.

Atualmente há vários *softwares* leitores de telas em ampla utilização, entre os quais podemos citar os *softwares* proprietários *JAWS (Job Access With Speech)*, da *Freedom Scientific*, e o *Virtual Vision*, da *MicroPower*; o sistema operacional *freeware DOSVOX*, do Núcleo de Computação Eletrônica da Universidade Federal do Rio de Janeiro (UFRJ); e os *softwares* livres *ORCA*, da *Accessibility Program Office of Sun Microsystems, Inc.* e o *NVDA (NonVisual Desktop Access)*.

Como objeto da pesquisa, o presente trabalho propõe uma análise investigativa do processo de desenvolvimento do *software* leitor de telas livre *NVDA*.

2.3. O Projeto NVDA

Nesta seção são apresentadas informações básicas do projeto *NVDA*, características de funcionamento desse leitor de telas livre e outras informações complementares inerentes a esse projeto.

2.3.1. O surgimento do NVDA

O *NVDA (NonVisual Desktop Access)* é *software* um leitor de telas gratuito, disponibilizado para computadores que executam o sistema operacional *Microsoft Windows*. O *NVDA* surgiu de um projeto *open source* iniciado no ano de 2006, em South East Queensland, Austrália.

O projeto foi idealizado por Michael Curran, um estudante australiano cego que à época cursava o segundo ano de bacharelado em Ciência da

Computação. Diante da cegueira, Michael se viu obrigado a adquirir um *software* leitor de telas comercial para uso pessoal, profissional e acadêmico. Apesar do leitor proporcionar um acesso relativamente bom aos computadores que Michael precisava usar, com o passar do tempo e convívio com pessoas em situação semelhante mundo afora, pelo menos três problemas ficaram claros para ele:

- O alto custo da licença que esse *software* comercial possui, impondo grandes restrições a pessoas que muitas vezes não possuem o capital necessário para aquisição da licença;
- Por melhores que fossem as soluções fornecidas pelo *software* assistivo em questão, o condicionamento dos usuários cegos às políticas e ideias das empresas se tornam um problema. Cada pessoa em particular, como usuário e também consumidor, sempre ficará insatisfeito com algum detalhe, isso porque essas soluções geralmente são designadas para atender a todos de modo genérico e não específico. Como a licença desses *softwares* na maioria das vezes é restritiva, torna-se impossível a cada usuário ou grupo de usuários adaptar o *software* em virtude de necessidades específicas.
- Há problemas em aspectos morais e éticos, não sendo justo que pessoas cegas tenham de providenciar sozinhas os meios necessários e arcar por si só com os custos dos recursos de Tecnologia Assistiva, para dispor de acesso às mesmas informações que as demais pessoas dispõem. Sabendo disso, muitos optam por adquirir cópias ilegais desses *softwares*, o que, por um lado ameniza os gastos financeiros, mas de outro, mantém a sujeição desses usuários às políticas do fabricante do *software*, além de ferir a licença outorgada da fabricante (Uliana, 2013).

Diante de tais problemas, Michael resolveu abandonar a faculdade de Ciência da Computação e empenhou-se no desenvolvimento de um projeto *open source* capaz de contornar esses e outros problemas, dando início então ao projeto NVDA.

Posteriormente esse projeto ganhou reforço na equipe de desenvolvimento do também estudante cego James Teh. Juntos, Teh e Curran fundaram uma organização, sem fins lucrativos, denominada *NV Access*. Através de doações e aportes financeiros de diferentes instituições, foi possível o desenvolvimento do NVDA (NVDA, 2013).

Atualmente o *software* NVDA está traduzido para mais de 43 idiomas e utilizado por pessoas em mais de 120 países (NVDA, 2013).

2.3.2. Funcionamento do *software*

Em relação ao seu funcionamento, o processo de leitura de tela do NVDA atua da seguinte forma: o *software* comunica o que está escrito na tela do computador, por meio de som (voz sintética) ou *Braille* à pessoa com deficiência visual de forma que ela possa interagir com o computador e seus aplicativos.

De uma forma geral, as características do *software* incluem:

- A execução a partir de um dispositivo de armazenamento *USB* ou qualquer outra mídia portátil, sem a necessidade de instalação do *software* no computador;
- Navegação na Internet, quando utilizado o *Mozilla Firefox*;
- Instalador com recursos de voz sintetizada, fácil de usar;
- Suporta a leitura de *e-mail*, quando utilizado o *Mozilla Thunderbird*;
- Suporte para *Microsoft Internet Explorer*;
- Suporte básico para *Microsoft Outlook Express/Windows mail*;

- Suporte básico para *Microsoft Word e Excel*;
- Suporte para aplicativos *Java* acessíveis;
- Suporte para *Adobe Reader*;
- Suporte para *IBM Lotus Symphony*;
- Suporte para o *Prompt* de comandos do *Windows*;
- Anúncio automático do texto onde o mouse estiver e indicação audível opcional da posição do *mouse*.

2.3.3. Outras características do *software*

O NVDA é suportado pelo sistema operacional *Windows*, nas edições 32 ou 64 bits, a partir das versões *Windows XP*. É escrito na linguagem de programação *Python*, construído em módulos e a maior parte de seu código pode ser facilmente expandida com a finalidade de suportar novos programas e/ou controles. *Buffers* virtuais podem ser escritos para permitir que o NVDA suporte documentos complexos. Um *buffer* virtual é uma representação textual do documento, que permite ao usuário explorar e ler o texto e campos que estão no documento. O NVDA sempre tenta tornar controles e elementos de um programa ou sistema operacional o mais acessível possível, de forma que o usuário possa buscar ativamente qualquer informação de que precise (NAPNE/RENAPNE, 2013).

O NVDA é coberto pela licença GNU GPL (*General Public License – versão 2*). Essa licença permite ao usuário compartilhar e modificar o *software* da maneira desejada, contando que, ao redistribuir o *software* modificado, distribua juntamente a sua licença e disponibilize o código-fonte a todos que o queiram (GNU, 2013).

O NVDA está disponível para *download* no site² da *NV Access*. Após o *download*, o usuário terá a opção de utilizar uma versão diretamente executável ou instalar por completo o *software* em seu computador.

2.4. Ferramentas e técnicas utilizadas no Processo de Desenvolvimento de *Software Livre*

Os projetos de *software* livre começaram a crescer de forma mais alavancada a partir dos anos 90 com o desenvolvimento e disseminação do uso da *Web*. Nesse sentido, tais projetos puderam se tornar cada vez mais distribuídos e, com isso, seu grupo de colaboradores passou a ser maior, mais engajado e globalizado. Para maiores detalhes sobre as definições envolvendo projetos de *software* livre, consultar Reis (2003).

Visto tal crescimento, para manter a estrutura de desenvolvimento desses projetos sempre organizada e também para assegurar sua eficiência e qualidade, várias ferramentas e técnicas de apoio ao processo de desenvolvimento foram criadas e passaram a ser amplamente utilizadas pela comunidade livre.

Conforme já destacado neste trabalho, Gutwin et al. (2004) coloca que tais projetos contam com o apoio dessas várias ferramentas no processo de desenvolvimento do *software* livre. Dentre as ferramentas destacadas por ele estão as listas de discussão de *e-mail*, mensageiros eletrônicos e de videoconferência (*Skype*³, *IRC*, *Messengers*, etc), repositórios de código-fonte (por exemplo, o *GIT*⁴ e o *Tortoise SVN*⁵) e sistema de relatório de erros, utilizados para apoiar a comunicação e sincronização em práticas de trabalho.

² Disponível online em <http://www.nvaccess.org.br/download/>

³ Disponível online em <http://www.skype.com/>

⁴ Disponível online em <http://git-scm.com/>

Neste trabalho, focaremos na ferramenta de apoio para o reporte e correção de erros (*bugs*) e propostas de melhorias colocadas pelos membros da comunidade de usuários do sistema NVDA. Essa ferramenta, pública, nos fornecerá a base das amostras que serão analisadas nesse estudo investigativo.

2.4.1. A plataforma e processo de reporte de *bugs* do sistema NVDA

A organização *NV Access*, por meio de seu sítio eletrônico oficial, disponibiliza publicamente, uma plataforma para que sua comunidade global de usuários e desenvolvedores obtenham informações/notícias sobre o projeto NVDA (por exemplo, notícias sobre novas versões do *software*, novidades, etc.) e também permite que a comunidade possa interagir entre si.

Essa plataforma, denominada “*NVDA Community*”⁶ permite à comunidade, ainda, o acesso a artigos, tutoriais e outros materiais que envolvam o uso/desenvolvimento do projeto e principalmente, fornece acesso ao sistema de rastreamento e gerenciamento de *bugs*/melhorias/recursos do NVDA. A seguir será detalhado o funcionamento do sistema de rastreamento.

2.4.1.1. Sobre o sistema de *trac*

O sistema de rastreamento propriamente dito, também chamado de “*trac*”, consiste em um sistema de apoio à gestão de projetos de *software*, mais especificamente de relatos de questões (denominadas pela plataforma de “*issues*”) onde os usuários e até mesmo os desenvolvedores podem submeter relatos (denominados “*tickets*”) de *bugs* (erros/defeitos), melhorias ou pedidos de inserção de novos recursos ou funcionalidades ao *software* NVDA.

⁵ Disponível online em <http://tortoisesvn.net/>

⁶ Disponível online em <http://community.nvda-project.org/>

Tal sistema *trac* é distribuído abertamente pela *Edgewell Software*⁷ e disponibilizado com o nome comercial *Trac Integrated SCM & Project Management* (Trac, 2014). É mantido sob os termos da licença *BSD* (OSI, 2014).

Esse sistema de rastreamento de *issues* atua como uma importante ferramenta de interação entre desenvolvedores e usuários do *software* livre NVDA. A partir dele podemos colher informações importantes sobre como se dá o processo de interação entre a comunidade e ainda obter informações sobre questões chave de usabilidade do sistema. Nesse sentido, focamos o estudo na análise dos dados provenientes do sistema de rastreamento a fim de investigar quais critérios de usabilidade são mais relevantes/prioritários em relação à utilização e expansão do uso do *software* NVDA.

2.4.1.2. Cadastrando um novo *ticket*

Para que o usuário ou desenvolvedor possa realizar o cadastramento de um *ticket* (pedido de correção de um *bug* ou inclusão de novos recursos ao sistema) é necessário estar cadastrado na plataforma “*NVDA Community*” e com autenticação efetuada.

O cadastro nessa plataforma é rápido e simples. O usuário deverá encontrar o *link* “*Register*” para que o registro seja iniciado. Esse registro é realizado através do preenchimento de formulário próprio de cadastro na plataforma. A Figura 2.1 apresenta esse formulário.

⁷ Disponível online em <http://www.edgewall.org/>

NVDA
Donate ▶

Home | Contact | Guides | Issues | Test | Translate | Development

Register an account

Required

Username:

Password:

Confirm Password:

Email:

The email address is required for Trac to send you a verification token.
Entering your email address will also enable you to reset your password if you ever forget it.

For spam prevention, please type the name of the company which develops NVDA. You can find the answer at the bottom of this page. Your answer should be two words with a space between them.

Optional

Name:

[Login](#) | [Preferences](#) | [Trac Guide](#) | [About Trac](#) | [Register](#)

Nvaccess Developed and supported by NV Access, an organization committed to helping and promoting free and open source solutions for blind and vision-impaired people.

trac Powered by Trac 1.4.1
The Distributed Subversion

Copyright © 2014 NV Access, www.nvaccess.org

Figura 2.1 Formulário de cadastro de usuário na plataforma “NVDA Community”

Fonte: NVDA Community (2014)

Após se cadastrar, o usuário receberá via *e-mail* a confirmação e deverá acessar o *link* para validar seu cadastro. Em seguida estará apto a realizar a autenticação para utilizar o sistema de rastreamento da plataforma.

Concluído o cadastro e a autenticação, o usuário deve acessar o menu “*Issues*” na plataforma para ter acesso ao sistema de rastreamento de *bugs* do NVDA.

A tela inicial do menu “*Issues*” é apresentada na Figura 2.2. Essa tela fornece os parâmetros e regras para que o usuário possa utilizar a ferramenta de cadastro e rastreamento de *bugs* da forma adequada e padronizada pela comunidade dos membros.

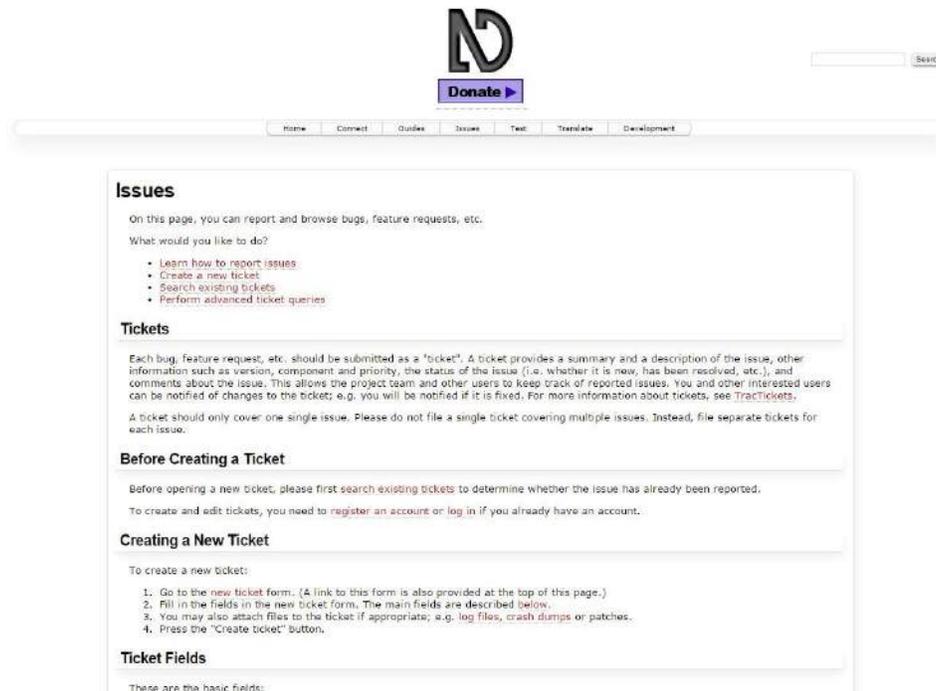


Figura 2.2 Tela do menu “Issues” contendo as diretrizes para cadastrar um *ticket* no sistema de rastreamento de *bugs*

Fonte: NVDA Community (2014)

É importante que o usuário da plataforma certifique-se, antes de realizar o cadastro, que um *ticket* semelhante ainda não tenha sido registrado no sistema. Uma busca pelos termos do *bug*/pedido de nova funcionalidade deverá ser executada para que o usuário certifique-se disso.

Os links “*Search existing tickets*” e “*Perform advanced ticket queries*” permitem o acesso à funcionalidade de busca. O primeiro *link* refere-se a uma busca mais simplificada, textual, enquanto o segundo permite que uma busca mais personalizada e elaborada possa ser executada.

A Figura 2.3 apresenta um exemplo da busca textual por um determinado termo presente na descrição de um *ticket* cadastrado anteriormente.

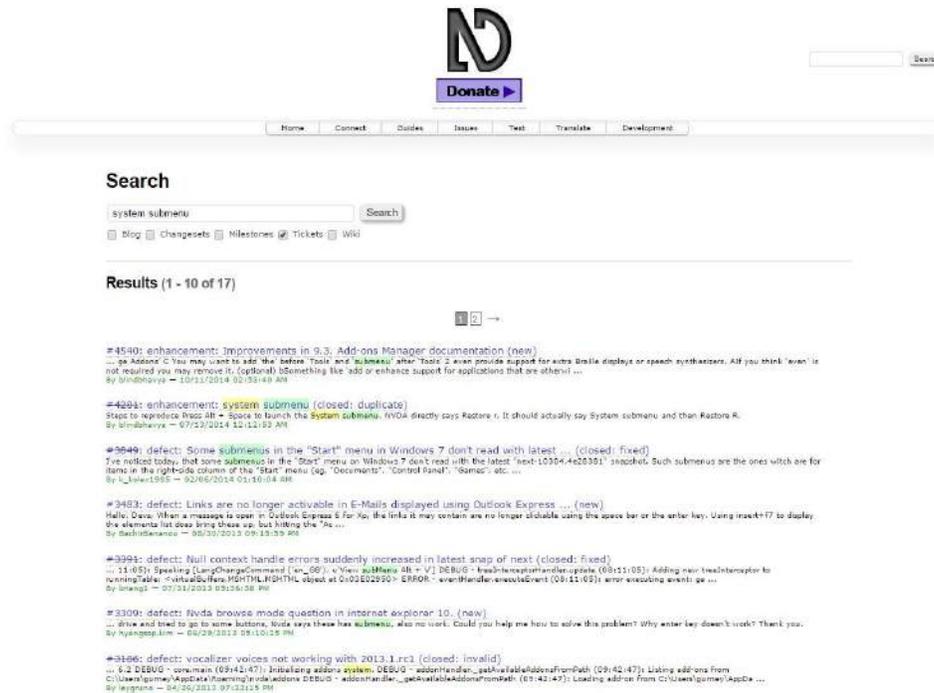


Figura 2.3 Página de busca por *tickets* já existentes no sistema

Fonte: NVDA Community (2014)

A figura acima mostra um exemplo dessa busca. Foi realizada uma consulta por *tickets* que incluísem o termo “*system submenu*” em seu resumo ou descrição. No caso acima, o sistema listou todos os *tickets* que se enquadram nessa consulta. O usuário deve analisar todos os resultados e observar se algum desses *tickets* listados incluem a descrição do *bug*/sugestão que deseja reportar. Caso não encontre um *ticket* que cubra o problema ou sugestão, o usuário deve então iniciar o cadastramento (relato).

