

Acessibilidade em métodos ágeis: uma revisão sistemática da literatura

Accessibility in agile methods: a systematic literature review

Accesibilidad en métodos ágiles: una revisión sistemática de la literatura

Recebido: 13/12/2019 | Revisado: 05/02/2020 | Aceito: 17/02/2020 | Publicado: 29/02/2020

Fernando D'Agostini Y Pablos

ORCID: <https://orcid.org/0000-0002-9453-1130>

Centro Estadual de Educação Tecnológica Paula Souza, Brasil

E-mail: fernandopablosweb@gmail.com

Marcelo Duduchi Feitosa

ORCID: <https://orcid.org/0000-0002-0969-4737>

Centro Estadual de Educação Tecnológica Paula Souza, Brasil

E-mail: mduduchi@fatecsp.br

Resumo

Métodos ágeis surgiram como uma resposta mais flexível aos métodos mais antigos de desenvolvimento de software, que eram mais focados em planejamento do que em interações. Em paralelo, houve uma evolução da qualidade do software desenvolvido, passando-se a considerar requisitos de sistema como interação do usuário, usabilidade e acessibilidade. Neste contexto o presente artigo tem por objetivo identificar por meio de uma revisão sistemática de literatura a atenção dada à acessibilidade no desenvolvimento de software em escritos acerca de métodos ágeis, verificando como a acessibilidade tem sido trabalhada em conjunto com os métodos ágeis de desenvolvimento de software. A pesquisa pôde encontrar estudos de diversas áreas do ciclo de vida do software, como processos de desenvolvimento, métodos de avaliação de acessibilidade e refatoração para acessibilidade, o que mostra que apesar da pesquisa na área ainda ser incipiente, evoluiu consideravelmente na última década.

Palavras-chave: Métodos ágeis; Acessibilidade; Desenvolvimento de software acessível.

Abstract

Agile methodologies were born as a more flexible response to older methods of software development, that were more focused on planning than on interactions. In parallel, there is an evolution of the quality of the developed software, considering systems requirements such as user interaction, usability and accessibility. In this way, a systematic literature review was

carried out aiming to identify the attention given to accessibility in software development in writings about agile methods, identifying how accessibility has been worked with agile software development methods. The research could find some studies in several areas of the software life cycle, such as development processes, accessibility assessment methods and accessibility refactoring, which show that although research in this area is still incipient, it has evolved considerably in the last decade.

Keywords: Agile methods; Accessibility; Accessible software development.

Resumen

Los métodos ágiles surgieron como una respuesta más flexible a los métodos de desarrollo de software más antiguos, que se centraron más en la planificación que en las interacciones. Paralelamente, hubo una evolución de la calidad del software desarrollado, tendo en mente los requisitos del sistema, como la interacción del usuario, usabilidad y accesibilidad. De esta manera, se realizó una revisión sistemática de la literatura para identificar la atención prestada a la accesibilidad en el desarrollo de software en escritos sobre métodos ágiles, identificando cómo se ha trabajado la accesibilidad junto con los métodos de desarrollo de software ágil. La investigación ha encontrado algunos estudios en diversas áreas del ciclo de vida del software, como los procesos de desarrollo, los métodos de evaluación de accesibilidad y la refactorización de la accesibilidad, o que muestran que si bien la investigación en el área aún es incipiente, ha evolucionado considerablemente durante la última década.

Palabras clave: Métodos ágiles; Accesibilidad; Desarrollo asequible de software.

1. Introdução

Nas últimas décadas, em todo o mundo, passou-se a dar maior atenção às pessoas com deficiência e às necessidades que acabam por fazer parte da rotina de uma grande parcela da população. No Brasil, por exemplo, o IBGE (2012) analisou a predominância de deficiências em seus diversos graus e contextos, considerando pessoas de todas as idades e de diferentes regiões do país, obtendo como resposta que 45.606.048 milhões de pessoas declararam ter pelo menos uma das deficiências investigadas, correspondendo a 23,9% da população brasileira à época.

Até 2030, todos os países do mundo terão a experiência do envelhecimento de sua população. Nas próximas quatro décadas, 22% da população mundial deverá ser de pessoas com acima de 60 anos (Bloom, et al., 2015). Este processo de envelhecimento populacional

exige interação de diversas áreas, identificando aspectos socioculturais, psicológicos, econômicos e relacionados à saúde desses indivíduos, bem como proteger das doenças e deficiências ou disfunções relacionadas à idade (Sander, et al., 2014).

Muitas pessoas idosas possuem alguns prejuízos sensoriais que podem dificultar a forma como interagem com o computador, como declínios na visão, em habilidades físicas, de escuta ou cognitivas (W3C, 2018), assim, também é importante, para essas pessoas, que as interfaces de softwares sejam acessíveis.

Desta forma, para a inclusão e para permitir a participação das pessoas com deficiência, uma das bandeiras existentes na alçada da tecnologia da informação é a acessibilidade *web*, que torna possível a participação na sociedade de milhões de sujeitos com necessidades especiais, disponibilizando interfaces que atendam suas necessidades e preferências (Conforto & Santarosa, 2002).

Em 1994, o Standish Group analisou projetos de desenvolvimento de software e o porquê de seus índices expressivos de falha, observando que apenas, em média, 31% dos projetos eram bem sucedidos, enquanto 53% enfrentaram dificuldades em cumprir o previsto no projeto e 16% falhavam por completo. Diversos fatores que auxiliam a gestão do projeto puderam ser identificados, como o “Envolvimento”, o “Suporte executivo” e a “Clara especificação de requisitos” (Standish Group, 1994).

No ano de 2013, comparativamente a 1994, não houve muita diferença, observando-se que 36% dos projetos foram bem sucedidos, 48% enfrentaram dificuldades e 16% falharam por completo. Porém, quanto aos fatores identificados para o sucesso dos projetos, há um novo que chama a atenção, que seria a utilização de “processo ágil”. Verifica-se que, comparativamente, ao se considerar todos os portes de projeto, 46% dos projetos ágeis foram bem sucedidos frente a apenas 14% do método de cascata, 44% enfrentaram dificuldades em métodos ágeis frente a 58% em método cascata e apenas 10% falharam em métodos ágeis em comparação a 28% no método cascata (Standish Group, 2014).