O cadastramento de um novo *ticket* é iniciado após o usuário selecionar o link “*Create a new ticket*”, localizado na página inicial do menu “*Issues*”.

Após a seleção deste *link* uma nova janela é mostrada para que o usuário possa inserir as informações do problema/sugestão a ser reportado.

A Figura 2.4 apresenta o formulário de preenchimento de tais informações.

The image shows a web browser window displaying the 'Create New Ticket' form. At the top, there is a navigation bar with links for Home, Connect, Guides, Issues, Test, Translate, and Development. Below the navigation bar, the form is titled 'Create New Ticket'. It contains a 'Summary' field, a 'Description' field with a rich text editor toolbar, and several dropdown menus for 'Type' (set to 'defect'), 'Component' (set to 'Core'), 'Version', and 'Operating system'. There is a checkbox for 'I have files to attach to this ticket' and 'Preview' and 'Create ticket' buttons. The page footer shows 'nvaccess' and 'trac' logos, along with copyright information for 2014.

Figura 2.4 Formulário de preenchimento das informações do *ticket* a ser cadastrado

Fonte: NVDA Community (2014)

No cadastro é solicitado um breve resumo sobre a questão a ser abordada no *ticket* (através do campo “*Summary*”) e a descrição detalhada do caso, bem como a forma de reproduzir o erro se for o caso (através do campo “*Description*”).

Também é necessário que seja selecionado o tipo do *ticket* no momento do cadastro. O usuário pode escolher 3 (três) tipos distintos:

- *Defect*: quando o *ticket* abordar um defeito ou algo que não funciona como o esperado no sistema;
- *Enhancement*: quando o *ticket* abordar uma sugestão de melhoria em algumas das funcionalidades já existentes;
- *Task*: quando o *ticket* abordar uma tarefa ou trabalho a ser realizada e que não se enquadre em nenhuma das duas categorias listadas acima.

Pode, ainda, ser selecionada pelo usuário a versão do sistema que o conteúdo do *ticket* se refere, o sistema operacional que o *bug*/sugestão de funcionalidade acomete e ainda o componente específico que a questão atinge (por exemplo, *core*, *installer*, *documentation*, etc.). Caso seja necessário, o usuário pode anexar algum documento que facilite a compreensão na possível correção do caso (como por exemplo, uma foto da tela do *bug*, um *log* de erro do sistema, etc.). Quanto mais informações, mais fácil para os desenvolvedores identificarem a questão colocada, para que então seja resolvida.

Para finalizar o envio do novo *ticket* basta que o botão “*Create Ticket*” seja acionado. O *ticket* então passará a ser público e estará disponível na ferramenta de rastreamento para discussão entre os membros da comunidade e também para avaliação e correção pelos desenvolvedores.

Após o relato de um *ticket*, o usuário responsável pelo seu cadastro ou qualquer outro membro da comunidade poderá modificar o seu conteúdo a fim de melhorar a descrição.

A Figura 2.5 apresenta um exemplo de um *ticket* cadastrado

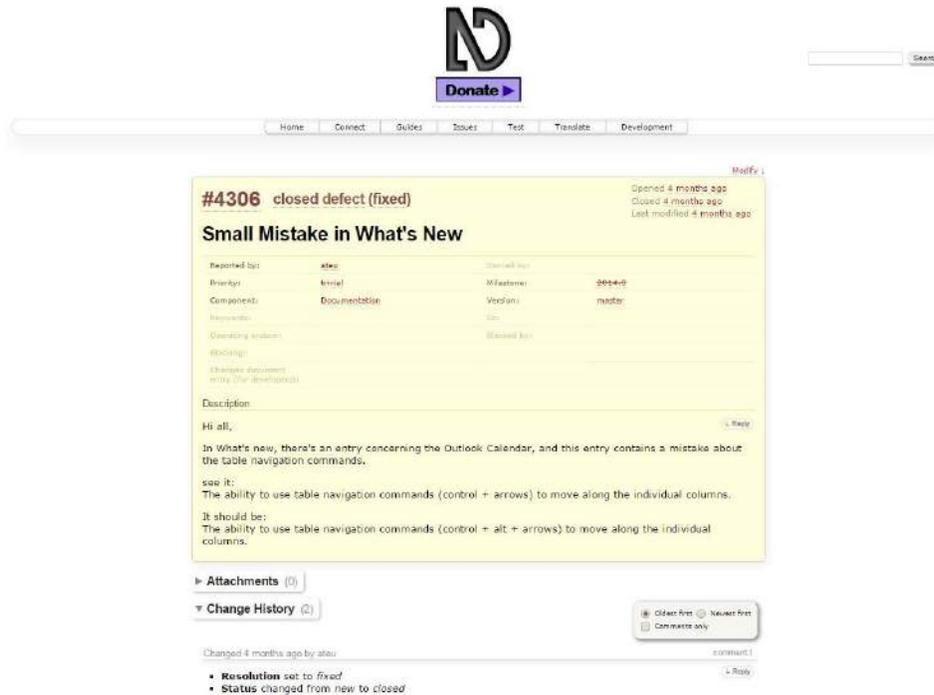


Figura 2.5 Exemplo de um *ticket* cadastrado no sistema de rastreamento de bugs/melhorias do NVDA

Fonte: NVDA Community (2014)

2.4.1.3. Processo de análise e correção de um *ticket*

O processo de análise e correção de um *ticket* inicia-se após o mesmo ter sido cadastrado por um usuário/desenvolvedor. Como o *ticket* estará disponível publicamente logo após sua criação, todos os usuários da comunidade terão acesso, e poderão discutir sobre o mesmo, contribuindo para a melhoria da descrição do problema/sugestão da funcionalidade.

Um usuário com o perfil de desenvolvedor pode analisar também questões técnicas sobre o conteúdo do *ticket*, validando-o e podendo atribuir

uma prioridade de correção (severidade) em relação ao tema. As prioridades aplicáveis a cada *ticket* podem ser:

- *Critical* (prioridade máxima – indicado para correção imediata);
- *Major* (prioridade alta para correção);
- *Minor* (prioridade menor para correção);
- *Trivial* (prioridade baixa de correção).

Quando um *ticket* é cadastrado por um usuário que não tenha o perfil de desenvolvedor, automaticamente o sistema de rastreamento atribui a severidade “*Minor*” para o caso. Porém, quando realizar a análise do *ticket*, o desenvolvedor poderá modificar sua prioridade caso julgar necessário.

Uma informação que também pode constar em um *ticket* é o “*owner*”. O *owner* é o desenvolvedor que está na responsabilidade de avaliação e correção do caso informado. Essa atribuição de *owner* poderá ser realizada por qualquer desenvolvedor no momento em que a análise do caso for iniciada.

Através destas definições de casos prioritários, os desenvolvedores poderão escalar aqueles casos que serão corrigidos primeiro e poderão então montar o plano de correção levando em consideração todas as demandas e discussões envolvidas em cada *ticket*.

Para apoiar os desenvolvedores em todo o processo e também para organizar os *tickets*, o sistema de rastreamento fornece um esquema de classificação denominado “*status*”. O *status* define a situação atual de análise/correção de um *ticket*. O Quadro 1 apresenta os possíveis *status* que um determinado *ticket* pode receber e seu significado.

Quadro 1 Status em que um *ticket* poderá tramitar durante o processo de correção

<i>Status</i>	Descrição da situação
<i>New</i>	Indica um <i>ticket</i> recém-criado. Nesse <i>status</i> o <i>ticket</i> encontra-se em processo de análise e correção por um desenvolvedor.
<i>Assigned</i>	Indica que um desenvolvedor analisou o <i>ticket</i> e atribuiu para correção, podendo ser outro desenvolvedor ou a si mesmo. Após a atribuição, passa a constar o nome (ou <i>e-mail</i>) do desenvolvedor no campo “ <i>owner</i> ” do <i>ticket</i> .
<i>Accepted</i>	Indica que um desenvolvedor analisou o <i>ticket</i> aceitou e o colocou na fila para ser corrigido/implementado.
<i>Incubating</i>	Indica que o defeito reportado pelo <i>ticket</i> está sob análise para que seja resolvido ou conhecidas suas causas.
<i>Closed</i>	Indica o fechamento (resolução) do <i>ticket</i> . Os motivos para seu fechamento são: <ul style="list-style-type: none"> ▪ <i>Fixed</i>: quando o caso foi corrigido ou quando a sugestão de nova funcionalidade/recurso foi implementada; ▪ <i>Invalid</i>: quando o caso apresenta alguma irregularidade ou invalidez; ▪ <i>Won'tfix</i>: quando o caso foi analisado e definido que não será corrigido/implementado; ▪ <i>Duplicate</i>: quando o caso reportado já foi abordado por outro <i>ticket</i>, cadastrado anteriormente; ▪ <i>Worksforme</i>: quando o caso apresentado no <i>ticket</i> caracteriza-se como a forma normal de funcionamento do NVDA e que não será modificado; ▪ <i>Can'tfix</i>: quando o <i>bug</i>/sugestão de recurso reportado pelo <i>ticket</i> não for possível de ser corrigido/implementado.
<i>Reopened</i>	Indica que um <i>ticket</i> foi reaberto por algum motivo (exemplo: <i>bug</i> não corrigido adequadamente, etc.).

Ao corrigir/verificar um caso, os desenvolvedores poderão ainda definir em qual versão do NVDA a correção/implementação do conteúdo solicitado no *ticket* estará à disposição para toda a comunidade de usuários. A essa versão prevista, denomina-se “*Milestone*” e cada uma delas possui um código de identificação próprio (por exemplo, “*NVDA 2014.1*”, “*NVDA 2014.3rc3*”, “*NVDA 2014.2*”).

A decisão pela correção/implementação das questões apontadas nos *tickets* leva em consideração a discussão entre a comunidade global de usuários e os desenvolvedores em cada *ticket*, cabendo aos desenvolvedores responsáveis por manter o NVDA, a tomada da decisão final.

2.5. Trabalhos relacionados

Como trabalhos relacionados ao tema que esse estudo aborda, temos diversos trabalhos que se assemelham na forma de tratamento de questões de usabilidade em projetos de *software* livre.

Twidale e Nichols (2005) realizaram um estudo sobre as discussões públicas envolvendo alguns projetos *open source*. Por meio da análise de relatórios de erros (*bugs*) dos projetos estudados, os autores realizaram um levantamento das características da interação entre os usuários nas discussões sobre o tema usabilidade e *design* de interfaces nos projetos. Essas discussões e os resultados obtidos permitem fomentar a melhoria na forma de como abordar questões de usabilidade no relato de casos dessa natureza, de forma em que fiquem explícitos. Exemplos práticos da interação entre os usuários para estes casos são apresentados. Contudo, não foi possível observar claramente quais os fatores de usabilidade devem estar explícitos, apenas como devem ser apresentados nos relatos de erros desses projetos.

Iivari (2008) realizou um estudo de caso interpretativo com o objetivo de compreender como a usabilidade é entendida em projetos de desenvolvimento de *software open source*. O estudo analisou as mensagens trocadas no fórum de discussão do projeto analisado. É deixado claro que os esforços para entendimento de questões de usabilidade nesses tipos de projetos ainda são poucos e pobres, apesar de que tem sido crescente e se mostrado um tema relevante para pesquisas atuais nesse contexto. Apesar da estrutura bem definida do estudo, os resultados obtidos não são tão claros quanto às formas em que podem levar a ações de melhorias do cenário analisado. Como exemplo, pode ser citado a não definição exata de papéis e integração dos especialistas de IHC (*Interface Humano Computador*) na atuação em tais projetos.

Já no trabalho de Hedberg e Iivari (2009) é abordada justamente esta questão da integração de profissionais especialistas em tais projetos, e de forma mais focada em como a comunicação é importante nesse aspecto de integração. É proposto um modelo de atuação destes profissionais de maneira a permitir que tenham autonomia no processo de tomada de decisões de usabilidade. A questão-chave, não explícita nesse trabalho, é a não informação de como deve ser o foco (maior ou menor) em determinadas nuances críticas de usabilidade desses projetos. Todavia, os resultados já permitem ter uma definição na forma de agir quando questões de usabilidade são levantadas.

Em um trabalho mais simplista apresentado em um *workshop*, Frishberg et al. (2002) discutiram sobre o que deve-se saber sobre questões de usabilidade em projetos de desenvolvimento de *software open source*. Antes mesmo da constatação feita por Iivari (2008), explicando que o tema “Usabilidade” tem um alvo de relevantes estudos de casos, esse trabalho já antevia que a usabilidade iria se tornar um valor fundamental no processo de desenvolvimento desses projetos para essa área. Por ser um trabalho mais simples, apenas é indicado

quais esforços devem ser encorajados para que o valor da usabilidade seja realmente conhecido nesse contexto. Não são apresentados quais pontos críticos em usabilidade devem receber uma atenção maior para que sua compreensão seja difundida de maneira mais acelerada nos projetos de *software open source*.

Despalatović (2013) apresentou em seu recente estudo uma revisão de artigos científicos com o intuito de demonstrar se existem problemas de usabilidade em projetos de desenvolvimento de *software* livre. Também verificou se existem soluções para possíveis problemas. A resposta obtida foi que realmente existem problemas. Também é constatado que essa área de desenvolvimento ainda não é suficientemente reconhecida por profissionais de IHC ou Engenharia de *Software* de modo em que não é objeto de grandes números de estudos na área. Essa afirmação contradiz a de Iivari (2008), onde a autora afirma que tais estudos têm sido cada vez mais conduzidos por pesquisadores da área. Os resultados obtidos por Despalatović (2013) não apresentam questões práticas de usabilidade que podem ser priorizadas, deixando apenas exposto quais processos e ferramentas que apoiam a definição de características de usabilidade dos projetos.

Anteriormente, Andreasen et. al (2006) também já argumentaram que os projetos de *software open source* não apresentam tanta ênfase na questão de usabilidade desses sistemas. É apresentado um estudo empírico contendo opiniões de desenvolvedores em relação à usabilidade e sobre como ela é praticada nos mais variados projetos. Como resultados, apresentam que os desenvolvedores se mostram interessados em alavancar essa área, porém ainda, na prática não é o que pode ser observado. Assim, percebe-se que é altamente necessário conhecer primeiro o que deve ser focado em usabilidade para que esses interesses sejam refletidos também na prática dos projetos.

3. MÉTODO

Este capítulo caracteriza a pesquisa realizada e os procedimentos metodológicos que envolvem o estudo.

3.1. Caracterização da pesquisa

A pesquisa proposta caracteriza-se, segundo classificação definida por Wainer (2007), como empírica, de natureza aplicada, qualitativa quanto à forma de abordagem, exploratória quanto aos objetivos, sendo realizada através do procedimento de estudo de caso.

Ainda segundo Wainer (2007) os métodos qualitativos diferem-se de métodos quantitativos pelo fato de que se ocupam de variáveis que não podem ser medidas, apenas observadas.

Em Ciência da Computação, de um modo geral, os métodos qualitativos se caracterizam por ser um estudo aprofundado de um sistema no ambiente onde ele está sendo usado, ou, em alguns casos, onde se espera que o sistema seja usado. Os métodos qualitativos sempre envolvem pessoas e na maioria das vezes sistemas (Wainer, 2007).

Em relação ao estudo de caso, Wainer (2007) o caracteriza com um estudo onde o pesquisador interage com os sujeitos geralmente de uma forma semiformal através de entrevistas e conversas programadas, normalmente tendo acesso a documentos, dados e outros materiais formais da organização. Ele ainda cita que o objetivo da pesquisa de estudo de caso é descobrir “o que as pessoas escrevem” e “o que as pessoas dizem”, ou, em outras palavras, descobrir as práticas formais da organização, valores, opiniões e atitudes dos sujeitos.

O estudo de caso desse trabalho foi realizado por meio de uma análise de artefatos de projeto, disponibilizados publicamente pela plataforma de rastreamento de *bugs* e reporte de casos/questões (*issues*) do projeto NVDA.

Neste estudo, não foi efetuada nenhuma intervenção direta com usuários ou desenvolvedores. O estudo de caso teve como objetivo identificar a forma como os projetos são conduzidos e como se dá a interação com os usuários com deficiência do *software* NVDA, por meio da análise dos dados (*tickets*) na plataforma pública de suporte ao desenvolvimento desse *software*.

3.2. Procedimentos metodológicos

Neste tópico são disponibilizados os procedimentos metodológicos utilizados na pesquisa. São descritos:

- O “*Design*” do método utilizado, que consiste em definição geral do método utilizado para realização da pesquisa e o porquê de sua utilização;
- Os procedimentos adotados para a coleta e organização dos dados;
- Os procedimentos utilizados para realização da análise e classificação dos dados coletados.

3.2.1. *Design*

Definimos como “*Design*” o arranjo geral da metodologia para coleta e análise dos dados desta pesquisa aplicada.

Foi utilizado como base o método de análise de dados denominado “Análise de Conteúdo”.

Esse método, amplamente utilizado em pesquisas cuja abordagem é qualitativa, é definido por Stemler (2001) como uma técnica sistemática e replicável, utilizada para comprimir palavras em categorias mais abstratas. Hoslti (1969, p. 24, tradução nossa) já definia a análise de conteúdo como “uma técnica para fazer inferências de forma objetiva e sistemática da identificação de características específicas de mensagens”. Para esse contexto, deve ser entendido como “conteúdo” qualquer organização de dados, seja na forma textual ou multimídia, para compreensão de um determinado assunto (Lazar, 2010).

Lazar (2010) explica que a análise de conteúdo realiza uma verificação em profundidade de um determinado conteúdo, na maioria das vezes textual, para geração de determinados conhecimentos sobre o texto analisado, uma vez que não podem ser “avaliados” de forma quantitativa.

O processo de analisar um determinado conteúdo e em seguida dele extrair informações, categorias ou padrões, é denominado “codificação”. A codificação é aplicada a um conjunto bem definido de dados, denominados de “*data-set*”. A partir desse *data-set*, inicia-se a obtenção da informação que se deseja classificar (Lazar, 2010).

A pessoa responsável pela análise e codificação do *data-set* é denominada “codificador” (Lazar, 2010). Em uma análise, podem ser utilizados quantos codificadores forem necessários para que se obtenham com precisão e validade os padrões de informações que se deseja colher a partir dos dados.

Assim, para o estudo em questão, esse método foi aplicado a um conjunto significativo de *tickets* (*data-set* do estudo) colhidos na plataforma pública de interação entre desenvolvedores e usuários do *software* NVDA, a fim de identificar quais questões-chaves de usabilidade são mais críticas/prioritárias

no processo de desenvolvimento desse *software*, e também para identificar outras características inerentes a esse processo de correção dos *tickets*.

Conforme já citado, cada *ticket* cadastrado/reportado por um usuário ou desenvolvedor descreve um caso de erro (*bug*) encontrado no *software* NVDA, sendo solicitada a sua manutenção ou até mesmo descreve uma sugestão de melhoria ou funcionalidade a ser adicionada ao *software* no futuro.

A Figura 3.1 apresenta um exemplo de um *ticket* cadastrado por um usuário. Nesse *ticket* é verificado um pedido de uma nova funcionalidade para leitura de legendas de vídeos pelo NVDA.

#2797 new enhancement
Opened 10 months ago
Last modified 5 months ago

Support for reading subtitles in videos

Reported by:	nvdakor	Owned by:	
Priority:	minor	Milestone:	
Component:	Core	Version:	
Keywords:		Cc:	
Operating system:		Blocked by:	
Blocking:			
What's New entry:			

Description

Hi,
A number of users use a screen reader which reads subtitles on videos (for example, language subtitles for foreign film videos). Currently, NVDA does not read subtitles present in some videos, so would it be possible to allow NVDA to read them? Thanks.

▼ **Change History** (4)

Oldest first
 Newest first
 Comments only

Changed 10 months ago by jteh comment:1

Please provide an example URL or application. The way subtitles are implemented differs greatly between video players. It'd also be good to know what screen readers do this.

Changed 10 months ago by nvdakor comment:2

Hi,
Just asked some users. They told me that they're using a Korean screen reader called Sense Reader. They are also using GOM player, as this is the only media player that people can listen to subtitles using Sense Reader. In addition, users suggested adding subtitle reading support for VLC media player as well. Thanks.

Changed 7 months ago by jteh comment:3

We can't add support for this unless the player exposes them to accessibility or native APIs.

Changed 5 months ago by vortex comment:4

I wrote a program which can do this for external subtitle files, and it is compatible with NVDA. You can find more at:
[my website](#)

Figura 3.1 Exemplo de um *ticket* com sugestão de nova funcionalidade reportada por um usuário.

Fonte: NVDA Community (2014)

A escolha desse método de pesquisa se deu pelos seguintes motivos:

- Pela forma qualitativa de abordagem do estudo;
- Pelo fato de que os dados a serem analisados encontrarem-se disponíveis em uma forma sistematicamente não organizada

nessa plataforma, onde não há relação de conteúdo entre quaisquer *tickets* cadastrados;

- E principalmente pelo objetivo geral do estudo proposto.

Desta forma, identificou-se que a Análise de Conteúdo seria o método mais adequado para a proposta do estudo.

Na seção a seguir detalha-se o processo para coleta de dados para a pesquisa.

3.2.2. Coleta e organização dos dados

A maneira de escolha do conjunto de dados (*data-set*) para realização desta pesquisa foi definida em comum acordo entre os codificadores. O estudo contou com a participação de 2 (dois) pesquisadores que atuaram como codificadores nessa classificação. O primeiro pesquisador foi Douglas Moreira (autor do trabalho) e o segundo o professor André Pimenta (orientador). Tais pesquisadores já apresentavam experiência em codificações anteriores.

O *data-set* contemplou os 200 *tickets* mais recentes⁸, colhidos no sistema e cujo *status* se enquadravam como “*closed*” (fechado), excluindo-se os casos fechados pelos motivos “*invalid*” (inválidos) e “*duplicate*” (duplicados).

A utilização de somente *tickets* “fechados” foi pelo fato de possuírem uma posição final de correção/implementação (ou de não correção/implementação), definida após a discussão entre os membros da comunidade e desenvolvedores. Já a decisão por não incluir os casos “inválidos” e “duplicados” visava que o *data-set* apresentasse apenas casos consistentes para a análise e codificação pelos pesquisadores.

⁸ O conjunto de *tickets* foi colhido em Setembro de 2014

A partir dos *tickets* selecionados, foi preparada e montada uma planilha para armazenamento e organização do conjunto de dados. Cada linha de dados na planilha se refere a um *ticket* único e as colunas aos detalhes (características) do mesmo. Os detalhes apresentados são:

- Número do *ticket* no *data-set* (1 a 200);
- Número identificador único do *ticket* na plataforma de rastreamento (ex: 4522);
- O tipo do *ticket* (“*defect*”, “*enhancement*” ou “*task*”);
- O *owner*⁹ do *ticket*;
- O usuário que reportou o *ticket*; e
- O motivo de fechamento desse *ticket* (“*fixed*”, “*worksforme*”, “*wontfix*” ou “*cantfix*”, explicados no Quadro 1).

O intuito dessa organização foi apenas para facilitar e preparar os dados para a codificação realizada pelos pesquisadores.

Também foi inserida uma coluna para a categorização do *ticket* (de “Usabilidade” ou “Funcionalidade”) que foram preenchidas pelos pesquisadores no momento da codificação.

Ainda foram inseridas nesta planilha mais quatro colunas, referentes às heurísticas de Nielsen que foram violadas individualmente pelos *tickets* classificados como “Usabilidade”.

Esta planilha completa poderá ser verificada no APÊNDICE A deste trabalho.

⁹ Desenvolvedor responsável pela análise e correção do caso na ferramenta *trac*

Após a seleção e preparação desse *data-set*, deu-se início à etapa de análise dos dados. Na seção seguinte serão descritos detalhes de como se deram os procedimentos para análise dos dados.

3.2.3. Métodos de análise de dados

A partir do conjunto de *tickets* selecionados e preparados, a análise de dados foi iniciada com o apoio da técnica de análise de conteúdo já apresentada na Seção 3.2.1.

Ainda sobre a análise de conteúdo, ela possui como uma de suas características essenciais a utilização de categorias para a extração sistemática de dados qualitativos, as quais podem ser obtidas a partir de modelos teóricos ou a partir de codificação aberta de dados, para que as categorias emergam dos dados (Flick, 2009).

Mais especificadamente, foi realizada a análise de conteúdo textual sobre os *tickets* selecionados para compor o *data-set*. Essa análise de conteúdo foi realizada por um pesquisador, e onde haviam dúvidas ou ambiguidades, as questões foram solucionadas por meio da discussão com um segundo pesquisador

A análise ocorreu em duas etapas. A primeira, relativa à codificação do conjunto completo dos *tickets* para a categorização individual dos mesmos, como problemas de usabilidade ou funcionalidade. A segunda fase é relativa à classificação dos *tickets* de “Usabilidade”, quanto às heurísticas de Nielsen violadas pelos casos.

A primeira etapa dessa análise ocorreu com a codificação de todos os *tickets* do *data-set* em busca de categorizá-los em classes mais abstrata de compreensão, assim como define a técnica de Análise de Conteúdo. A partir do

comum acordo entre os pesquisadores, foi definido que as categorias de classificação dos *tickets* fossem “Usabilidade” e “Funcionalidade”.

Para essa classificação foi levado em consideração definições básicas dos termos e significados das mesmas de acordo com normas e outros autores de trabalhos da área.

De uma forma mais ampla, a categoria “Usabilidade” englobou os *tickets* que apresentavam questões que feriam o modo de uso satisfatório do usuário em relação à forma de operação de algum componente ou modo de operação do NVDA. Já a categoria “Funcionalidade” englobou os *tickets* que reportaram pedidos de inclusões de novas formas de operação do sistema ou mudanças do esquema normal de funcionamento do mesmo que não tangessem a “usabilidade” do sistema.

Essa categorização permitiu que fossem separados os *tickets* que apenas focassem questões de usabilidade (categorizados em “Usabilidade”) daqueles que não faziam parte desse escopo (categorizados em “Funcionalidade”). Os dados obtidos por essa categorização foram inseridos na coluna destinada para armazenar tal informação na planilha de organização do *data-set*, citada na seção anterior.

A partir dessa categorização, passou-se então à segunda etapa da análise dos dados. Essa etapa consistiu-se na análise dos *tickets* pertencentes à categoria “Usabilidade”, visando verificar quais das 10 heurísticas de Nielsen eram violadas em cada um.

Assim, foi definida a seguinte metodologia: para cada *ticket* analisado nessa segunda etapa da análise, foi apontada uma heurística principal ao qual fosse a visivelmente mais violada entre as do conjunto de Nielsen. Também foram apontadas heurísticas complementares que por ventura também foram violadas pela descrição da questão apresentada pelo *ticket*.

Após a definição da heurística principal e das complementares para um *ticket*, os codificadores realizaram duas perguntas sobre essa classificação:

Q1. A aplicação das boas práticas de usabilidade definidas apenas pela heurística principal permite resolver a questão completa apontada pelo ticket?

Q2. As boas práticas de usabilidade apontadas pela heurística principal em conjunto com as heurísticas complementares permitem resolver a questão completa apontado pelo ticket?

Para estas perguntas, foram respondidos “Sim” ou “Não” para cada *ticket* classificado anteriormente como de “Usabilidade”.

A definição das heurísticas violadas e também as respostas para as perguntas foram definidas em comum acordo entre os 2 (dois) pesquisadores que atuaram como codificadores na classificação dos *tickets* em categorias.

O Quadro 2 apresenta um exemplo fictício desta classificação heurística dos *tickets* de usabilidade.

Quadro 2 Exemplo de uma a classificação das heurísticas de Nielsen para os *tickets* de “Usabilidade”

Número do <i>ticket</i>	Heurística Principal violada	Heurísticas Complementares violadas	Somente a Heurística Principal resolve a questão o <i>ticket</i>?	O conjunto de heurísticas principais e complementares resolvem a questão do <i>ticket</i>?
#4550	H5	H1	Não	Sim
#4555	H7	H2, H6.	Sim	Sim

Os dados verídicos foram incorporados à planilha de organização do *data-set*, naquelas quatro colunas separadas para essa classificação e citada na seção anterior.

Após a realização destas etapas, foi possível concluir a etapa de análise dos dados. Essa análise possibilitou apresentar os resultados e fomentou a formulação das hipóteses inerentes ao estudo.

4. RESULTADOS E DISCUSSÃO

Nos subtópicos seguintes serão apresentados os dados obtidos pela análise dos fatores críticos de usabilidade e funcionalidade envolvidos no processo de desenvolvimento do *software* NVDA.

4.1. Dados resultantes da análise

A partir dos dados da análise, foi possível conhecer e elencar algumas questões prioritárias de usabilidade e características de funcionalidade que podemos considerar críticas para o sucesso e crescimento do uso do *software* leitor de telas NVDA. Esses dados estão listados a seguir.

4.1.1. Funcionalidade vs. usabilidade

Os problemas categorizados como “Funcionalidade” reportavam pedidos de inclusão de novos itens ao sistema ou questões específicas de comportamento que não apresentavam muitas características envolvendo a usabilidade do sistema, mas de requisitos puramente funcionais.

Já os problemas categorizados como “Usabilidade” reportavam *bugs* ou pedidos de mudanças no sistema ligados diretamente a questões de usabilidade, inerentes à forma da interação do sistema com o usuário que poderia interferir na experiência ou conforto dos mesmos na utilização do NVDA.

Do total de 200 *tickets* analisados, suas categorizações forneceram os resultados apresentados na Tabela 1.

Tabela 1 Categoria dos *tickets* analisados (em número bruto e %)

Categoria	Nº de <i>tickets</i>	% de <i>tickets</i>
Funcionalidade	51	25,5%
Usabilidade	149	74,5%

Destaca-se aqui o número elevado de casos de “Usabilidade”. Esse fato reforça que questões de usabilidade são um fator de extrema importância no processo de desenvolvimento do *software* NVDA e que merecem atenção no desenvolvimento do projeto.

Foram separados nessa monografia exemplos de casos categorizados como “Usabilidade” e também como “Funcionalidade”

O caso #4306 apresenta um relato de *bug* na seção “*What’s new*” sobre uma nova *release* (lançamento de nova versão) do NVDA. A Figura 4.1 apresenta a descrição deste *ticket*.

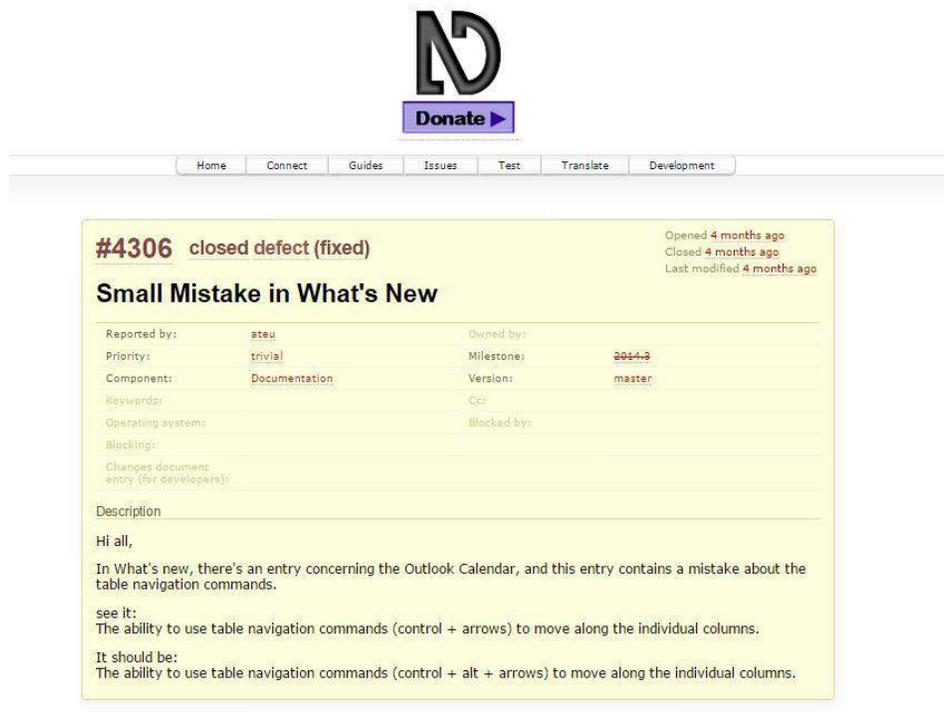


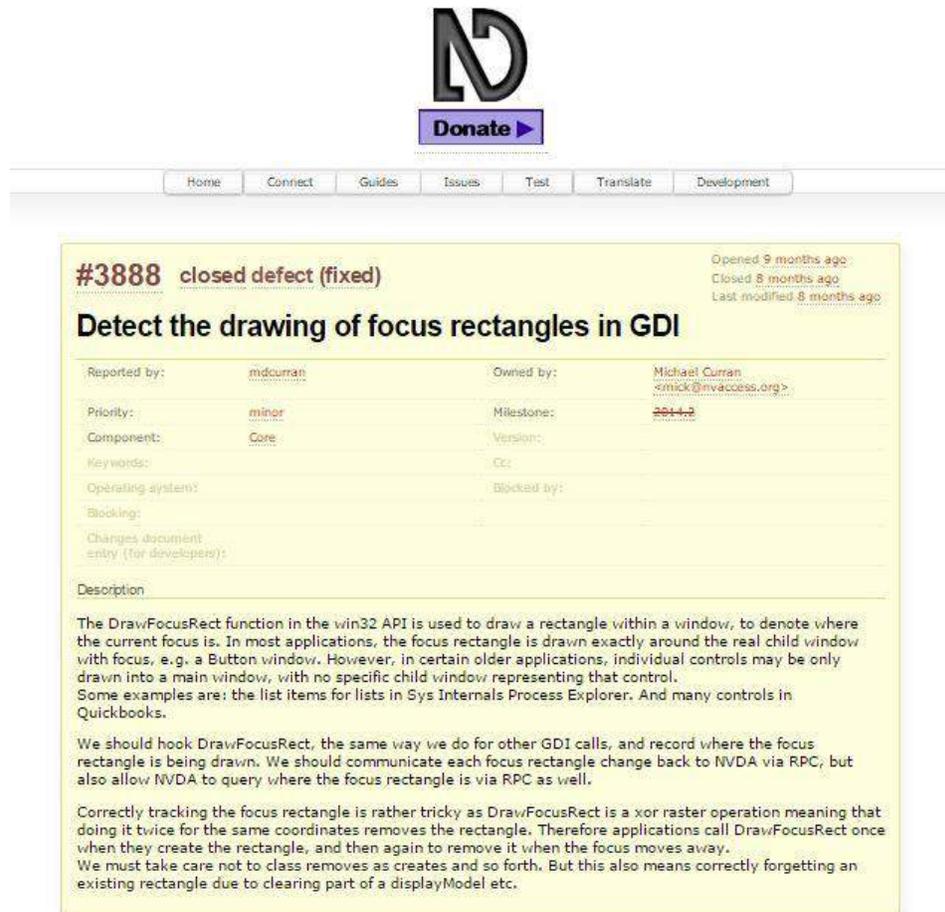
Figura 4.1 *Ticket* reportando um problema de usabilidade