Os chamados métodos ágeis vieram em resposta a estas necessidades de aprimoramento de projetos de software, dando maior ênfase à flexibilidade e à interação do que ao foco no planejamento prévio, que ocasionava baixa flexibilidade e pouca interação com os consumidores (Serrador & Pinto, 2015). Eles também trouxeram ganho de motivação e poder para os desenvolvedores de *software* (Dingsøyr, et al., 2012).

Estes importantes aspectos, dentre diversos outros, têm influenciado mudanças na área de desenvolvimento de software na atualidade. De um lado, temos o aspecto da acessibilidade de software, que é fruto de dois fatores: a crescente importância dada à inclusão de pessoas com deficiência (PCD) e o envelhecimento da população mundial. De outro lado, temos a crescente utilização de métodos ágeis na última década, que trouxe maior flexibilidade e interação frente aos modelos mais tradicionais (Serrador & Pinto, 2015)

O objetivo do trabalho é identificar por meio de uma revisão sistemática da literatura a atenção dada à acessibilidade no desenvolvimento de software em escritos acerca de métodos ágeis, verificando como a acessibilidade tem sido trabalhada em conjunto com os métodos ágeis de desenvolvimento de software.

2. Métodos ágeis

Os métodos ágeis ganharam força a partir do Manifesto Ágil (Agile Alliance, 2001), que teve o objetivo de prover melhores formas de desenvolver softwares, focando em indivíduos e interações, funcionalidades, colaboração e mudanças ágeis, frente aos chamados modelos prescritivos da época, que davam maior ênfase a processos, documentações e planejamento.

Erickson, Lyytinen & Siau (2005) definem agilidade como deixar de lado todo o peso dos modelos prescritivos de desenvolvimento de software e promover respostas rápidas às mudanças de ambientes, em requisitos, bem como a aceleração da execução do projeto. Williams & Cockburn (2003) observam que desenvolvimento ágil de software é relacionado a frequentes retroalimentações e mudanças.

Os principais atributos dos métodos ágeis são a abordagem incremental (ciclos rápidos de desenvolvimento, pequenas entregas), a frequente cooperação (cliente próximo e interação de desenvolvedores), a forma direta (método fácil de aprender e de modificar, e suficientemente documentado) e as frequentes adaptações (capacidade de se adaptar às frequentes mudanças) (Abrahamsson, et al., 2002).

Dentre os principais métodos ágeis conhecidos, podem ser destacados os mais utilizados na atualidade, que são o Scrum, o Extreme Programming (XP), o Lean e o Kanban (Version One, 2018). Cada um desses métodos possui diferentes características, porém observam os princípios do Manifesto Ágil.

Serrador & Pinto (2015) analisaram em um estudo de larga escala se métodos ágeis possuem melhores resultados do que os modelos prescritivos no sentido de sucesso de projetos. Apesar de boa parte dos respondentes trabalharem com equipes que possuem características ágeis, mas não serem completamente ágeis, observou-se que aqueles que possuíam mais características relacionadas a métodos ágeis possuíam maior sucesso frente aos demais. Também se observou que para os projetos desenvolvidos por equipes ágeis, não há relação entre o sucesso dos projetos com o tempo de experiência dos participantes.

3. Deficiências e acessibilidade

Ao longo da história, as deficiências já foram vistas por diversos pontos diferentes, muitas vezes de forma negativa, porém o conceito foi evoluindo com o passar do tempo, e, nos tempos atuais, elas são vistas como perdas ou anormalidades de estruturas ou funções como a psicológica, fisiológica ou anatômica e que gere incapacidade para desempenho de alguma atividade que é considerada normal para o ser humano (Silva, 2020).

Para a construção de uma sociedade participativa, seus cidadãos devem possuir formas de interagir entre si. Seguindo essa linha de pensamento, políticas de inclusão devem ser engendradas, criando-se, assim, uma sociedade em que todos podem participar com equidade e de acordo com suas especificidades (Conforto & Santarosa, 2002).

A deficiência tem sido encarada como uma questão de direitos humanos e, dessa forma, conforme a Convenção Internacional sobre os Direitos das Pessoas com Deficiência e Seu Protocolo Facultativo da ONU, o Governo Federal do Brasil assinou o Decreto nº 6.949 (Brasil, 2009) que visa assegurar, dentre outros pontos, a acessibilidade, tanto em seu sentido físico e estrutural das cidades, quanto no âmbito de acesso à informação e comunicação.

Enquanto para a maioria das pessoas a tecnologia facilita a vida, para pessoas com deficiência a tecnologia pode tornar as coisas possíveis. Com a tecnologia, elas podem acessar informações, exercer uma atividade e encontrar formas alternativas de lazer, por exemplo (Fernandes & Godinho, 2003). A acessibilidade pode ser compreendida como o acesso aos ambientes físicos, bens e serviços, às pessoas e à informação (Melo & Baranauskas, 2006).

O termo acessibilidade tem sido empregado neste sentido de tornar possível a interação com todo o leque de pessoas com deficiência e pode ser conceituado, no sentido da acessibilidade *web*, com o fato de pessoas com deficiências usarem a *web* e, ainda mais especificamente, poderem percebê-la, entendê-la, bem como navegar, interagir e contribuir

com ela. Também outro conceito bem aceito, conforme os mesmos autores, é de que as tecnologias são acessíveis se for possível utilizá-las tanto por pessoas com deficiência quanto por aquelas que não as possuem (Yesilada, et al., 2012).

4. Metodologia

A fim de identificar a relação entre acessibilidade e desenvolvimento de software com métodos ágeis, realizou-se uma revisão sistemática de literatura (RSL). Uma RSL se trata de uma forma de estudo secundário que define uma metodologia para identificar, analisar e interpretar toda evidência disponível sobre uma questão específica de pesquisa sem vieses e de forma replicável (Kitchenham & Charters, 2007). A revisão se baseou na proposta de Kitchenham & Charters (2007), que apresenta boas práticas para se realizar uma RSL na área de pesquisa de engenharia de software.