Fonte: NVDA Community (2014)

O caso descrito na Figura 4.1 apresenta uma solicitação de correção na documentação relativa às novidades de uma versão mais atualizada do NVDA (“*What's New*”). Nessa documentação é apresentado o comando de acesso à uma ação do sistema, porém esse comando é mostrado com erro (incompleto).

Este *ticket* foi categorizado como pertencente à categoria de “Usabilidade” pelo fato de que a experiência do usuário na utilização da documentação do *software* não ocorreu de forma satisfatória.

Já o caso #3888 apresenta um relato de melhoria relativa à forma que o NVDA utiliza uma determinada *API* (*Application Programming Interface*) para desenhos de retângulos de focos em uma janela. A Figura 4.2 apresenta a descrição deste *ticket*.



NVDA
Donate ▶

Home Connect Guides Issues Test Translate Development

#3888 closed defect (fixed) Opened 9 months ago
Closed 8 months ago
Last modified 8 months ago

Detect the drawing of focus rectangles in GDI

Reported by:	mdcurren	Owned by:	Michael Curran <mick@nvaccess.org>
Priority:	minor	Milestone:	2014.2
Component:	Core	Version:	
Keywords:		CC:	
Operating system:		Blocked by:	
Blocking:			
Changes document entry (for developers):			

Description

The DrawFocusRect function in the win32 API is used to draw a rectangle within a window, to denote where the current focus is. In most applications, the focus rectangle is drawn exactly around the real child window with focus, e.g. a Button window. However, in certain older applications, individual controls may be only drawn into a main window, with no specific child window representing that control. Some examples are: the list items for lists in Sys Internals Process Explorer. And many controls in Quickbooks.

We should hook DrawFocusRect, the same way we do for other GDI calls, and record where the focus rectangle is being drawn. We should communicate each focus rectangle change back to NVDA via RPC, but also allow NVDA to query where the focus rectangle is via RPC as well.

Correctly tracking the focus rectangle is rather tricky as DrawFocusRect is a xor raster operation meaning that doing it twice for the same coordinates removes the rectangle. Therefore applications call DrawFocusRect once when they create the rectangle, and then again to remove it when the focus moves away. We must take care not to class removes as creates and so forth. But this also means correctly forgetting an existing rectangle due to clearing part of a displayModel etc.

Figura 4.2 *Ticket* reportando um problema de funcionalidade

Fonte: NVDA Community (2014)

O caso reportado na Figura 4.2 apresenta uma proposta para otimização da utilização desta *API* no contexto do NVDA. Devido ao fato de que o *ticket*

relata uma questão de mudança no esquema de operação de uma tarefa no NVDA, não tendo uma relação direta quanto à satisfação do usuário em sua experiência de uso, esse *ticket* foi categorizado como de “Funcionalidade”.

4.1.2. Usuários que reportaram os *tickets*

Através dos resultados, foi possível observar que a comunidade de usuários do NVDA é bastante engajada e participativa no projeto. Vários relatores diferentes dos *tickets* foram notados. O Quadro 3 apresenta o número de relatores distintos para os *tickets* que compuseram o *data-set* analisado.

Quadro 3 Número de relatores distintos que compuseram o *data-set* de *tickets* analisados

Nº de relatores distintos dos <i>tickets</i> componentes do <i>data-set</i>	65
--	----

Apesar de a plataforma de rastreamento dos *tickets* não apresentar um mecanismo claro para diferenciação entre usuários finais do NVDA e usuários que também são desenvolvedores, pela análise dos *tickets* foi possível reconhecer apenas três usuários que possuíam o perfil de desenvolvedor, uma vez que foi possível observar esta informação pelo *owner* de alguns casos e também pelo papel e argumentos que cada usuário apresentou nas discussões que envolviam cada *ticket*.

Porém, essa informação não mostrava-se tão precisa, uma vez que a mesma não é apresentada de forma explícita na plataforma. Assim, também foi

adotado o procedimento de verificação dos registros de *commits*¹⁰ efetuados no código do NVDA (através do sistema de versionamento de código *Git*) na busca por mais desenvolvedores que por ventura não tenham sido identificados no conjunto de relatores dos *tickets* do *data-set*.

Com isso, foi certificado que não possuía nenhum outro usuário com o perfil de desenvolvedor entre os relatores dos *tickets* analisados. Desta forma, apenas os três usuários desenvolvedores levantados anteriormente foram considerados para a pesquisa.

Destes 3 usuários desenvolvedores, foi possível observar que 2 deles eram os idealizadores do projeto NVDA: *Michael Curran* e *James Teh*. O outro desenvolvedor aparecia com o nome *Mesar Hameed*.

Também foi possível observar que esses desenvolvedores, além de responsáveis por julgar e corrigir as questões apontadas nos *tickets*, também atuaram como relatores. Uma parcela significativa dos *tickets* que compuseram o *data-set* foram relatados por eles, conforme mostra o Quadro 4.

Quadro 4 N° de desenvolvedores que também atuaram como relatores de *tickets* que formaram o *data-set*

Desenvolvedor-relator	N° de <i>tickets</i> relatados	% do total de <i>tickets</i>
<i>Michael Curran</i>	39	19,5
<i>James Teh</i>	23	11,5
<i>Mesar Hameed</i>	2	1
Total	64	32

¹⁰ *Commit* consiste em um comando que efetiva uma transação no banco de dados de um sistema. Um *commit* só pode ser executado por um desenvolvedor que tenha acesso ao código do sistema. Para maiores detalhes, consultar GIT (2014).

Excetuando os desenvolvedores, houveram usuários finais que participaram e relataram mais *tickets* que outros. A Tabela 2 apresenta a distribuição dos principais usuários relatores dos *tickets* que compuseram o *data-set* analisado.

Tabela 2 Usuários finais que mais reportaram *tickets* no conjunto do *data-set*

Usuários relatores	Nº de <i>tickets</i> relatados	% do total de <i>tickets</i>
<i>blindbhavya</i>	21	10,5
<i>ateu</i>	7	3,5
<i>nvdakor</i>	6	3
<i>k_kolev1985</i>	5	2,5
<i>briang1, marcozehe, oaron, rnickelson, vgjh2005 (5 distintos)</i>	4 (cada)	2 (cada)
<i>erion, leonarddr, nishimotz, ondrosik, sumandogra, the_dark_man (6 distintos)</i>	3 (cada)	1,5 (cada)
<i>aleskis, dallasobrien, dgoldfield, hcdhoundiyal, jacobchawala, jorgtum, khsruru, mhorspool, msuch, pzajda, shadyar, zahari_bgr (12 distintos)</i>	2 (cada)	1 (cada)
<i>advck1123, ahiiron, aliminator, andrewd, animesh.mr, annek, ashikhirani, ashleycox, aslani.ali, baphomet, birkirrg, dhankuta, digitaltoast, driemer.riemer, finnbox64, haltersweb, hyongsop.kim, kaban, kattygeltmeyer, kredh, levilibegas, lpintes, mikeray, mk360, mohammed, motter, netblue, santosh, siddu, supanut2000, surfer0627, taghavi, twinsen, vijaynayar, xonic (35 distintos)</i>	1 (cada)	0,5 (cada)

4.1.3. Caracterização dos problemas de usabilidade em relação às Heurísticas de Nielsen

Quanto à caracterização dos *tickets* de usabilidade em relação às Heurísticas de Nielsen, foi possível obter os resultados mostrados na Tabela 3.

Tabela 3 Distribuição das heurísticas quanto aos problemas apresentados nos *tickets* classificados como de usabilidade no *data-set*

Nº da Heurística	Principal		Complementar	
	Nº de <i>tickets</i>	% de <i>tickets</i> (aprox.)	Nº de <i>tickets</i>	% de <i>tickets</i> (aprox.)
H1	55	36,91	14	9,39
H2	1	0,67	26	17,44
H3	5	3,35	3	2,01
H4	17	11,40	31	20,80
H5	2	1,34	17	11,40
H6	1	0,67	1	0,67
H7	39	26,17	30	20,13
H8	11	7,38	16	10,73
H9	14	9,39	5	3,35
H10	5	3,35	6	4,02

Esse quadro mostra a distribuição das heurísticas que foram violadas nos problemas reportados pelos *tickets* classificados como de usabilidade (149 *tickets*).

Para este conjunto, é possível notar que em relação à heurística principal inerente a um único *ticket* de usabilidade, as que mais aparecem foram a H1, que diz respeito ao *feedback* que o usuário recebe em um tempo adequado ao utilizar o sistema; a H7, que cita a questão da flexibilidade no uso do sistema através de aceleradores (atalhos) e a H4, que apresenta questões de padronização e consistência de funções e termos do sistema.

Essas heurísticas foram classificadas como principal em 55, 39 e 17 *tickets* reportados, respectivamente.

Já em relação às heurísticas complementares de um *ticket*, as que mais apareceram foram a H4, H7 e H2 (que trata da padronização entre os termos do sistema e o mundo real). Essas heurísticas foram classificadas como complementares em 31, 30 e 26 *tickets* reportados, respectivamente.

Quanto às duas perguntas realizadas após a classificação das heurísticas violadas em cada *ticket* de usabilidade, os resultados obtidos estão expostos na Tabela 4.

Tabela 4 Respostas para as perguntas efetuadas após a classificação das heurísticas violadas por cada *ticket*

Questão	Resposta “Sim”		Resposta “Não”	
	Nº	%	Nº	%
<i>Q1 – A aplicação das boas práticas de usabilidade definidas apenas pela heurística principal permite resolver a questão completa apontada pelo ticket?</i>	88	59,06	61	40,93
<i>Q2 – As boas práticas de usabilidade apontadas pela heurística principal em conjunto com as heurísticas complementares permitem resolver a questão completa apontado pelo ticket?</i>	149	100	0	0

A Tabela 4 apresenta que quase 60% dos casos de usabilidades do conjunto do *data-set* poderiam ser completamente solucionados ao se aplicar as boas práticas referentes à heurística principal associada a esse *ticket*.

Já quando leva-se em consideração a aplicação da heurística principal e das heurísticas complementares dos *tickets*, a solução dos casos seria integral, atingindo 100% dos problemas levantados.

Alguns exemplos de *tickets* foram separados para ilustrar quando ocorre a violação das heurísticas propostas por Jakob Nielsen:

- *Violação à Heurística 1: Visibilidade e status do sistema*

O caso #3913 apresenta um exemplo de violação à essa heurística. A Figura 4.3 apresenta a descrição do *ticket* desse caso.

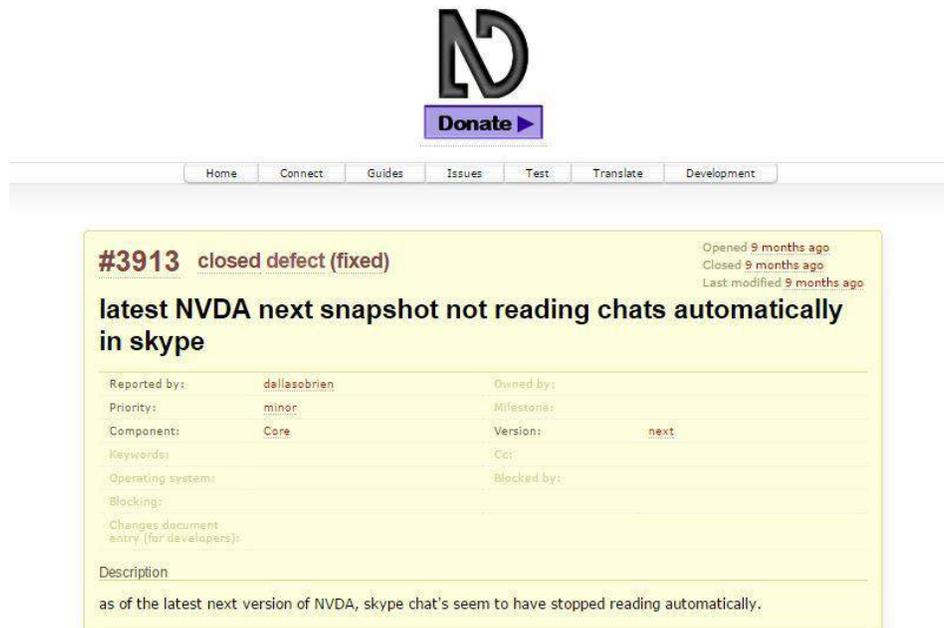


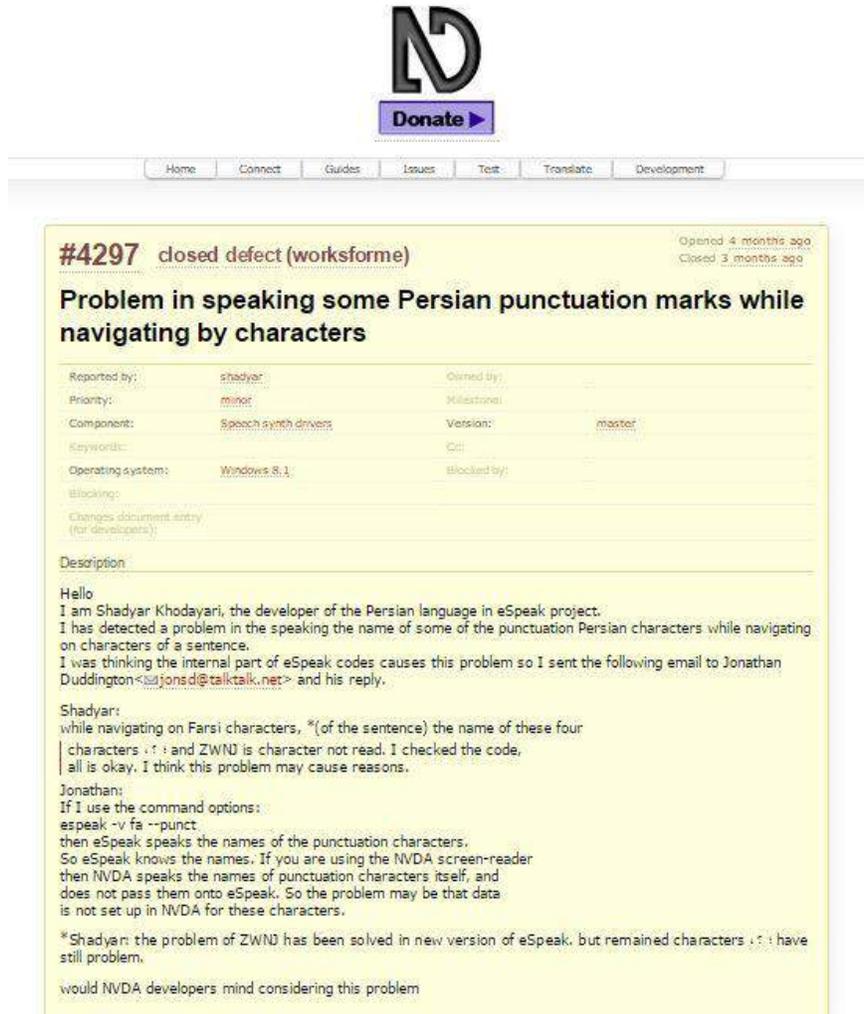
Figura 4.3 Exemplo de um *ticket* de usabilidade que viola as condições descritas pela heurística 1 de Jakob Nielsen

Fonte: NVDA Community (2014)

A violação a essa heurística pode ser notada quando o usuário cita que na última versão do NVDA não é possível que o mesmo leia automaticamente o conteúdo dos *chats* quando se utiliza o *Skype*. Isso caracteriza a falta de um *feedback* adequado ao usuário, deixando-o sem saber o que está acontecendo no sistema.

- *Violação à Heurística 2: Correspondência entre o sistema e o mundo real*

O caso #4297 apresenta um exemplo de violação a essa heurística. A Figura 4.4 apresenta a descrição do *ticket* desse caso.



N
Donate ▶

Home Connect Guides Issues Text Translate Development

#4297 closed defect (workforme) Opened 4 months ago
Closed 3 months ago

Problem in speaking some Persian punctuation marks while navigating by characters

Reported by: [shadyar](#) Owned by: [shadyar](#)
 Priority: [minor](#) Milestone: [None](#)
 Component: [Speech synth drivers](#) Version: [master](#)
 Keywords: [None](#) Cc: [None](#)
 Operating system: [Windows 8.1](#) Blocked by: [None](#)
 Blocking: [None](#)
 Changes document entry (for developers): [None](#)

Description

Hello
 I am Shadyar Khodayari, the developer of the Persian language in eSpeak project.
 I has detected a problem in the speaking the name of some of the punctuation Persian characters while navigating on characters of a sentence.
 I was thinking the internal part of eSpeak codes causes this problem so I sent the following email to Jonathan Duddington <sjjonsd@talktalk.net> and his reply.

Shadyar:
 while navigating on Farsi characters, *(of the sentence) the name of these four
 | characters : : : and ZWJ is character not read. I checked the code,
 | all is okay. I think this problem may cause reasons.

Jonathan:
 If I use the command options:
 espeak -v fa --punct
 then eSpeak speaks the names of the punctuation characters.
 So eSpeak knows the names. If you are using the NVDA screen-reader
 then NVDA speaks the names of punctuation characters itself, and
 does not pass them onto eSpeak. So the problem may be that data
 is not set up in NVDA for these characters.

*Shadyar: the problem of ZWJ has been solved in new version of eSpeak, but remained characters : : : have still problem.
 would NVDA developers mind considering this problem

Figura 4.4 Exemplo de um *ticket* de usabilidade que viola as condições descritas pela heurística 2 de Jakob Nielsen

Fonte: NVDA Community (2014)

A violação a essa heurística pode ser notada quando o usuário que se diz desenvolvedor da linguagem Persa, referente ao projeto *eSpeak* (2014), e cujo sintetizador é utilizada no NVDA. É citado que há um problema de leitura de alguns caracteres inerentes a esse sintetizador, não retratando a pronúncia

correta para a linguagem Persa e consistindo assim uma incompatibilidade da interface do sistema com representações do mundo real.

- *Violação à Heurística 3: Controle e liberdade do usuário*

O caso #4338 apresenta um exemplo de violação a essa heurística. A Figura 4.5 apresenta a descrição do *ticket* desse caso.



The image shows a screenshot of a web interface for the NVDA project. At the top, there is a large 'N' logo and a 'Donate' button. Below this is a navigation menu with links for Home, Connect, Guides, Issues, Test, Translate, and Development. The main content area displays a ticket for issue #4338, which is a 'closed defect (cantfix)'. The ticket title is 'Windows Projection Options dialog'. It includes metadata such as 'Opened 3 months ago', 'Closed 3 months ago', and 'Last modified 3 months ago'. The ticket details are as follows:

Reported by:	blindbhavya	Owned by:	
Priority:	minor	Milestone:	
Component:	Speech	Version:	2014.2
Keywords:		Cc:	
Operating system:	Windows 7	Blocked by:	

Below the metadata, there is a 'Blocking:' section with the text: 'Changes document entry (for developers)'. The 'Description' section contains the following text:

Steps to reproduce
 Press Windows + P to open the Projection Options dialog box.
 When you are sure that it has opened, keep the right arrow pressed down for a 10 seconds.
 NVDA should start saying all the options of the dialog over and over again. During this, press the right arrow key. Ideally, NVDA should stop everything else and say only the new option gaining focus.

Figura 4.5 Exemplo de um *ticket* de usabilidade que viola as condições descritas pela heurística 3 de Jakob Nielsen

Fonte: NVDA Community (2014)

A violação a essa heurística pode ser notada quando o usuário apresenta a situação envolvendo as opções de diálogo para projeção das janelas. As opções são todas lidas e logo em seguida repetidas, ficando o usuário sem o controle da situação, não sendo possível controlar a leitura das opções.

- *Violação à Heurística 4: Consistência e padrões*

O caso #4389 apresenta um exemplo de violação a essa heurística. A Figura 4.6 apresenta a descrição do *ticket* desse caso.

NVDA
Donate ▶

Home Connect Guides Issues Test Translate Development

#4389 closed enhancement (cantfix) Opened 3 months ago
Closed 3 months ago
Last modified 3 months ago

in input help mode, the function key is not recognized by NVDA

Reported by:	blindbhavya	Owned by:	
Priority:	minor	Milestones:	
Component:	Core	Version:	2014.2
Keywords:		Cc:	
Operating system:	Windows 7	Blocked by:	
Blockings:			
Changes document entry (for developers):			

Description

Steps to reproduce
 Ensure that the keyboard in use has the function (AKA fn) key (present on most laptops).
 Enter Input Help Mode (NVDA + 1).
 Press the fn key.
 Actual Result : NVDA stays silent, probably because it doesn't recognize the fn key like other keys on the keyboard.
 Expected Result : NVDA should say function when the function/fn key is pressed just like it does for all other keys on the keyboard.

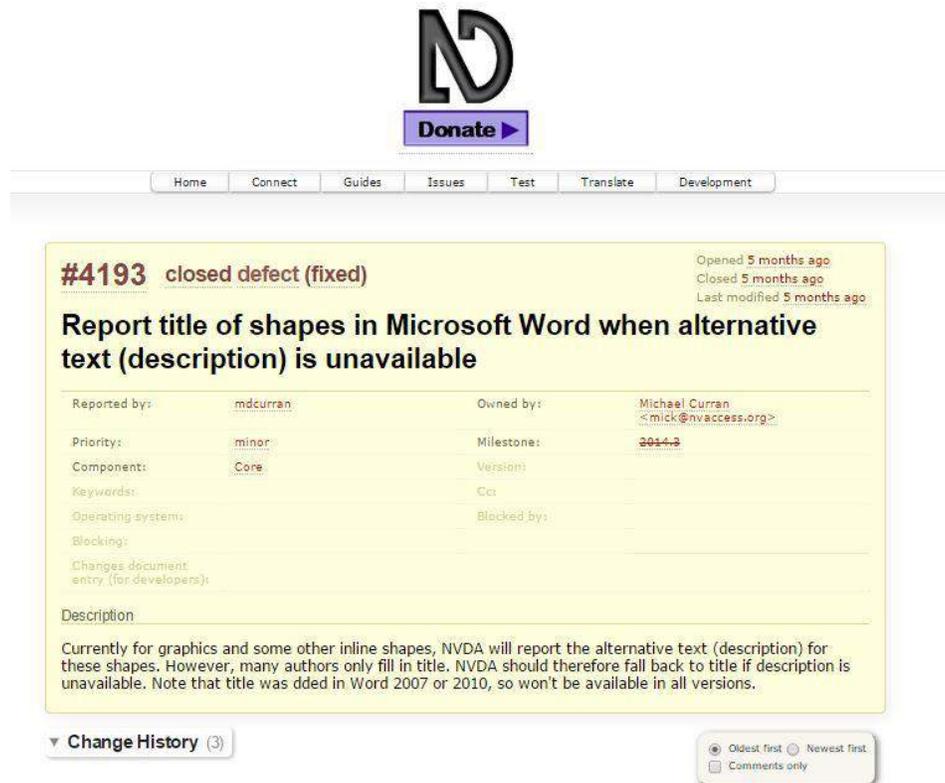
Figura 4.6 Exemplo de um *ticket* de usabilidade que viola as condições descritas pela heurística 4 de Jakob Nielsen

Fonte: NVDA Community (2014)

A violação a essa heurística pode ser notada quando o usuário informa que ao acessar o modo de entrada de ajuda, a função da tecla *fn* não é reconhecida quando se está utilizando o NVDA. Apresentando assim um problema de inconsistência e padrão, uma vez que a função desta tecla é reconhecida por vários outros *softwares*.

- *Violação à Heurística 5: Prevenção de erros*

O caso #4193 apresenta um exemplo de violação a essa heurística. A Figura 4.7 apresenta a descrição do *ticket* desse caso.



NVDA
Donate

Home Connect Guides Issues Test Translate Development

#4193 closed defect (fixed) Opened 5 months ago
Closed 5 months ago
Last modified 5 months ago

Report title of shapes in Microsoft Word when alternative text (description) is unavailable

Reported by:	mdcurren	Owned by:	Michael Curran <mick@nvaccess.org>
Priority:	minor	Milestone:	2014.3
Component:	Core	Version:	
Keywords:		Cc:	
Operating system:		Blocked by:	
Blocking:			
Changes document entry (for developers):			

Description

Currently for graphics and some other inline shapes, NVDA will report the alternative text (description) for these shapes. However, many authors only fill in title. NVDA should therefore fall back to title if description is unavailable. Note that title was added in Word 2007 or 2010, so won't be available in all versions.

▼ Change History (3)
 Oldest first Newest first
 Comments only

Figura 4.7 Exemplo de um *ticket* de usabilidade que viola as condições descritas pela heurística 5 de Jakob Nielsen

Fonte: NVDA Community (2014)

A violação a essa heurística pode ser observada quando o usuário informa que os gráficos e outras formas utilizadas no *software MS Office Word* que não possuem o atributo “*alt*” (alternativo) preenchido, não tem a sua

identificação lida. Uma solução seria o NVDA realizar a leitura do título padrão da forma, evitando assim o erro de usabilidade.

- *Violação à Heurística 6: Reconhecimento ao invés de recordação*

O caso #3828 apresenta um exemplo da violação a essa heurística. A Figura 4.8 apresenta a descrição do *ticket* desse caso.

The image shows a screenshot of a web browser displaying a ticket from the NVDA Community. At the top, there is the NVDA logo and a 'Donate' button. Below that is a navigation menu with links for Home, Connect, Guides, Issues, Test, Translate, and Development. The main content area is a yellow box containing the ticket details for #3828, which is a 'closed task (workforme)'. The title of the ticket is 'NVDA doesn't recognize abbr html tag'. The ticket was opened 9 months ago and closed 9 months ago. The reporter is 'oaron' and the owner is 'mimar'. The priority is 'minor', the component is 'Browse mode', and the version is 'next'. The description includes an HTML example: `<p><abbr title="Nonvisual Desktop Access">NVDA</abbr> is a free and open source screenreader. NVDA can't recognize meaning of an abbreviation. About the abbr tag: http://www.w3schools.com/tags/tag_abbr.asp`. Below the description, there is a 'Change History' section showing that the resolution was set to 'workforme' and the status was changed from 'new' to 'closed'. A comment from 'comment1' explains that in both Firefox and Internet Explorer, the expanded abbreviation will be reported if you report the current navigator object (desktop NVDA+numpad5, laptop NVDA+shift+o). This matches the experience for sighted users where the abbreviation is only shown if the user moves their mouse over the abbreviation; i.e. user request.

Figura 4.8 Exemplo de um *ticket* de usabilidade que viola as condições descritas pela heurística 6 de Jakob Nielsen

Fonte: NVDA Community (2014)

A violação a essa heurística ocorre quando o usuário informa que o NVDA não reconhece a *tag HTML* “*abbr*”. O desenvolvedor informa que essa ação só é reconhecida após o usuário reportar qual o objeto atual de foco e então selecionar as teclas de atalho. Porém esta informação não está visível ao usuário não sendo possível recordar os atalhos de acionamento dessa função.

- *Violação à Heurística 7: Flexibilidade e eficiência no uso*

O caso #3816 apresenta um exemplo da violação a essa heurística. A Figura 4.9 apresenta a descrição do *ticket* desse caso.

NVDA
Donate ▶

Home Connect Guides Issues Test Translate Development

#3816 closed defect (fixed) Opened 10 months ago
Closed 10 months ago

Recently introduced delay in "mouse tracking"

Reported by:	k.kolev1985	Owned by:	James Teh <jamie@nvaccess.org>
Priority:	minor	Milestone:	
Component:	Core	Version:	next
Keywords:	regression	CC:	
Operating system:	Windows 7	Blocked by:	
Blocking:	#3804		

Changes document entry (for developers):

Description

Since one of the last next snapshots (I think from the one from the 22nd of January), I've noticed that there is a delay between the time I point the mouse cursor at an object and the time when NVDA reports that pointed object. If I use an older next snapshot (eg. the one from the 17th of January), there is no such delay. The delay doesn't occur with the stable 2013.3 as well - only with the last few snapshots.

Steps to reproduce it:

1. Load NVDA 2013.3 or one of the older next snapshots (eg. the one from the 17th of January).
2. If not enabled, enable the mouse tracking feature.
3. Point the mouse cursor at an object on the screen. NVDA should report it immediately, and it does.
4. Then unload NVDA 2013.3 or the older next snapshot and load the latest next snapshot.
5. Enable the "mouse tracking", if it is not enabled.
6. Point the mouse cursor at an object on the screen. NVDA should report it immediately, but it does not - there is a noticeable delay.

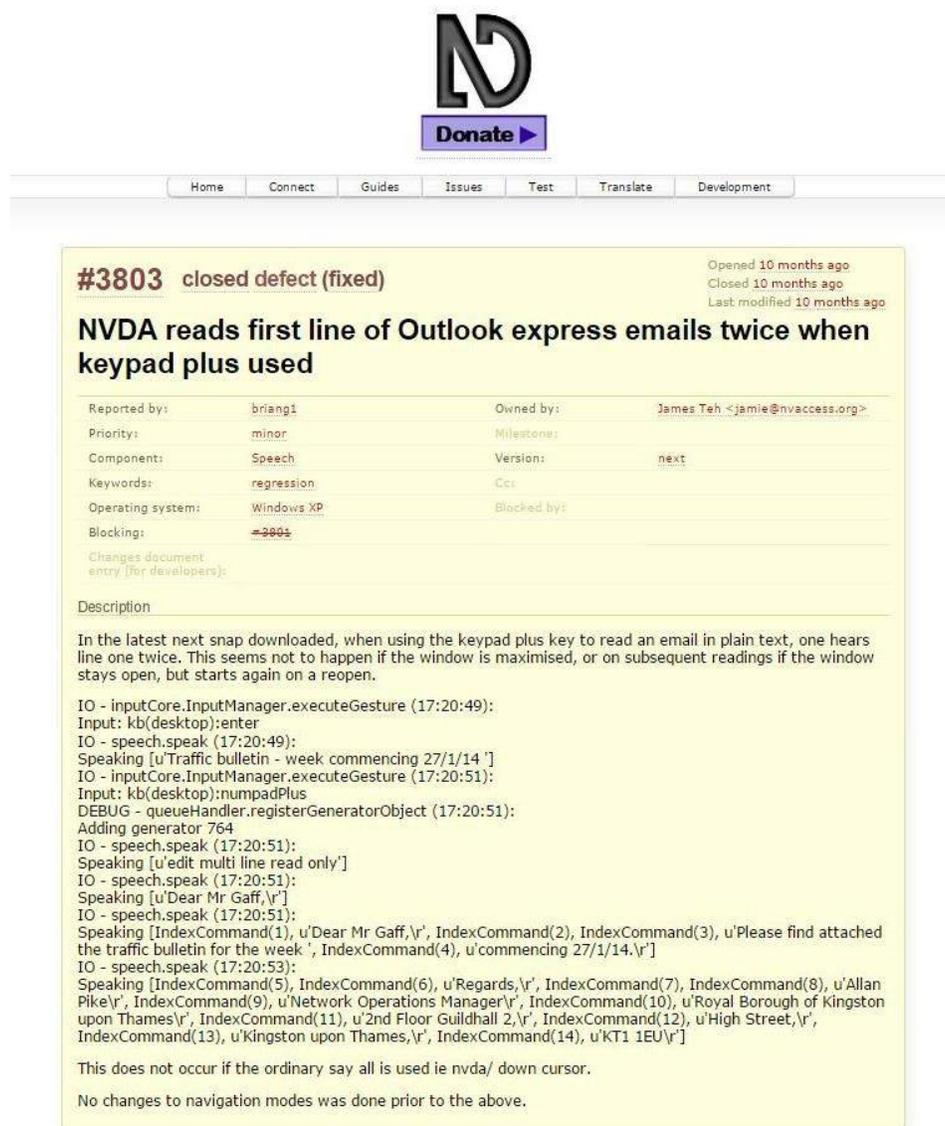
Figura 4.9 Exemplo de um *ticket* de usabilidade que viola as condições descritas pela heurística 7 de Jakob Nielsen

Fonte: NVDA Community (2014)

A violação a essa heurística ocorre quando o usuário informa que o “*mouse tracking*”, função que indica o conteúdo apontado pelo cursor do *mouse*, não apresenta um tempo de resposta (de leitura) satisfatório e não permite a agilidade no uso do sistema.