Como estratégia de busca de estudos primários, utilizaram-se as bases de dados IEEEExplore, EI Compendex (Engineering Village), Google Scholar e ACM Digital Library. Para identificar estudos primários em potencial para responder à questão desta pesquisa, empregou-se a seguinte *string*: ("agile" OR "scrum" OR "Extreme Programming" OR "XP" OR "Kanban" OR "Lean" OR "ágil") AND ("accessibility" OR "universal design" OR "design for all" OR "inclusive design" OR "acessibilidade" OR "design universal" OR "design para todos" OR "design inclusivo").

Após a identificação de novas palavras-chave pertinentes, uma segunda busca foi realizada nas mesmas bases com a *string*: ("usability" OR "usabilidade") AND ("disability" OR "deficiência") AND ("agile" OR "scrum" OR "Extreme Programming" OR "XP" OR "Kanban" OR "Lean" OR "ágil").

Uma vez que nem todos os mecanismos de busca utilizam a mesma sintaxe, as *strings* foram adaptadas de acordo com cada mecanismo, porém mantendo a consistência com as palavras-chaves apresentadas.

Como critérios para seleção de artigos, propuseram-se os critérios de inclusão e exclusão constantes da Tabela 1.

Tabela 1 – Critérios de inclusão e de exclusão de artigos

Critérios de inclusão (CI)	Critérios de exclusão (CE)
CI01. Artigos de periódicos e congressos relacionados	CE01. Artigos que não possuem foco em

à área	acessibilidade e métodos ágeis
CI02. Artigos em português ou inglês	CE02. Artigos de periódicos de outras áreas não pertinentes
CI03. Produções que possuem referências e procedimentos devidos	CE03. Artigos de congressos não pertinentes à área
CI04. Publicações entre os anos de 2007 e 2017	CE04. Artigos duplicados ou muito similares

Fonte: elaborado pelos autores

Propôs-se um modelo para extração dos seguintes dados úteis à análise dos estudos primários identificados, que contempla: título, objetivos do estudo, método, participantes envolvidos, forma de coleta e análise dos dados, resultados, conclusões e demais análises. Os dados extraídos dos estudos primários selecionados foram sintetizados com base nestas informações. Após isto, houve a categorização.

A qualidade dos artigos identificados também é apontada por Kitchenham & Charters (2007) como um ponto crítico, pois aprimora ainda mais os critérios de inclusão e exclusão de artigos, esclarece a importância individual dos resultados dos artigos e, deste modo, reduz as vieses e maximiza a validade interna e externa da pesquisa.

Assim, estabeleceram-se 07 critérios de qualidade (CQ), que estão expostos na Tabela 2.

Tabela 2 – Critérios de qualidade para análise dos artigos (CQ)

Questionamento
CQ01. Os objetivos e questões de pesquisa são claros?
CQ02. A pesquisa agregou novos conhecimentos?
CQ03. O método utilizado foi bem descrito e é apropriado à pesquisa?
CQ04. A pesquisa foi planejada para atingir seus objetivos e responder sua questão de pesquisa?
CQ05. A amostragem foi suficiente?
CQ06. Os laços entre dados, interpretação e conclusão são claros?
CQ07. As conclusões e análises apresentadas são bem embasadas?

Fonte: elaborado pelos autores

Então, cada um dos artigos foi avaliado para cada um dos CQ's elencados, atribuindo-se as notas: 1 (pouco), 2 (médio) e 3 (muito). Após a análise criterial, a somatória das notas de

todos os critérios resultou na qualidade geral do artigo (QG).

5. Resultados

Os resultados da primeira busca foram filtrados por relevância, sendo possível separar os artigos disponíveis relacionados ao tema em cada uma das bases, conforme Tabela 3.

Tabela 3 – Artigos disponíveis nas bases de dados pesquisadas na primeira busca

Base de dados	Quantidade total	Analisados	Selecionados
IEEEExplore	1.983	1.983	02
EI Compendex (Engineering Village)	136	136	02
Google Scholar	243.000	1.500	08
ACM Digital Library	211	211	0

Fonte: elaborado pelos autores

Após a aplicação dos CI e CE's, localizaram-se 02 artigos na base IEEEExplore, 02 na base EI Compendex, 08 na base Google Scholar e 0 na base ACM Digital Library.

Os resultados da segunda busca foram filtrados por relevância, sendo possível separar os artigos disponíveis relacionados ao tema em cada uma das bases, conforme a Tabela 4.

Tabela 4 – Artigos disponíveis nas bases de dados pesquisadas na segunda busca

Base de dados	Quantidade total	Analisados	Selecionados
IEEEExplore	108	108	01
EI Compendex (Engineering Village)	05	05	0
Google Scholar	4.230	1.000	05
ACM Digital Library	02	02	0

Fonte: elaborado pelos autores

Após a aplicação dos CI e CE's, localizaram-se 01 artigo na base IEEEExplore, 0 na base EI Compendex, 05 na base Google Scholar e 0 na base ACM Digital Library.

Nas duas buscas, optou-se por não realizar a análise completa dos resultados da base Google Scholar, uma vez que os resultados foram filtrados por relevância e, quando se optou por interromper a análise, mais de 200 artigos prévios não possuíam sequer relação indireta

com o assunto pesquisado.

Os artigos selecionados nas duas buscas foram analisados de acordo com os CQ's especificados, constando da Tabela 5.

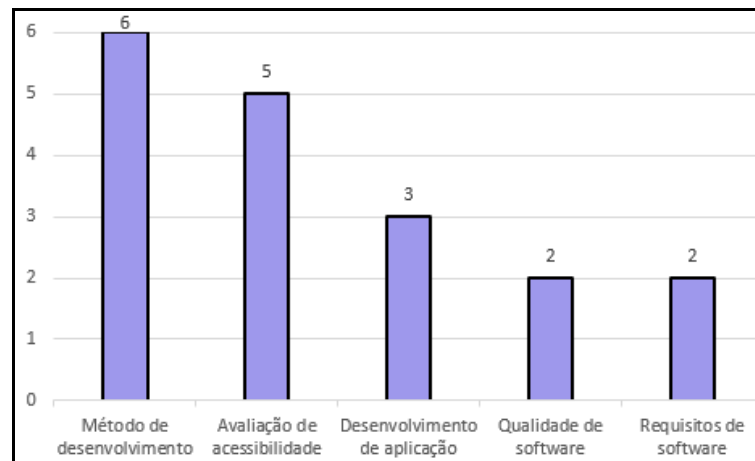
Tabela 5 – Qualidade geral das literaturas (QG)

Artigo	QG
(Bai, Mork & Stray, 2017)	21
(Balasubramani et al., 2016)	21
(Bonacin, Baranauskas & Rodrigues, 2009)	21
(Farid & Mitropoulos, 2012)	19
(Garrido et al., 2014)	21
(González-González, Toledo-Delgado & Muñoz-Cruz, 2015)	19
(Koenig et al., 2014)	20
(Luján-Mora & Masri, 2012)	7
(Masri & Luján-Mora, 2011)	10
(Medina et al., 2010)	18
(Prior et al., 2013)	21
(Reichling, Cherfi, 2013)	21
(Sánchez-Gordón, Luján-Mora, 2017)	10
(Shelly & Barta, 2010)	10
(Tiangtae et al., 2017))	13
(Watanabe, Fortes & Dias, 2012)	21
(Williams et al., 2014)	21
(Younas et al., 2017)	19

Fonte: elaborado pelos autores

Ainda como resultado da pesquisa, realizou-se uma síntese da literatura localizada. Os artigos selecionados foram categorizados em “método de desenvolvimento”, “avaliação de acessibilidade”, “desenvolvimento de aplicação”, “qualidade de software” e “requisitos de software”, conforme ilustra a Figura 1, e a síntese de cada um deles é apresentada a seguir.

Figura 1 – Categorização dos artigos sobre métodos ágeis e acessibilidade



Fonte: elaborado pelos autores

Como é possível observar na Figura 1, a maioria dos artigos está relacionada a métodos de desenvolvimento e avaliações de acessibilidade. Já os artigos sobre desenvolvimento de aplicação, qualidade de software e requisitos de software aparecem em menor número.

As referências acerca de métodos de desenvolvimento propõem novas formas de se desenvolver com métodos ágeis ou comprovar sua eficiência.

Bonacin, Baranauskas & Rodrigues (2009) apresentam um modelo de processo para desenvolvimento ágil de software que leva em conta requisitos de acessibilidade, chamado de Modelo de Processo Ágil Inclusivo. Esse modelo promove a participação de usuários e stakeholders com os valores de acessibilidade universal e design inclusivo, constrói uma visão compartilhada do contexto social, inclui mais do que problemas técnicos no desenvolvimento do sistema e promove a inclusão digital a partir de atividades participativas. O método se baseia nos conceitos de XP. Após todos os ciclos do processo proposto, como ciclo de versão e de prototipação, no ciclo de desenvolvimento, há inspeção de código por especialistas de acessibilidade e usabilidade. O modelo foi validado por meio de estudo de caso, trazendo resultados favoráveis.

Medina, et al. (2010), considerando a acessibilidade como um aspecto importante, porém não o único, focam em usabilidade para usuários com deficiência. Propuseram uma forma de refatoração de aplicações que já eram acessíveis, mas não possuíam usabilidade. Consideram que apesar da existência de ferramentas que transformam o código em acessível ou o uso de tecnologias assistivas, como leitores de telas, para que uma aplicação seja de fato universalmente usável, é necessária intervenção manual, devendo vir desde os primeiros estágios do design da aplicação, bem como deve haver participação de todos os interessados

como consumidores, designers e usuários. Os passos propostos para a refatoração são de capturar requisitos da aplicação, desenvolver a aplicação com base nas recomendações da WCAG, detectar problemas de usabilidade universal e refatorar a aplicação para obter uma aplicação que não possui os problemas encontrados. O ciclo de refatoração é realizado incrementalmente, baseando-se no método WebTDD. Na pesquisa são apresentados um catálogo de refatorações para problemas de cegos e pessoas com deficiência visual, um processo de desenvolvimento e um metamodelo que representa os elementos da aplicação e as mudanças.

Lujan-Mora & Masri (2012) buscam comprovar que métodos ágeis são os melhores métodos para se desenvolver sites e aplicativos acessíveis, já que, como métodos ágeis são iterativos e diversas versões são aprimoradas, torna-se mais fácil desenvolver sistemas e sites acessíveis. Consideram que a melhor forma de desenvolver sistemas acessíveis é a partir da realização de testes de acessibilidade com envolvimento de usuários. Afirmam que para cada perfil de usuário com deficiência, cinco usuários devem testar para se adquirir um bom retorno de problemas de acessibilidade e usabilidade.

Reichling & Cherfi (2013) propõem a integração da acessibilidade desde a elicitación e análise de requisitos prosseguindo ao longo de todo o processo de desenvolvimento. O método proposto se divide em três partes principais: análise, design e avaliação. A etapa de análise ajuda a entender os requisitos dos usuários e necessidades especiais para definir o contexto de uso, como conhecimento destes, tecnologias assistivas utilizadas, frequência de uso, deficiências físicas e cognitivas analisadas, entre outros. Nessa fase já devem constar os requisitos de acessibilidade. A fase de design produz o design que atende aos requisitos levantados na etapa anterior. Há envolvimento dos usuários no processo, card sorting, em que os participantes nomeiam informações que estarão disponíveis e testes de usabilidade com os usuários. A fase de avaliação avalia a qualidade do produto, bem como a conformidade com os padrões de acessibilidade e a satisfação dos usuários finais recebendo seus feedbacks. O método mescla design centrado no usuário com acessibilidade. Os princípios devem constar antes das sprints, para já inserir o usuário desde o início e as metas referentes à acessibilidade são definidas logo no início, dividindo as atividades em ciclos iterativos, de forma a aprimorar constantemente a acessibilidade, sendo esta a primeira fase. A segunda fase consiste em elaborar uma completa visão da arquitetura do produto e suas interações. Na última fase os entregáveis já são mais perceptíveis e mensuráveis, sendo separados em sprints. São necessárias métricas para monitorar o projeto, que se dividem em três níveis: meta, questões e métricas. Os autores validaram a proposta e, ao segui-la, observaram que houve melhora da

relevância do produto e da satisfação dos usuários quanto à acessibilidade, bem como se percebeu que considerar a acessibilidade desde o início poupa tempo e dinheiro.