- *Violação à Heurística 8: Estética e design minimalista*

O caso #3803 apresenta um exemplo da violação a essa heurística. A Figura 4.10 apresenta a descrição do *ticket* desse caso.



NVDA
Donate ▶

Home Connect Guides Issues Test Translate Development

#3803 closed defect (fixed) Opened 10 months ago
Closed 10 months ago
Last modified 10 months ago

NVDA reads first line of Outlook express emails twice when keypad plus used

Reported by:	briang1	Owned by:	James Teh <jamie@nvaccess.org>
Priority:	minor	Milestone:	
Component:	Speech	Version:	next
Keywords:	regression	Cc:	
Operating system:	Windows XP	Blocked by:	
Blocking:	#2801		

Changes document entry (for developers):

Description

In the latest next snap downloaded, when using the keypad plus key to read an email in plain text, one hears line one twice. This seems not to happen if the window is maximised, or on subsequent readings if the window stays open, but starts again on a reopen.

```
IO - inputCore.InputManager.executeGesture (17:20:49):
Input: kb(desktop):enter
IO - speech.speak (17:20:49):
Speaking [u'Traffic bulletin - week commencing 27/1/14 ']
IO - inputCore.InputManager.executeGesture (17:20:51):
Input: kb(desktop):numpadPlus
DEBUG - queueHandler.registerGeneratorObject (17:20:51):
Adding generator 764
IO - speech.speak (17:20:51):
Speaking [u'edit multi line read only']
IO - speech.speak (17:20:51):
Speaking [u'Dear Mr Gaff,\r']
IO - speech.speak (17:20:51):
Speaking [IndexCommand(1), u'Dear Mr Gaff,\r', IndexCommand(2), IndexCommand(3), u'Please find attached the traffic bulletin for the week ', IndexCommand(4), u'commencing 27/1/14.\r']
IO - speech.speak (17:20:53):
Speaking [IndexCommand(5), IndexCommand(6), u'Regards,\r', IndexCommand(7), IndexCommand(8), u'Allan Pike\r', IndexCommand(9), u'Network Operations Manager\r', IndexCommand(10), u'Royal Borough of Kingston upon Thames\r', IndexCommand(11), u'2nd Floor Guildhall 2,\r', IndexCommand(12), u'High Street,\r', IndexCommand(13), u'Kingston upon Thames,\r', IndexCommand(14), u'KT1 1EU\r']
```

This does not occur if the ordinary say all is used ie nvda/ down cursor.

No changes to navigation modes was done prior to the above.

Figura 4.10 Exemplo de um *ticket* de usabilidade que viola as condições descritas pela heurística 8 de Jakob Nielsen

Fonte: NVDA Community (2014)

A violação a essa heurística ocorre quando o usuário informa que ao usar o teclado de leitura *plus*, as linhas de *e-mail* no *software Microsoft Office Outlook* são lidas de forma duplicada, apresentando assim uma leitura desnecessária.

- *Violação à Heurística 9: Ajudar os usuários a reconhecer, diagnosticar e recuperar-se de erros*

O caso #3827 apresenta um exemplo da violação a essa heurística. A Figura 4.11 apresenta a descrição do *ticket* desse caso.

N
Donate ▶

Home Connect Guides Issues Test Translate Development

#3827 closed defect (fixed) Opened 9 months ago
Closed 9 months ago

sometimes, there is an error when changing the active tab in Notepad++

Reported by: PZajda Owned by:
 Priority: minor Milestone:
 Component: Core Version: next
 Keywords: Cc:
 Operating system: Windows XP Blocked by:
 Blocking:

Changes document entry (for developers):

Description

Hi,

Sometimes, when I press Ctrl+Tab to change the active tab, I have the following error:
 ERROR - eventHandler.executeEvent (16:11:27):
 error executing event: gainFocus on <NVDAObjects.IAccessible.sysListView32.ListItem object at 0x03DB5EFD> with extra args of {}
 Traceback (most recent call last):

```

File "eventHandler.pyc", line 136, in executeEvent
File "eventHandler.pyc", line 84, in init
File "eventHandler.pyc", line 91, in next
File "NVDAObjects\init.pyc", line 843, in event_gainFocus
File "NVDAObjects\behaviors.pyc", line 458, in reportFocus
File "NVDAObjects\init.pyc", line 787, in reportFocus
File "speech.pyc", line 320, in speakObject
File "speech.pyc", line 238, in speakObjectProperties
File "baseObject.pyc", line 34, in get
File "baseObject.pyc", line 110, in _getPropertyViaCache
File "NVDAObjects\IAccessible\sysListView32.pyc", line 386, in _get_name
File "NVDAObjects\IAccessible\sysListView32.pyc", line 333, in _getColumnContent
  
```

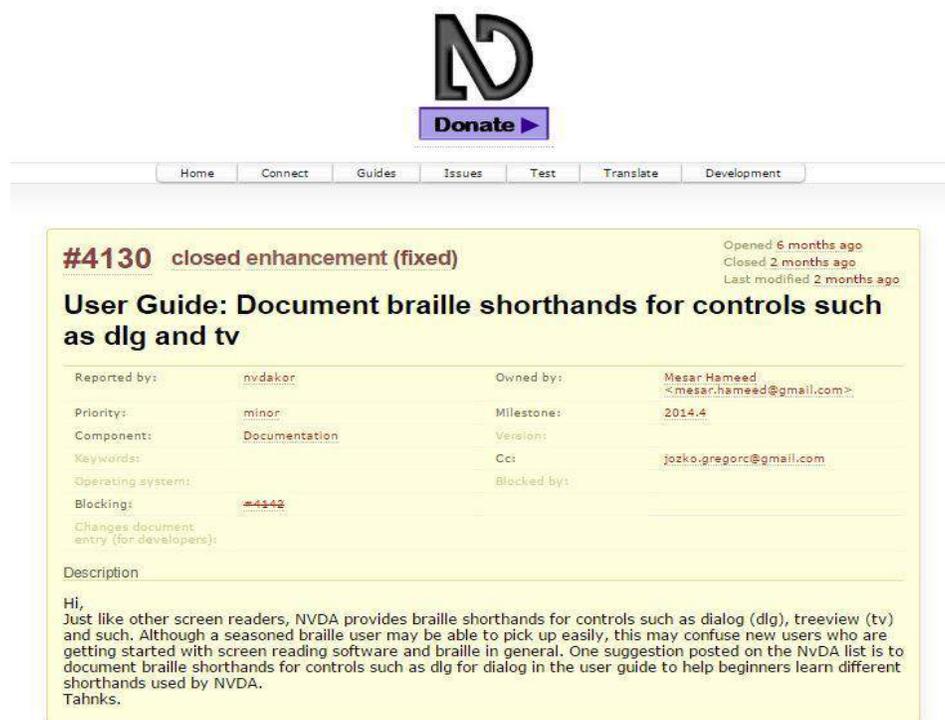
Figura 4.11 Exemplo de um *ticket* de usabilidade que viola as condições descritas pela heurística 9 de Jakob Nielsen

Fonte: NVDA Community (2014)

A violação a essa heurística ocorre após o usuário informar que um erro inesperado ocorre quando se troca de aba ativa ao utilizar o NVDA para o *software Notepad++*¹¹. Um erro técnico é apresentado ao usuário, de forma não amigável e não fornecendo passos para que possa se recuperar do problema.

- *Violação à Heurística 10: Ajuda e documentação*

O caso #4130 apresenta um exemplo da violação a essa heurística. A Figura 4.12 apresenta a descrição do *ticket* desse caso.



NVDA
Donate ▶

Home Connect Guides Issues Test Translate Development

#4130 closed enhancement (fixed) Opened 6 months ago
Closed 2 months ago
Last modified 2 months ago

User Guide: Document braille shorthands for controls such as dlg and tv

Reported by:	nvdakor	Owned by:	Mesar Hameed < mesar.hameed@gmail.com >
Priority:	minor	Milestone:	2014.4
Component:	Documentation	Version:	
Keywords:		Cc:	jozko.gregorc@gmail.com
Operating system:		Blocked by:	
Blocking:	#4142		
Changes document entry (for developers):			

Description

Hi,
Just like other screen readers, NVDA provides braille shorthands for controls such as dialog (dlg), treeview (tv) and such. Although a seasoned braille user may be able to pick up easily, this may confuse new users who are getting started with screen reading software and braille in general. One suggestion posted on the NVDA list is to document braille shorthands for controls such as dlg for dialog in the user guide to help beginners learn different shorthands used by NVDA.
Tahnks.

Figura 4.12 Exemplo de um *ticket* de usabilidade que viola as condições descritas pela heurística 10 de Jakob Nielsen

Fonte: NVDA Community (2014)

¹¹ Disponível online em <http://notepad-plus-plus.org/>

A violação a essa heurística ocorre quando o usuário informa que funções de estenografias suportadas pelos leitores de telas, inclusive o NVDA, não estão documentadas e assim um usuário pode não ter acesso a essa função caso tenha dúvida, uma vez que não consta na documentação de ajuda.

4.1.4. Tratamento dos casos pelos desenvolvedores

Em relação ao tratamento que os casos relatados pelos *tickets* componentes do *data-set* receberam pelos desenvolvedores, foi possível obter resultados amplos. Estes resultados estão expostos na Tabela 5.

Tabela 5 Relação dos tickets de usabilidade e funcionalidade quanto ao motivo de fechamento dos mesmos

Categoria	Fixed		Wontfix		Worksforme		Cantfix		Total
	Nº	%	Nº	%	Nº	%	Nº	%	
Usabilidade	93	62,41*	25	16,77*	22	14,76*	9	6,04*	149
Funcionalidade	34	66,66*	8	15,68*	7	13,72*	2	3,92*	51
Total	127	63,50	33	16,50	29	14,50	11	5,50	200

* Porcentagem em relação ao total da categoria específica

A Tabela 5 apresenta a relação de *tickets* de usabilidade e funcionalidade quanto aos motivos de fechamento dos mesmos. É apresentado para cada categoria o número e a porcentagem relativa ao motivo de fechamento do caso.

Pode-se observar que, do total de casos analisados no *data-set*, 63,50% dos *tickets* tiveram o seu relato corrigido/implementado. Também é possível notar que os casos da categoria “Funcionalidade” apresentaram uma porcentagem ligeiramente maior de correção em relação aos casos da categoria de “Usabilidade” (66,66% e 62,41%, respectivamente).

Entretanto, ao realizar um teste de Chi-Square, não foi encontrada nenhuma diferença significativa entre a distribuição de problemas nas categorias de correção quando comparados os problemas de usabilidade e funcionalidade (X^2 , p-valor = 0,92). Este teste foi utilizado por ser um teste estatístico comumente utilizado para comparar dados observados com os dados esperados em uma pesquisa, para obtenção de um acordo com uma hipótese específica. Mais detalhes sobre esse teste, consultar Plackett (1983).

4.2. Dificuldades encontradas

As dificuldades encontradas no desenvolvimento desse trabalho concentraram-se, em sua maior parte, na etapa das análises dos dados. Como cada *ticket* do conjunto de *data-set* foi analisado individualmente, muitas vezes foi difícil compreender o que o usuário queria informar. Para estes casos, a discussão ocorrida entre usuários do sistema e desenvolvedores assegurou uma melhor compreensão ou até o seu correto entendimento.

Outro ponto de dificuldade ocorreu na fase de codificação dos *tickets* para suas respectivas categorizações. Algumas vezes houve conflitos entre as codificações apresentadas pelos pesquisadores codificadores. Muitas vezes um *ticket* apresentava “termos” ou “sentenças” que remetiam tanto à categoria de “Usabilidade” quando “Funcionalidade”, uma vez que as diferenças entre tais definições são separadas por uma linha tênue. Todavia, tais conflitos foram sanados através da verificação da discussão documentada por membros da comunidade de usuários acerca do referido *ticket* e também por argumentações trocadas entre os codificadores, até que houvesse um consenso e acordo para a categorização correta de cada *ticket*.

Caso semelhante também ocorreu durante a classificação das heurísticas de Nielsen que foram violadas pelos *tickets* categorizados em “Usabilidade”. Da

mesma forma, argumentações e discussões permitiram que os conflitos fossem resolvidos para que a classificação ocorresse com sucesso.

Por fim, a identificação das pessoas envolvidas no processo de cadastro e correção dos *tickets* mostrou-se uma tarefa complicada, uma vez que a plataforma de rastreamento dos casos não apresentava explicitamente se um determinado usuário da comunidade possuía somente perfil de usuário final ou também de usuário e desenvolvedor. Essa dificuldade foi superada pela identificação dos usuários através dos textos dos seus relatos e também através das discussões que envolviam o escopo do *ticket*. Uma outra forma que permitiu identificar o perfil do usuário desenvolvedor foi através da informação *owner*, inerente a alguns casos relatados. Complementarmente também foi verificado o registro de *commits* nos códigos do NVDA, visando a identificação de desenvolvedores.

5. CONCLUSÃO

A conclusão deste trabalho permitiu inicialmente a realização de uma breve revisão da literatura sobre os processos que envolvem o desenvolvimento de *softwares* livres. Também foi possível revisar a literatura que aborda questões de usabilidade, mais especificadamente o contexto de usabilidade no processo de desenvolvimento de um projeto de *software* livre.

Também foi apresentada uma descrição detalhada acerca do *software* leitor de telas livre NVDA, tais como suas características, modo de funcionamento e abordagem à sua plataforma global de comunidade de usuários (tanto utilizadores do leitor de telas, quanto os seus desenvolvedores). Também foi apresentada e contextualizada a ferramenta de rastreamento de *bugs/melhorias* do NVDA, inserida na própria plataforma de comunidade de usuários do projeto *NV Access*. Foi demonstrado como um usuário pode cadastrar *ticket (issue)* e como pesquisar por *tickets* já existentes. Também foi demonstrado o processo básico de correção destes *tickets* e envolvimento dos usuários e desenvolvedores em tal processo.

A partir desta ferramenta de rastreamento de *bugs/melhorias* do NVDA, foi possível definir o conjunto de dados (*data-set*) a ser analisado e em seguida extrair os resultados da pesquisa.

Este trabalho teve como proposta apresentar um estudo do processo de desenvolvimento do *software* livre NVDA a fim de encontrar problemas envolvendo funcionalidade e usabilidade na utilização desse leitor de telas.

Através dos resultados, questões críticas/prioritárias envolvendo a usabilidade do NVDA foram mapeadas após análises dos *tickets* investigados no sistema de rastreamento de *bugs* e melhorias.

Para estes resultados, foi possível verificar que a maior parte dos *tickets* analisados no estudo abordavam problemas diretamente relacionados com a usabilidade do sistema no que se refere à satisfação e conforto do usuário na experiência de uso do NVDA.

Foi possível analisar tais casos e verificar que em sua maioria apresentavam questões críticas de usabilidade, envolvendo mais precisamente o *feedback* e visualização do *status* do sistema, a flexibilidade e agilidade no uso do sistema e questões de consistências e padrões de uso, com base nas heurísticas de Nielsen que mais foram violadas.

Pode-se concluir que o fator *feedback* e visualização de *status* do sistema é um ponto crítico de usabilidade do NVDA, uma vez que, por se tratar de um *software* cujo uso é destinado a pessoas com deficiências visuais, caso o sistema não informe perfeitamente “onde o usuário está” e “o que está ocorrendo ao seu redor”, o uso do sistema estará seriamente prejudicado e a experiência do usuário não será boa.

O fator flexibilidade e agilidade no uso também foi reconhecido como fator crítico, uma vez que grande parte dos usuários desse tipo de sistema sentem uma necessidade de agilidade no processo de utilização do *software*. E assim gerando satisfação no seu uso.

Já o fator de consistência e padrões na utilização do NVDA foi considerado outro ponto crítico pelo fato de que os usuários tendem a memorizar padrões para facilitar o uso do sistema. Uma vez que esses padrões não são consistentes, a experiência no uso se torna prejudicada.