Já Williams, et al. (2014) propõem uma forma de equipes ágeis desenvolverem interfaces acessíveis a usuários idosos. Os autores acreditam que equipes ágeis, por possuírem adaptabilidade, comunicabilidade com a audiência e simplicidade, são mais propensas a adaptarem interfaces a usuários idosos. As boas práticas identificadas foram a de permitir testes de usuário entre sprints para verificar se está utilizável da forma atual, escutar uma grande variedade de usuários idosos com perfis diferentes, estar preparado para adicionar requisitos para facilitar o uso do aplicativo, priorizar a simplicidade no desenvolvimento e criação de interfaces e criar tutorial ao final de desenvolver todos os requisitos. Os autores realizaram um estudo de caso com usuários idosos, tendo retorno positivo dos usuários.

González-González, Toledo-Delgado & Muñoz-Cruz (2015) apresentam técnicas baseadas em design centrado no usuário e em métodos ágeis, em que nas sprints há envolvimento dos usuários no processo de desenvolvimento. Os autores desenvolveram um sistema web que suporta o ensino de operações matemáticas básicas a crianças com necessidades educativas especiais, com a participação destes no processo de desenvolvimento, levando em conta princípios de design universal. Houve uso de personas e protótipos, já que a participação de pessoas com o perfil não é tão simples. Um grupo de profissionais das áreas de matemática e psicologia e também professores das crianças criaram personas adaptadas ao caso. Também foram realizados testes de protótipos, avaliações de usabilidade e redesign interativo, bem como grupos de foco com especialistas. Diversas heurísticas foram analisadas e os problemas foram identificados em cada uma delas, visando readaptar o sistema para torná-lo mais acessível e usável. O software foi desenvolvido conforme o método proposto e testes foram realizados com especialistas para observar os prós e contras.

As referências sobre a avaliação de acessibilidade estão relacionadas a métodos, estratégias e análises sobre avaliação de acessibilidade de sistemas web e portais.

Para Shelly & Barta (2010), a validação da acessibilidade tem sido tratada principalmente com auditorias realizadas após a entrega do sistema pronto e apenas alguns testes com usuários são realizados, porém não desde o início do projeto. Propõem então que haja, independente do modelo de processo de desenvolvimento ou método utilizado, que o time de teste possua testes relacionados a acessibilidade, que seriam realizados desde o design e ao longo de todo o processo de desenvolvimento, validando possíveis problemas de acessibilidade, bem como tomando atitude preventiva quanto aos erros. Afirmam ainda que testes com usuários e testes com usuários com deficiências são inestimáveis, porém não são

tão precisos, uma vez que supõem que estando acessível a uma determinada tecnologia assistiva ou perfil, também estaria para os demais, o que pode não ser real. As atuais formas de verificação, como os testes de usabilidade e os testes beta, ou testes automatizados, possuem seu valor, porém não são sistemáticos o suficiente e possuem alto custo para realização. Assim, concluem que a saída seria a utilização de testes formais de software, por serem capazes de isolar mais adequadamente erros encontrados, bem como assegurar a análise ao longo de todo o processo de desenvolvimento.

Masri & Lujan-Mora (2011) propõem um método ágil para avaliação de acessibilidade web que envolve diversos passos, como o passo analítico, que engloba um estágio de pré-análise e outro de avaliação automática do conteúdo e o de avaliação empírica, que envolve usuários e planejamento de testes, ao final propondo uma métrica de verificação de acessibilidade.

Watanabe, Fortes & Dias (2012), por meio de estudo de caso comparando os testes de aceitação de acessibilidade propostos com validadores automatizados em interfaces desenvolvidas, reportam o desenvolvimento e uso de uma ferramenta de validação automática de acessibilidade para aplicações ricas de internet (RIA), validando, assim, para cada cenário e estratégia de uso.

Sanchez-Gordón & Luján-Mora (2017) apresentam um método de avaliação de acessibilidade web para equipes ágeis que envolve cinco estágios, que vão desde o planejamento de testes e controle, análise de testes e design, implementação de teste e execução, avaliação de critérios finais até o relato e atividades de fechamento de testes. Com base em diversas atividades de planejamento, controle, design, acessibilidade e várias outras atividades, envolvem-se ferramentas, experts da área e testes de usuário.

Bar, Mork & Stray (2017) analisam o custo-benefício dos métodos de avaliação de acessibilidade no desenvolvimento ágil de software realizando dois estudos de caso com dois softwares diferentes: o primeiro verificando o custo-benefício de diversos métodos e o segundo analisando os métodos que possuem mais valor para testar acessibilidade, com base nos resultados do primeiro. Os métodos foram quantificados em baixos, médios e altos custos envolvidos. Com base nos resultados, é apresentado um método ágil com base em uma espiral, com as seguintes etapas: desenvolvimento, teste e revisão em sucessivas iterações, levando em conta as validações de acessibilidade. Concluíram que o melhor custo-benefício é a ferramenta de simulação de visão reduzida; após isso, verificador automático de acessibilidade, na sequência, checklist do WCAG, leitor de tela e finalmente testes com personas.

As referências sobre desenvolvimento de aplicação são relacionadas ao desenvolvimento de alguma aplicação acessível com o uso de métodos ágeis.

Prior et al. (2013) relatam o uso de um método ágil em um projeto de desenvolvimento de software de comunicação aumentativa e alternativa, que permite que adultos com sérios prejuízos de fala e físicos possam relatar suas histórias do dia-a-dia com mais facilidade. O projeto foi desenvolvido com uma equipe especializada, que se baseou no método ágil Scrum. Houve envolvimento de pacientes com paralisia cerebral, deficiências cognitivas e de comunicação, que validaram wireframes ao longo do desenvolvimento todo. O uso de métodos ágeis garantiu maior liberdade à equipe e facilitou o envolvimento dos usuários.

Koenig, et al. (2014) apresentam métodos e desafios para o design centrado no usuário no domínio da reabilitação neuropsicológica. São considerados desenvolvimento de aplicações de realidade virtual para treino neuropsicológico e outros fatores de design. Desenvolveu-se uma aplicação que simula um escritório utilizado para avaliação cognitiva com o uso de métodos ágeis, pelo fato de tornar mais fácil a inserção de usuários no processo de desenvolvimento. Pesquisadores clínicos participaram do processo de desenvolvimento, bem como pacientes com danos cerebrais foram inseridos após o desenho e desenvolvimento iniciais.

Tiangtae, et al. (2017) relatam a experiência do desenvolvimento de um software utilizando métodos ágeis para a comunidade que possui surdez na Tailândia. O software possuía dois requisitos funcionais essenciais: apresentar uma animação 3D e traduzir tailandês para linguagem de sinais. Os requisitos não-funcionais foram mais sofisticados, pois era necessário que a animação possuísse precisão de modo a ser inteligível pelos usuários. Durante o desenvolvimento, tradutores voluntários e estudantes com surdez testaram o que estava sendo desenvolvido. Inicialmente houve muitos problemas, porém, com ajustes, o software passou nos testes. Os testes de aceitação foram realizados por representantes de organizações de surdos e após muitos ajustes, por volta de metade das palavras foram aprovadas, o que não foi completamente satisfatório. O problema identificado foi a falta de expressões faciais, o que dificulta o entendimento. Assim, uma nova versão com expressões faciais passou a ser desenvolvida. Observou-se que não é possível utilizar os atuais processos de engenharia de software para usuários com deficiência. O software não atendeu plenamente as necessidades dos interessados e a participação de apenas estudantes e os tradutores voluntários durante os testes de desenvolvimento não foi suficiente, pois possivelmente se houvessem participado desde o início do processo poderia ter havido melhor resultado.

As referências sobre qualidade de software dizem respeito a melhorias relacionadas à

qualidade do software desenvolvido.

Garrido, et al. (2014) apresentam a utilização de refatoração para melhorar a acessibilidade de aplicações web para usuários de leitores de tela sem comprometer a usabilidade para outros públicos, gerando assim uma "usabilidade acessível" para aplicações web. Os autores afirmam que métodos ágeis, por já realizarem refatoração, facilitam o uso das práticas propostas. As práticas são: simplificar o conteúdo, organizá-lo, adicionar sumários, indicar tamanho, remover conteúdos desnecessários, alterar menus inacessíveis, mostrar todos os links de estrutura, inserir checkboxes nos lugares corretos, entre outros. Realizaram um estudo de caso de refatoração de um leitor de e-mail da Universidade de Granada, que era inacessível. Os pesquisadores realizaram uma busca inicial por problemas conhecidos em sistemas de e-mail, após isso aplicou-se um teste com um usuário cego e questionários foram aplicados a outros usuários cegos tendo por fim especificar outros problemas. Após isto, os problemas foram removidos das interfaces por meio de refatoração. Foram separados quatro grupos: o primeiro com 6 participantes cegos que utilizaram a aplicação sem refatoração, o segundo com 6 participantes cegos que utilizaram a aplicação após a refatoração, o terceiro com 6 usuários que não possuem deficiência visual utilizando a versão prévia à refatoração e o quarto com 6 participantes que não possuem deficiência visual que utilizaram a versão refatorada. Foram atribuídas três tarefas idênticas a cada grupo. A versão refatorada obteve resultados melhores para todos os grupos, especialmente para os participantes cegos. Notou-se que com apenas algumas mudanças é possível aprimorar a acessibilidade da aplicação web.

Já Balasubramani, et al. (2016) propõem um método, nomeado "bug bash", ou "festa das falhas", de entregar produtos de alta qualidade no ambiente ágil. As áreas a serem analisadas (como segurança, usabilidade, performance, acessibilidade) são elencadas na etapa de planejamento e, após isso, testadores especialistas nas áreas propostas se tornam os responsáveis por testar o software desenvolvido. São criadas "salas de guerra", em que os participantes são reunidos para resolverem problemas de forma rápida e os dados e a aplicação já devem estar disponíveis para teste nessas reuniões previamente. O líder organiza essas reuniões para testar estratégias e facilitar o trabalho do time. Diversos pontos em separado podem ser analisados, como segurança, testes de integração, testes funcionais e também testes de acessibilidade. Após o evento, feedbacks são recebidos, é realizada uma retrospectiva e um painel é criado com itens em destaque. Para validar o método, os autores analisaram as falhas de um software sem utilizar o método comparativamente com utilizando o método, observando que o uso do "bug bash" facilita para que erros ainda não identificados sejam localizados. Desta forma, o método traz uma melhora na qualidade do software, maior

domínio, melhora da visão do consumidor e aprimora a qualidade.

Por fim, as referências acerca de requisitos de software apresentam melhorias relacionadas a requisitos funcionais e não-funcionais para métodos ágeis.

Farid & Mitropoulos (2012) apresentam, uma vez que métodos ágeis não possuem muitas práticas consolidadas de identificação, modelagem e ligação de requisitos funcionais com requisitos não-funcionais, uma ferramenta de modelagem de requisitos não-funcionais. Realizaram dois estudos de caso com dois sistemas distintos e concluíram que a ferramenta desenvolvida pode ajudar a modelar requisitos funcionais, não-funcionais e operacionalizações para uso em métodos ágeis.

Já Younas, et al. (2017) especificam um guia de elicitação de requisitos não-funcionais para métodos ágeis, dentre os quais a acessibilidade. É criado um cartão que mapeia os requisitos não-funcionais, com base em um processo de elicitação deste tipo de requisitos. O processo leva em conta a elicitação através de entrevista, definição do tipo de software, identificação dos requisitos (aprofundamento teórico sobre o assunto, levando o time a implementar o requisito elencado com mais propriedade), seleção de especialistas no assunto, identificação do problema, lista de requisitos não-funcionais candidatos, validação com especialista, validação com o usuário e o encerramento. Os autores realizaram um estudo de caso em que os estudantes reportaram os requisitos não-funcionais utilizando o método e não o utilizando, concluindo-se que o uso deste traz benefícios frente a outros métodos que não são desenvolvidos exclusivamente para métodos ágeis.

6. Conclusão

Com base nos resultados obtidos, é possível notar que poucos estudos foram realizados relacionando os métodos ágeis com a acessibilidade web, o que demanda maior atenção da comunidade científica.

A maioria dos artigos encontrados está relacionada a métodos de desenvolvimento e avaliações de acessibilidade. Já os artigos sobre desenvolvimento de aplicação, qualidade de software e requisitos de software aparecem em menor número.