Pelas classificações dos *tickets* de usabilidade quanto às heurísticas de Nielsen, foi possível constatar que se em todos os casos fossem aplicadas as diretrizes propostas pela heurística principal, inerente ao *ticket*, já seria solucionado quase 60% dos problemas de usabilidades levantados. Quando se

incluem as heurísticas complementares, a solução para os problemas atinge a totalidade dos casos (100%).

Apesar da percepção da baixa prioridade de problemas de usabilidade para esse tipo de projeto de *software*, constatado por Andreasen et. al (2006) e Livari (2008), não foram encontradas diferenças significativas no tratamento de problemas de usabilidade e funcionalidade no projeto analisado.

As conclusões obtidas por esse trabalho permitirão que organizações de Tecnologia da Informação (TI), cujo foco de negócio seja o desenvolvimento de *softwares* livres utilizados como recursos de Tecnologia Assistiva por pessoas com deficiência visual, conheçam quais os principais fatores de usabilidade devem ser priorizados no processo de desenvolvimento desses *softwares*, a fim de se obter um produto com alta qualidade e aceitação de seus usuários.

Além disso, tais considerações permitem que os fatores críticos de usabilidade sejam focados no processo de desenvolvimento do sistema NVDA para alavancar a qualidade do produto de forma a atrair cada vez mais usuário.

5.1. Trabalhos futuros

Como sugestão para trabalhos futuros, considero interessante realizar uma análise completa de todo o processo de desenvolvimento do NVDA, desde a fase de levantamento de requisitos até a fase de liberação de novas *releases* de versões aos seus usuários. Essa análise possibilitaria compreender esse processo e facilitaria a identificação de onde os fatores de usabilidade críticos levantados pelo trabalho poderiam ser trabalhados, melhorando o produto final.

REFERÊNCIAS BIBLIOGRÁFICAS

BALDWIN, C; CLARK, K. **The Architecture of Participation: Does Code Architecture Mitigate Free Riding in the Open Source Development Model?** *Manage. Sci.* 52, 7 (2006), 1127, 1116.

BERSCH, R. **Introdução à Tecnologia Assistiva**. Disponível online em <<http://www.assistiva.com.br/Introducao%20TA%20Rita%20Bersch.pdf>> 2008. Acesso em 16 de junho de 2013.

BRASIL. Subsecretaria Nacional de Promoção dos Direitos da Pessoa com Deficiência. Comitê de Ajudas Técnicas. **Tecnologia Assistiva**. – Brasília: CORDE, 2009. 138 p.

COOK, A. M.; POLGAR, J. M. **Assistive Technologies: Principles and Practices**. Mosby – Year Book, Inc., 1995.

DESPALATOVIĆ, L. **The Usability of Free/Libre/Open Source Projects: A Review**. *International Journal of Computer and Information Technology* (ISSN: 2279 – 0764), vol. 02, n. 5, September 2013. pp. 938-963.

ESPEAK PROJECT. **eSpeak text to speech**. Disponível online em <<http://espeak.sourceforge.net/>>. Acesso em 22 de outubro de 2014.

FLICK, U. **Introdução à Pesquisa Qualitativa**. Bookman editora, 3a edição, 2009.

FRISHBERG, N.; DIRKS, A. M.; BENSON, C.; NICKELL, S.; SMITH, S. **Getting to know you: open source development meets usability.** In CHI '02: Extended Abstracts on Human Factors in Computing Systems. Minneapolis, MI. ACM Press, April, 2002. New York: ACM. pp. 932-933.

GIT. **Documentation.** Disponível online em <<http://git-scm.com/docs/git-commit>>. Acessado em 10 de Setembro de 2014.

GNU PROJECT – GNU – **A Definição de Software Livre.** Disponível online em <<http://www.gnu.org/philosophy/free-sw.html>>. Acessado em 13 de Agosto de 2013.

GNU PROJECT – GNU – **Licenças de Software Livre.** Disponível online em <<http://www.gnu.org/licenses/licenses.html>>. Acessado em 13 de Agosto de 2013.

GNU PROJECT – GNU – **O manifesto GNU.** Disponível online em <<http://www.gnu.org/gnu/manifesto.html>>. Acessado em 13 de Agosto de 2013.

GUTWIN, C.; PENNER, R.; SCHNEIDER, K. **Group awareness in distributed software development.** In CSCW '04: Proceedings of the 2004 ACM conference on Computer supported cooperative work. ACM Press, New York, NY, 2004. pp. 72-81.

HARS, A.; OU, S. **Working for free? Motivations for participating in open-source projects**. 2002. *International Journal of Electronic Commerce*, 6(3), 25–39.

HEDBERG, H.; IIVARI, N. **Integrating HCI Specialists into Open Source Software Development Projects**. In *Open Source Ecosystems: Diverse Communities Interacting*. Ed. C. Boldyreff et al.: OSS 2009, IFIP AICT 299, 2009. pp. 251-263.

HOLSTI, O. R. **Content Analysis for the Social Sciences and Humanities**. Ed. Addison-Wesley Publishing Company. Reading, MA, 1969. pp. 24-41.

IIVARI, N. **Usability in Open Source Software Development: An Interpretive Case Study**. ECIS 2008 Proceedings. 2008. Paper 42. Disponível online em < <http://aisel.aisnet.org/ecis2008/42>>. Acessado em 24 de Outubro de 2014.

INTERNATIONAL STANDARDS ORGANIZATION – ISO. **ISO 9241-11 - Ergonomic requirements for office work with visual display terminals (VDTs). Part 11 - Guidelines for specifying and measuring usability**. Genève: International Organisation for Standardisation. 1998.

INTERNATIONAL STANDARDS ORGANIZATION – ISO. **ISO 9241-171: Ergonomics of human-system interaction. Part 171: Guidance on software accessibility**. 2008.

LAZAR, J., FENG, J.H., & HOCHHEISER, H. **Research Methods in Human-Computer Interaction**. John Wiley & Sons Ltd., Chichester, UK. 2010.

LAKHANI, K.; VON HIPPEL, E. **How open source software works: Free user-to-user assistance**. 2003. *Research Policy*, 32(6), 923–943.

MACHADO, I. M. R.; PEREIRA, L. M. **Avaliação de Processo de Desenvolvimento de Software Livre - Estudo de Caso EAD Livre**. Simpósio Mineiro de Sistemas de Informação (3º SMSI). Minas Gerais: PUC Minas/Instituto de Informática, 2006.

MEYERS, M. **Qualitative research in business and management**. Los Angeles. Sage Publications. 2009.

MOCKUS, A.; FIELDING, R.T.; HERBSLEB, J. **A case study of open source software development: the Apache server**. In ICSE '00: 22nd International Conference on Software Engineering. Limerick, Ireland, June 4-1, Article 9. ACM. 2000. pp.263–272.

MOLICH, R.; NIELSEN, J. **Improving a human-computer dialogue**. *Communications of the ACM* 33,3 (Março), 338-348. 1990.

NIELSEN, J. **Enhancing the explanatory power of usability heuristics**. In CHI'94: SIGCHI Conference on Human Factors in Computing Systems. Boston, MA, April 24-28. ACM. 1994a. pp. 152-158.

NIELSEN, J. **Heuristic evaluation**. In NIELSEN, J.; MACK, R.L. (Eds.); **Usability Inspection Methods**. John Wiley & Sons, New York, NY. 1994b.

NIELSEN, J.; MOLICH, R. **Heuristic evaluation of user interfaces**. Proc. ACM CHI'90: SIGCHI Conference on Human Factors in Computing Systems. Seattle, WA, 1-5 April, 1990. pp. 249-256.

NIELSEN, J. **Usability Engineering**. 1ª ed. Boston: Academic Press, 1993. vol. 1. ISBN 0-12-518405-0.

NAPNE/RENAPI – IFRS CAMPUS BENTO GONÇALVES. **Manual Leitor de Tela NVDA**. IFRS. Bento Gonçalves. 2011.

NONVISUAL DESKTOP ACCESS – NVDA – **NVDA Project**. Disponível online em <<http://www.nvaccess.org/>>. Acesso em 16 de junho de 2013.

NVDA COMMUNITY – **Wiki: Issues**. Disponível online em <<http://community.nvda-project.org/wiki/Issues>>. Acesso em 25 de outubro de 2014.

OPEN SOURCE INICIATIVE – OSI. **The Open Source Definition**. Disponível online em <<http://opensource.org/osd>>. Acesso em 13 de agosto de 2013.

OPEN SOURCE INICIATIVE – OSI. **The BSD 2 - Clause License**. Disponível online em <<http://opensource.org/licenses/bsd-license.php>>. Acesso em 29 de outubro de 2014.

OREG, S.; NOV, O. **Exploring motivations for contributing to open source initiatives: The roles of contribution context and personal values.** *Comput. Hum. Behav.* 24, 5 (2008), 2055–2073.

PLACKETT, R, L. **Karl Pearson and the Chi-squared test.** *In* 51st International Statistical Review, Longman Group Limited, UK. 1983, pp. 59-72.

PETRIE, H.; BEVAN, N. **The evaluation of accessibility, usability and user experience.** *In* The Universal Access Handbook. Ed. Constantine Stephanidis, CRC Press, 2009, pp. 10-20.

PRESSMAN, R. S. **Engenharia de Software: Uma abordagem profissional.** 7ª ed. Bookman. 2011. p. 40.

REIS, C. R. **Caracterização de um Modelo de Processo para Projetos de Software Livre.** Dissertação de mestrado. Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, 2003.

SCHWARTZ, J. I. **Construction of software. In: Practical Strategies for Developing Large Systems.** Menlo Park: Addison-Wesley, 1st. ed., 1975.

SOMMERVILLE, I. **Software Engineering: International Edition.** 9th. ed. University of St Andrews, Scotland. Publicado por Pearson Higher Ed USA, 2010. ISBN 978-013-705346-9. p. 28.

STALLMAN, R. **The GNU Operating System and the Free Software Movement.** In *Open Sources: Voices from the Open Source Revolution.* O'Reilly & Associates Inc, 1999. Ed. 1. p. 280.

STEMLER, S. **An overview of content analysis.** Practical assessment, research & evaluation 7.17 (2001): 137-146.

TAURION, C. **Software Livre: Potencialidades e Modelos de Negócios.** Rio de Janeiro, Brasil: Brasport. 1ª Ed. 2004. p. 15.

TERRY, M.; KAY, M.; LAFRENIERE, B. **Perceptions and Practices of Usability in the Free/Open Source Software (FOSS) Community.** In CHI'10: IGCHI Conference on Human Factors in Computing Systems. April 10–15, 2010, Atlanta, GA, USA.

TINSLEY, H. E. A., WEISS, D. J. **Interrater reliability and agreement of subjective judgements.** 1975. *Journal of Counseling Psychology*, 22, 358-376.

TRAC. **Integrated SCM & Project Management.** Disponível online em <<http://trac.edgwall.org/>>. Acesso em 29 de outubro de 2014.

TWIDALE, M.B.; NICHOLS, D.M., **Exploring Usability Discussions in Open Source Development.** *System Sciences*, 2005. HICSS '05: 38th Annual Hawaii International Conference, 03-06 Jan. 2005. 198 p. doi: 10.1109/HICSS.2005.266

ULIANA, C.C. **NVDA – Software Livre - Leitor de Tela para Windows**. Disponível online em <<http://www.bengalalegal.com/nvda>> 2008. Acessado em 25 de junho de 2013.

W3C BRASIL – ESCRITÓRIO BRASILEIRO DO WORLD WIDE WEB CONSORTIUM. **Pesquisa sobre uso de Tecnologias Assistivas: Ampliadores e leitores de tela**. 2013. Disponível online em <<http://accessibilidade.w3c.br/pesquisa/resultados-preliminares/>>. Acesso em 30 de março de 2013.

WAINER, J. **Métodos de pesquisa quantitativa e qualitativa para a ciência computação**. In T. KOWALTOWSKI e K. BREITMAN (Org.), Atualização em Informática da Sociedade Brasileira de Computação. 2007. pp. 221-262.

WEB ACCESSIBILITY IN MIND – WEBAIM. **Screen Reader User Survey #4 Results**. 2012. Disponível online em <<http://webaim.org/projects/screenreadersurvey4/>>. Acesso em 30 de março de 2013.

APÊNDICE A: PLANÍLHA DO DATA-SET DE TICKETS UTILIZADOS PARA ANÁLISE

Nº	ID Ticket	Categoria	Tipo	Owner	Autor	Motivo	HN. Princ.	HN. Comp.	Q1	Q2
1	4445	Usabilidade	Defect	mdcurran	xonic	worksforme	1	**	Sim	Sim
2	4428	Usabilidade	Defect	-	santosh	worksforme	3	8	Não	Sim
3	4426	Usabilidade	Enhancement	mesar hameed	mhameed	fixed	1	7	Sim	Sim
4	4425	Usabilidade	Defect	jteh	vgjh2005	worksforme	7	4	Não	Sim
5	4423	Usabilidade	Defect	michael curran	mdcurran	fixed	1	**	Sim	Sim
6	4418	Funcionalidade	Defect	michael curran	mdcurran	fixed	**	**	**	**
7	4411	Usabilidade	Defect	-	sumandogra	wontfix	7	1	Sim	Sim
8	4409	Funcionalidade	Defect	mdcurran	andrewd	fixed	**	**	**	**
9	4407	Usabilidade	Defect	-	marcozehe	wontfix	1	7	Sim	Sim
10	4406	Usabilidade	Enhancement	jteh	kattygeltmeyer	wontfix	10	9	Sim	Sim
11	4404	Usabilidade	Defect	-	haltersweb	wontfix	8	3	Sim	Sim
12	4403	Usabilidade	Enhancement	jteh	blindbhavya	worksforme	4	2	Sim	Sim
13	4392	Usabilidade	Enhancement	-	blindbhavya	wontfix	1	2	Sim	Sim
14	4389	Usabilidade	Enhancement	-	blindbhavya	cantfix	4	7	Não	Sim
15	4383	Funcionalidade	Defect	michael curran	mdcurran	fixed	**	**	**	**
16	4375	Usabilidade	Defect	mdcurran	mk360	fixed	5	9	Não	Sim
17	4374	Funcionalidade	Enhancement	michael curran	jteh	fixed	**	**	**	**

Nº	ID Ticket	Categoria	Tipo	Owner	Autor	Motivo	HN. Princ.	HN. Comp.	Q1	Q2
18	4370	Usabilidade	Defect	mdcurran	jteh	fixed	7	5	Não	Sim
19	4369	Usabilidade	Defect	mdcurran	jteh	fixed	1	7, 9	Não	Sim
20	4368	Funcionalidade	Task	jteh	jteh	fixed	**	**	**	**
21	4367	Usabilidade	Enhancement	-	blindbhavya	worksforme	7	10, 6	Não	Sim
22	4366	Funcionalidade	Enhancement	-	ashikhirani	wontfix	**	**	**	**
23	4363	Usabilidade	Enhancement	-	blindbhavya	wontfix	7	**	Sim	Sim
24	4361	Usabilidade	Enhancement	-	blindbhavya	cantfix	7	**	Sim	Sim
25	4358	Funcionalidade	Enhancement	-	blindbhavya	wontfix	**	**	**	**
26	4356	Usabilidade	Task	-	blindbhavya	wontfix	8	**	Sim	Sim
27	4352	Funcionalidade	Enhancement	-	blindbhavya	wontfix	**	**	**	**
28	4350	Usabilidade	Defect	mdcurran	nishimotz	fixed	8	2	Não	Sim
29	4343	Usabilidade	Defect	-	khsruru	cantfix	7	2, 3	Não	Sim
30	4342	Usabilidade	Defect	-	aleskis	wontfix	3	5	Não	Sim
31	4338	Usabilidade	Defect	-	blindbhavya	cantfix	3	7	Sim	Sim
32	4332	Usabilidade	Enhancement	-	blindbhavya	wontfix	1	5	Sim	Sim
33	4329	Funcionalidade	Defect	-	finnbox64	worksforme	**	**	**	**
34	4326	Usabilidade	Enhancement	-	blindbhavya	wontfix	7	**	Sim	Sim
35	4324	Usabilidade	Enhancement	-	jorgtum	wontfix	1	**	Sim	Sim
36	4316	Usabilidade	Defect	-	blindbhavya	wontfix	1	2, 4	Não	Sim