As referências acerca de métodos de desenvolvimento propõem novas formas de se desenvolver com métodos ágeis ou comprovar sua eficiência, as referências sobre avaliação de acessibilidade se referem a métodos, estratégias e análises sobre avaliação de acessibilidade de sistemas web e portais, as referências sobre desenvolvimento de aplicação são relacionadas ao desenvolvimento de alguma aplicação acessível com o uso de métodos

ágeis, as referências sobre qualidade de software dizem respeito a melhorias relacionadas à qualidade do software desenvolvido. As referências sobre qualidade de software dizem respeito a melhorias relacionadas à qualidade do software desenvolvido e, por fim, as referências acerca de requisitos de software apresentam melhorias relacionadas a requisitos funcionais e não-funcionais para métodos ágeis.

Observa-se também que diversas referências localizadas não apresentam comprovação de validade, uma vez que não fazem validação de suas propostas ou métodos, podendo-se extrair destas apenas seu conteúdo teórico. Apesar disto, é possível observar que houve evolução da quantidade de publicações na área na última década, o que, apesar de ser incipiente, demonstra interesse da comunidade acadêmica em relacionar os dois temas tão atuais e que podem se inter-relacionar.

Como sugestões para continuidade dessa pesquisa, podem ser realizados levantamentos (survey) referentes aos aspectos observados nessa revisão sistemática da literatura. Ainda, cada um dos aspectos pode ser analisado individualmente de forma qualitativa, através de estudos de caso individualizados, para verificar na prática como a acessibilidade está sendo trabalhada ao longo do processo de desenvolvimento de software.

Referências

Abrahamsson, P., Salo, O., Ronkainen, J., & Warsta, J. (2002). Agile Software Development Methods: Review and Analysis. ArXiv, abs/1709.08439.

Agile Alliance. (2001). Manifesto for Agile Software Development. Disponível em: <https://agilemanifesto.org>. Acesso em: 17 mai. 2017.

Bai, A., Mork, H. C., & Stray, V. (2017). A cost-benefit analysis of accessibility testing in agile software development results from a multiple case study. *Int. J. Adv. Softw*, 10(1). Disponível em: <https://www.duo.uio.no/handle/10852/70786>.

Balasubramani, U. M., Iyer, K., Krishnan, B. S., & Kovvuri, H. (2016, October). Bug Bash: An Efficient Approach to Increase Test Coverage and Ensure Product Quality in an Agile Environment. In 2016 IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW) (pp. 29-36). IEEE. doi:10.1109/ISSREW.2016.16

Bloom, D. E., Chatterji, S., Kowal, P., Lloyd-Sherlock, P., McKee, M., Rechel, B., Rosenberg, L. & Smith, J. P. (2015). Macroeconomic implications of population ageing and selected policy responses. *The Lancet*, 385(9968), 649-657. doi:10.1016/S0140-6736(14)61464-1

Bonacin, R., Baranauskas, M. C. C., & Rodrigues, M. A. (2009). An agile process model for inclusive software development. In *International Conference on Enterprise Information Systems* (pp. 807-818). Springer, Berlin, Heidelberg.

Brasil, C. (2009). Decreto nº 6.949, de 25 de agosto de 2009. Promulga a Convenção Internacional sobre os Direitos das Pessoas com Deficiência e seu Protocolo Facultativo, assinados em Nova York, em 30 de março de 2007. *Diário Oficial da União*, (163).

Conforto, D., & Santarosa, L. M. C. (2002). Acessibilidade à Web: Internet para todos. *Informática na educação: teoria & prática*. Porto Alegre. Vol. 5, n. 2 (nov. 2002), p. 87-102. Disponível em: <https://www.seer.ufrgs.br/InfEducTeoriaPratica/article/viewFile/5276/3486>.

Dingsøyr, T., Nerur, S., Balijepally, V., & Moe, N. B. (2012). A decade of agile methodologies: Towards explaining agile software development. *The Journal of Systems and Software*, 85, 1213-1221. doi:<https://doi.org/10.1016/j.jss.2012.02.033>.

Erickson, J., Lyytinen, K., & Siau, K. (2005). Agile modeling, agile software development, and extreme programming: the state of research. *Journal of Database Management (JDM)*, 16(4), 88-100. Farid, W. M.; Mitropoulos, F. J. Normatic: A visual tool for modeling non-functional requirements in agile processes. In: *Southeastcon, 2012 Proceedings of IEEE*. IEEE, 2012. p. 1-8.

Fernandes, J. & Godinho, F. Acessibilidade aos sítios Web da AP para Cidadãos com Necessidades Especiais, 2003. Disponível em: <http://www.acessibilidade.gov.pt/manuais/manualv2.doc>.

Garrido, A., Rossi, G., Medina, N. M., Grigera, J., & Firmenich, S. (2014). Improving accessibility of Web interfaces: refactoring to the rescue. *Universal Access in the Information Society*, 13(4), 387-399. Disponível em: <https://link.springer.com/article/10.1007/s10209->

013-0323-2.

González-González, C. S., Toledo-Delgado, P., & Muñoz-Cruz, V. (2015). Agile human centered methodologies to develop educational software. *Dyna*, 82(193), 187-194. doi: <http://dx.doi.org/10.15446/dyna.v82n193.53495>.

Instituto Brasileiro de Geografia e Estatística – IBGE. Censo Demográfico 2010. Características gerais da população, religião e pessoas com deficiência. Rio de Janeiro: IBGE, 2012. Disponível em: http://biblioteca.ibge.gov.br/visualizacao/periodicos/94/cd_2010_religiao_deficiencia.pdf. Acesso em: 10 fev. 2020.

Kitchenham, B., & Charters, S. (2007). Guidelines for performing systematic literature reviews in software engineering. Disponível em: https://s3.amazonaws.com/academia.edu.documents/35830450/2_143465389588742151.pdf?response-content-disposition=inline%3B%20filename%3DSystematic_Literature_Reviews_SLR.pdf&X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=AKIAIWOWYYGZ2Y53UL3A%2F20200210%2Fus-east-1%2Fs3%2Faws4_request&X-Amz-Date=20200210T143824Z&X-Amz-Expires=3600&X-Amz-SignedHeaders=host&X-Amz-Signature=d3ad6865175121ba315f7abbf1755bc9c1e9d5bbda52207867364da8b7037062. Acesso em: 10 fev. 2020.