Nº	ID Ticket	Categoria	Tipo	Owner	Autor	Motivo	HN. Princ.	HN. Comp.	Q1	Q2
37	4313	Funcionalidade	Defect	-	blindbhavya	worksforme	**	**	**	**
38	4312	Usabilidade	Enhancement	-	blindbhavya	wontfix	7	**	Sim	Sim
39	4306	Usabilidade	Defect	-	ateu	fixed	4	**	Sim	Sim
40	4305	Usabilidade	Defect	mdcurran	nvdakor	fixed	1	4	Não	Sim
41	4297	Usabilidade	Defect	-	shadyar	worksforme	2	1	Não	Sim
42	4295	Usabilidade	Defect	james teh	dgoldfield	fixed	1	7	Sim	Sim
43	4292	Usabilidade	Enhancement	-	blindbhavya	worksforme	1	**	Sim	Sim
44	4290	Usabilidade	Defect	-	ateu	worksforme	1	**	Sim	Sim
45	4282	Usabilidade	Enhancement	-	blindbhavya	wontfix	7	8, 2	Não	Sim
46	4280	Usabilidade	Defect	-	blindbhavya	wontfix	7	8	Não	Sim
47	4268	Usabilidade	Defect	-	blindbhavya	worksforme	7	8	Não	Sim
48	4267	Funcionalidade	Defect	-	blindbhavya	worksforme	**	**	**	**
49	4264	Funcionalidade	Defect	michael curran	jteh	fixed	**	**	**	**
50	4262	Funcionalidade	Defect	-	rnickelson	fixed	**	**	**	**
51	4257	Usabilidade	Defect	-	nvdakor	fixed	1	**	Sim	Sim
52	4256	Usabilidade	Enhancement	-	khsruru	wontfix	1	**	Sim	Sim
53	4255	Usabilidade	Task	jteh	ateu	fixed	7	1, 8	Não	Sim
54	4254	Usabilidade	Defect	jteh	siddu	fixed	3	2	Não	Sim
55	4249	Usabilidade	Enhancement	-	dhankuta	wontfix	7	2	Sim	Sim

Nº	ID Ticket	Categoria	Tipo	Owner	Autor	Motivo	HN. Princ.	HN. Comp.	Q1	Q2
56	4244	Funcionalidade	Defect	jteh	jteh	fixed	**	**	**	**
57	4241	Usabilidade	Defect	-	vijaynayar	wontfix	7	2, 4	Não	Sim
58	4240	Funcionalidade	Defect	michael curran	mdcurran	fixed	**	**	**	**
59	4234	Usabilidade	Enhancement	-	aslani.ali	worksforme	7	2,4	Sim	Sim
60	4232	Usabilidade	Defect	mdcurran	levilibegas	fixed	9	5,4	Não	Sim
61	4230	Usabilidade	Enhancement	-	mikeray	wontfix	4	2	Sim	Sim
62	4226	Usabilidade	Defect	michael curran	mdcurran	fixed	7	4	Sim	Sim
63	4225	Funcionalidade	Enhancement	jteh	jteh	fixed	**	**	**	**
64	4223	Usabilidade	Defect	jteh	jteh	fixed	1	8	Sim	Sim
65	4222	Usabilidade	Defect	jteh	jteh	fixed	7	4	Sim	Sim
66	4219	Funcionalidade	Enhancement	michael curran	mdcurran	fixed	**	**	**	**
67	4215	Usabilidade	Defect	-	twinsen	cantfix	4	2	Não	Sim
68	4214	Usabilidade	Defect	michael curran	mdcurran	fixed	1	7	Sim	Sim
69	4213	Usabilidade	Defect	michael curran	mdcurran	fixed	4	2	Sim	Sim
70	4212	Usabilidade	Defect	michael curran	mdcurran	fixed	8	2	Sim	Sim
71	4211	Funcionalidade	Defect	michael curran	mdcurran	fixed	**	**	**	**
72	4203	Usabilidade	Defect	michael curran	mdcurran	fixed	4	**	Sim	Sim
73	4199	Usabilidade	Defect	michael curran	mdcurran	fixed	7	5	Não	Sim
74	4198	Usabilidade	Defect	michael curran	mdcurran	fixed	8	7	Não	Sim

Nº	ID Ticket	Categoria	Tipo	Owner	Autor	Motivo	HN. Princ.	HN. Comp.	Q1	Q2
75	4193	Usabilidade	Defect	michael curran	mdcurran	fixed	5	2	Não	Sim
76	4185	Usabilidade	Defect	michael curran	mdcurran	fixed	1	7	Sim	Sim
77	4184	Funcionalidade	Defect	-	ahiiron	fixed	**	**	**	**
78	4179	Usabilidade	Defect	mdcurran	nishimotz	fixed	9	5	Sim	Sim
79	4170	Funcionalidade	Defect	jteh	jteh	fixed	**	**	**	**
80	4169	Usabilidade	Defect	mdcurran	marcozehe	fixed	1	5	Não	Sim
81	4166	Usabilidade	Enhancement	-	kredh	wontfix	7	10, 2	Não	Sim
82	4165	Usabilidade	Defect	mdcurran	mdcurran	fixed	1	7	Sim	Sim
83	4157	Usabilidade	Defect	-	motter	fixed	4	**	Sim	Sim
84	4152	Usabilidade	Defect	mdcurran	sumandogra	fixed	1	2	Sim	Sim
85	4151	Funcionalidade	Defect	mdcurran	sumandogra	fixed	**	**	**	**
86	4149	Usabilidade	Defect	-	the_dark_man	fixed	1	5	Não	Sim
87	4148	Funcionalidade	Enhancement	-	surfer0627	worksforme	**	**	**	**
88	4140	Funcionalidade	Defect	-	rnickelson	fixed	**	**	**	**
89	4137	Usabilidade	Defect	-	blindbhavya	wontfix	8	4	Sim	Sim
90	4134	Usabilidade	Defect	michael curran	the_dark_man	fixed	7	**	Sim	Sim
91	4130	Usabilidade	Enhancement	mesar hameed	nvdakor	fixed	10	7	Sim	Sim
92	4129	Funcionalidade	Defect	jteh	jteh	fixed	**	**	**	**
93	4128	Funcionalidade	Defect	jteh	jteh	fixed	**	**	**	**

Nº	ID Ticket	Categoria	Tipo	Owner	Autor	Motivo	HN. Princ.	HN. Comp.	Q1	Q2
94	4125	Funcionalidade	Defect	mdcurran	nishimotz	fixed	**	**	**	**
95	4122	Usabilidade	Task	jteh	driemer.riemer	cantfix	8	7	Sim	Sim
96	4113	Usabilidade	Defect	jteh	msuch	cantfix	10	5	Sim	Sim
97	4106	Usabilidade	Defect	jteh	jteh	fixed	1	7	Sim	Sim
98	4104	Usabilidade	Defect	michael curran	mdcurran	fixed	7	4	Sim	Sim
99	4103	Funcionalidade	Defect	jteh	jteh	fixed	**	**	**	**
100	4098	Funcionalidade	Defect	Jame teh	rnickelson	fixed	7	**	Sim	Sim
101	4096	Funcionalidade	Defect	michael curran	mdcurran	fixed	**	**	**	**
102	4095	Funcionalidade	Defect	michael curran	mdcurran	fixed	**	**	**	**
103	4092	Usabilidade	Defect	jteh	marcozehe	fixed	1	7	Sim	Sim
104	4091	Usabilidade	Defect	michael curran	mdcurran	fixed	1	8	Sim	Sim
105	4090	Funcionalidade	Defect	michael curran	mdcurran	fixed	**	**	**	**
106	4084	Usabilidade	Defect	-	mhorspool	worksforme	9	5	Sim	Sim
107	4082	Funcionalidade	Defect	mhameed	mhorspool	wontfix	**	**	**	**
108	4069	Usabilidade	Enhancement	michael curran	mdcurran	fixed	7	1, 4	Sim	Sim
109	4068	Usabilidade	Defect	-	msuch	fixed	7	4	Sim	Sim
110	4067	Usabilidade	Defect	-	vgjh2005	fixed	7	3	Sim	Sim
111	4066	Usabilidade	Defect	james teh	digitaltoast	fixed	8	4, 2	Não	Sim
112	4061	Funcionalidade	Enhancement	-	mohammed	wontfix	**	**	**	**

Nº	ID Ticket	Categoria	Tipo	Owner	Autor	Motivo	HN. Princ.	HN. Comp.	Q1	Q2
113	4059	Funcionalidade	Defect	-	vgjh2005	cantfix	**	**	**	**
114	4058	Funcionalidade	Defect	mdcurran	marcozehe	fixed	**	**	**	**
115	4057	Usabilidade	Enhancement	james teh	zahari_bgr	fixed	7	8	Não	Sim
116	4056	Usabilidade	Enhancement	jteh	nvdakor	fixed	1	8	Não	Sim
117	4051	Funcionalidade	Enhancement	-	animesh.mr	wontfix	**	**	**	**
118	4044	Usabilidade	Defect	james teh	nvdakor	fixed	1	9	Não	Sim
119	4043	Usabilidade	Defect	michael curran	mdcurran	fixed	4	2, 7	Não	Sim
120	4038	Usabilidade	Enhancement	-	ashleycox	worksforme	1	7	Não	Sim
121	4034	Funcionalidade	Enhancement	jteh	zahari_bgr	wontfix	**	**	**	**
122	4031	Usabilidade	Defect	michael curran	rnickelson	fixed	1	7	Sim	Sim
123	4030	Usabilidade	Task	jteh	vgjh2005	wontfix	7	2	Sim	Sim
124	4013	Usabilidade	Enhancement	-	supanut2000	worksforme	10	8	Não	Sim
125	4000	Funcionalidade	Defect	-	k_kolev1985	fixed	**	**	**	**
126	3991	Usabilidade	Defect	jteh	erion	fixed	1	**	Sim	Sim
127	3965	Usabilidade	Defect	-	jorgtum	wontfix	7	1	Não	Sim
128	3948	Funcionalidade	Defect	-	advck1123	worksforme	**	**	**	**
129	3947	Usabilidade	Defect	-	erion	wontfix	1	5	Sim	Sim
130	3939	Funcionalidade	Defect	jteh	jteh	fixed	**	**	**	**
131	3937	Usabilidade	Task	-	oaron	wontfix	7	10	Sim	Sim

Nº	ID Ticket	Categoria	Tipo	Owner	Autor	Motivo	HN. Princ.	HN. Comp.	Q1	Q2
132	3922	Usabilidade	Defect	-	netblue	worksforme	8	7	Sim	Sim
133	3913	Usabilidade	Defect	-	dallasobrien	fixed	1	**	Sim	Sim
134	3907	Usabilidade	Defect	-	ateu	worksforme	1	**	Sim	Sim
135	3906	Usabilidade	Defect	-	ateu	worksforme	1	**	Sim	Sim
136	3905	Usabilidade	Defect	jteh	ateu	fixed	1	7	Não	Sim
137	3901	Usabilidade	Defect	mdcurran	k_kolev1985	fixed	7	8	Não	Sim
138	3900	Usabilidade	Defect	mdcurran	k_kolev1985	fixed	1	7	Não	Sim
139	3899	Usabilidade	Defect	james teh	dallasobrien	fixed	1	7, 8	Não	Sim
140	3897	Usabilidade	Defect	jteh	pzajda	fixed	1	8, 7	Não	Sim
141	3895	Usabilidade	Enhancement	-	jacobchawala	cantfix	7	4	Sim	Sim
142	3894	Usabilidade	Defect	-	jacobchawala	worksforme	1	7	Não	Sim
143	3888	Funcionalidade	Defect	michael curran	mdcurran	fixed	**	**	**	**
144	3882	Usabilidade	Defect	michael curran	mdcurran	fixed	1	7	Não	Sim
145	3881	Usabilidade	Enhancement	michael curran	mdcurran	fixed	9	5	Não	Sim
146	3878	Usabilidade	Enhancement	-	hcdhoundiyal	cantfix	7	2	Sim	Sim
147	3874	Usabilidade	Enhancement	michael curran	mdcurran	fixed	7	8	Não	Sim
148	3873	Funcionalidade	Defect	michael curran	wmdcurran	fixed	**	**	**	**
149	3872	Usabilidade	Defect	jteh	lpintes	fixed	1	7	Sim	Sim
150	3870	Usabilidade	Enhancement	-	hcdhoundiyal	worksforme	1	7	Sim	Sim

Nº	ID Ticket	Categoria	Tipo	Owner	Autor	Motivo	HN. Princ.	HN. Comp.	Q1	Q2
151	3865	Usabilidade	Defect	james teh	ondrosik	fixed	10	4	Sim	Sim
152	3863	Usabilidade	Defect	-	erion	fixed	9	5	Sim	Sim
153	3860	Usabilidade	Defect	michael curran	dgoldfield	fixed	1	2, 4	Não	Sim
154	3859	Usabilidade	Defect	jteh	briang1	fixed	9	5	Sim	Sim
155	3856	Usabilidade	Defect	-	aliminator	worksforme	4	1	Sim	Sim
156	3851	Usabilidade	Enhancement	jteh	jteh	fixed	9	10	Sim	Sim
157	3849	Usabilidade	Defect	-	k_kolev1985	fixed	1	4	Sim	Sim
158	3845	Usabilidade	Defect	-	the_dark_man	fixed	1	4, 5	Sim	Sim
159	3844	Usabilidade	Defect	-	taghavi	fixed	4	1	Sim	Sim
160	3842	Funcionalidade	Enhancement	michael curran	mhameed	fixed	**	**	**	**
161	3839	Usabilidade	Defect	james teh	jteh	fixed	8	1	Não	Sim
162	3835	Usabilidade	Task	jteh	hyongsop.kim	fixed	4	2	Não	Sim
163	3834	Usabilidade	Enhancement	michael curran	leonarddr	fixed	7	1	Não	Sim
164	3833	Usabilidade	Defect	mdcurran	leonarddr	fixed	9	5	Sim	Sim
165	3828	Usabilidade	Task	-	oaron	worksforme	6	10	Sim	Sim
166	3827	Usabilidade	Defect	-	pzajda	fixed	9	5	Sim	Sim
167	3826	Usabilidade	Defect	michael curran	leonarddr	fixed	4	9	Não	Sim
168	3825	Usabilidade	Defect	jteh	jteh	fixed	3	4	Sim	Sim
169	3821	Usabilidade	Enhancement	-	aleskis	fixed	1	8	Não	Sim

Nº	ID Ticket	Categoria	Tipo	Owner	Autor	Motivo	HN. Princ.	HN. Comp.	Q1	Q2
170	3816	Usabilidade	Defect	james teh	k_kolev1985	fixed	7	1	Não	Sim
171	3810	Funcionalidade	Defect	-	annek	worksforme	**	**	**	**
172	3804	Usabilidade	Defect	jteh	briang1	fixed	9	**	Sim	Sim
173	3803	Usabilidade	Defect	james teh	briang1	fixed	8	4	Sim	Sim
174	3801	Funcionalidade	Enhancement	jteh	jteh	fixed	**	**	**	**
175	3800	Funcionalidade	Defect	michael curran	mdcurran	wontfix	**	**	**	**
176	3796	Funcionalidade	Defect	michael curran	jteh	fixed	**	**	**	**
177	3794	Funcionalidade	Enhancement	james teh	jteh	fixed	**	**	**	**
178	3793	Usabilidade	Defect	jteh	jteh	fixed	1	2,4	Sim	Sim
179	3786	Usabilidade	Defect	jteh	nvdakor	worksforme	1	4	Sim	Sim
180	3785	Usabilidade	Defect	michael curran	mdcurran	fixed	7	1	Sim	Sim
181	3784	Usabilidade	Defect	michael curran	mdcurran	fixed	7	4	Não	Sim
182	3783	Usabilidade	Enhancement	-	oaron	fixed	9	4	Não	Sim
183	3782	Usabilidade	Defect	jteh	briang1	fixed	9	4	Sim	Sim
184	3781	Usabilidade	Defect	michael curran	mdcurran	fixed	4	1	Não	Sim
185	3777	Usabilidade	Defect	michael curran	mdcurran	fixed	1	4	Sim	Sim
186	3774	Usabilidade	Enhancement	jteh	ondrosik	fixed	4	7, 2	Não	Sim

Nº	ID Ticket	Categoria	Tipo	Owner	Autor	Motivo	HN. Princ.	HN. Comp.	Q1	Q2
187	3770	Usabilidade	Defect	michael curran	mdcurran	fixed	4	1, 7	Não	Sim
188	3769	Funcionalidade	Task	mdcurran	kaban	worksforme	**	**	**	**
189	3760	Usabilidade	Defect	-	baphomet	worksforme	1	4	Sim	Sim
190	3758	Usabilidade	Defect	jteh	jteh	fixed	1	4	Sim	Sim
191	3757	Usabilidade	Defect	michael curran	mdcurran	fixed	1	4	Sim	Sim
192	3756	Funcionalidade	Defect	michael curran	mdcurran	fixed	**	**	**	**
193	3755	Usabilidade	Defect	michael curran	mdcurran	fixed	1	7	Não	Sim
194	3754	Usabilidade	Defect	michael curran	mdcurran	fixed	7	8, 1	Não	Sim
195	3753	Usabilidade	Defect	mdcurran	mdcurran	fixed	1	4	Sim	Sim
196	3748	Usabilidade	Defect	james teh	ondrosik	fixed	9	10	Não	Sim
197	3741	Usabilidade	Enhancement	jteh	birkirrg	fixed	1	2	Não	Sim
198	3739	Usabilidade	Defect	mdcurran	shadyar	fixed	9	4	Não	Sim
199	3735	Funcionalidade	Defect	-	ateu	cantfix	**	**	**	**
200	3732	Usabilidade	Defect	jteh	oaron	fixed	4	7	Sim	Sim