Koenig, S. T., Krch, D., Chiaravalloti, N., Lengenfelder, J., Nikelshpur, O., Lange, B., DeLuca, J. & Rizzo, A. A. (2014). Agile development of a virtual reality cognitive assessment. *Journal of accessibility and design for all*, 4(2), 53-68. Disponível em: <https://upcommons.upc.edu/handle/2099/14949>.

Luján-mora, Sergio; Masri, Firas. Integration of web accessibility into agile methods. In: *Proceedings of the 14th International Conference on Enterprise Information Systems (ICEIS 2012)*. 2012. p. 123-127. doi:10.5220/0004095001230127.

Masri, F., & Luján-Mora, S. (2011). A combined agile methodology for the evaluation of web

accessibility. In IADIS International Conference Interfaces and Human Computer Interaction (IHCI 2011) (pp. 423-428). Disponível em: <https://pdfs.semanticscholar.org/c5ad/6e73a9556bb1266e2dd91c9dc9c667fac5f1.pdf>.

Medina, N. M., Burella, J., Rossi, G., Grigera, J., & Luna, E. R. (2010). An incremental approach for building accessible and usable web applications. In International Conference on Web Information Systems Engineering (pp. 564-577). Springer, Berlin, Heidelberg. doi: https://doi.org/10.1007/978-3-642-17616-6_49.

Melo, A. M. & Baranauskas, M. C. C. (2006) Design Inclusivo de Sistemas de Informação na Web. In: TEIXEIRA, C. A. C. et al. (Org.). Tópicos em Sistemas Interativos e Colaborativos. São Carlos: SBC, (1), p. 167-212.

Prior, S., Waller, A., Black, R., & Kroll, T. (2013, April). Use of an agile bridge in the development of assistive technology. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (pp. 1579-1588). ACM. REICHLING, M.; CHERFI, S. S. Integrating accessibility as a quality property in web developments. In: Research Challenges in Information Science (RCIS), 2013 IEEE Seventh International Conference on. IEEE, 2013. p. 1-12. doi:<https://doi.org/10.1145/2470654.2466210>.

Sanchez-Gordon, S., & Luján-Mora, S. (2017). A method for accessibility testing of web applications in agile environments. In Proceedings of the 7th World Congress for Software Quality (WCSQ). Em processo de publicação. (citado na página 13, 15, 85).

Sander, M., Oxlund, B., Jespersen, A., Krasnik, A., Mortensen, E. L., Westendorp, R. G. J., & Rasmussen, L. J. (2014). The challenges of human population ageing. *Age and ageing*, 44(2), 185-187. SERRADOR, P.; PINTO, J.K. Does Agile work? — A quantitative analysis of agile project success. *International Journal of Project Management*, v. 33, n.5, p. 1040-1051, 2015. doi:<https://doi.org/10.1016/j.ijproman.2015.01.006>.

Shelly, C. C., & Barta, M. (2010). Application of traditional software testing methodologies to web accessibility. In Proceedings of the 2010 International Cross Disciplinary Conference on Web Accessibility (W4A) (p. 11). ACM. doi: <https://doi.org/10.1145/1805986.1806002>.

Silva, E.P.F.J, Barros, N.C., Melo, R. I. O, Pontes, N.K.S & Cordeiro, A.P.M. (2020). Evolução do conceito de deficiência no decorrer da história: do modelo médico ao social. *Research, Society and Development*, 9(2). doi: <http://dx.doi.org/10.33448/rsd-v9i2.1835>
Standish Group. (1995). The CHAOS report. Disponível em: <http://www.standishgroup.com>.

Standish Group (2014). CHAOS REPORT: 21st Anniversary edition. Disponível em: <http://www.standishgroup.com>.

Tiangtae, N., Ramingwong, S., Ramingwong, L., Potikanond, D., Homkong, N., & Maneerat, N. (2017). Developing Software for the Deaf Community: Conquering an Extreme Case Scenario. In 2017 21st International Computer Science and Engineering Conference (ICSEC) (pp. 1-5). IEEE. VERSION ONE. 12th Annual State of Agile Report. 2018. Disponível em: <http://stateofagile.versionone.com>.

Watanabe, W. M., Fortes, R. P., & Dias, A. L. (2012). Using acceptance tests to validate accessibility requirements in RIA. In Proceedings of the International Cross-Disciplinary Conference on Web Accessibility (p. 15). ACM. doi:<https://doi.org/10.1145/2207016.2207022>.

Williams, D., Wang, M. T., Chang, C. H., Ahamed, S. I., & Chu, W. (2014, June). Agile Development for the Creation of Proper Human-Computer Interfaces for the Elderly. In International Conference on Smart Homes and Health Telematics (pp. 29-38). Springer, Cham. doi:https://doi.org/10.1007/978-3-319-14424-5_4.

Williams, L., & Cockburn, A. (2003). Agile software development: it's about feedback and change. *IEEE Computer*, 36(6), 39-43. Disponível em: <http://i.web.umkc.edu/il8d5/CS551%5CAGile%20Software%20Development.doc>.

World Wide Web Consortium – W3C. Older Users and Web Accessibility: Meeting the Needs of Ageing Web Users, 2018. Disponível em: <https://www.w3.org/WAI/older-users>.

Yesilada, Y., Brajnik, G., Vigo, M., & Harper, S. (2012). Understanding web accessibility and its drivers. In Proceedings of the international cross-disciplinary conference on web accessibility (p. 19). ACM. doi:<https://doi.org/10.1145/2207016.2207027>.

Younas, M., Jawawi, D. N. A., Ghani, I., & Kazmi, R. (2017). Non-functional requirements elicitation guideline for agile methods. *Journal of Telecommunication, Electronic and Computer Engineering (JTEC)*, 9(3-4), 137-142. Disponível em: <https://journal.utem.edu.my/index.php/jtec/article/view/2933>.

Porcentagem de contribuição de cada autor no manuscrito

Fernando D'Agostini Y Pablos – 50%

Marcelo Duduchi Feitosa – 50%