

Accessibility Not on Demand

An Impaired Situation

João de Sousa e Silva¹, Ramiro Gonçalves¹ and António Pereira²

¹INESC TEC, Universidade de Trás-os-Montes e Alto Douro, Apartado 1013 5001-801, Vila Real, Portugal

²Departamento de Informática, Instituto Politécnico de Leiria, Leiria, Portugal

Keywords: Software Engineering, Software Accessibility, Web Accessibility, Digital Accessibility, Software.

Abstract: Digital accessibility is recognized as a fundamental tool for an egalitarian society. Nevertheless, software accessibility is an under addressed topic in the discipline of software engineering and the academy in general. As a result, its development and implementation is compromised. This problem is depicted here with the help of some experiments that shows the poor attention which is dedicated to this topic. Some hypotheses that try to explain this problem are formulated, and some possible solutions are debated. As a conclusion, some insights are given and a new possible researched avenue is presented.

1 INTRODUCTION

Accessibility in software is widely recognised as a need for people with disabilities, and something which empowers the life of this segment of the population (Sánchez-Gordón and Moreno, 2014) (Gonçalves et al., 2015). Also, it would be nearly impossible to find anyone who does not believe that implementing accessibility in software is not the right thing to do. However, reality is not kind in this area.

A very bad situation concerning the levels of accessibility in software (Gonçalves et al., 2015), both Web and native, becomes evident when living with digital accessibility as a necessity.

If we look closely at the work that has been developed, it is possible to see that some research have been made, but this is mostly focussed on Web accessibility (Sánchez-Gordón and Moreno, 2014) (Gonçalves et al., 2015) (Moreno et al., 2011). It is actually possible to see some effort in Web accessibility developments, where on the other hand the accessibility in native software seems to be more delegated to industry. Unfortunately, as a disable person who uses the Web on a daily basis, through a screen reader software, it is clear that something is still considerable wrong.

Even Web accessibility, a topic which is academically addressed, in practice is still very weak. It is possible to find several quite inaccessible Web sites (Gonçalves et al., 2012), even if some of them

can be checked by a Web accessibility validator (Rømen and Svanæs, 2012) and get an A mark. I can ensure that there are Web Sites that display the Web accessibility logo and yet are totally unusable. This leads to another issue, the usability (ISO 9241-11, 1998) (Gonçalves et al., 2013). Paradoxically, it is possible to build a technically accessible software, while at the same time it is unusable for persons with disabilities (Nielsen, 2002). This, for example, is due to an incomprehensible layout scheme, improper or wrong labelling, or simply the lack of important features, such as the inexistent access to a menu through keyboard (Braga et al., 2014). At this point, it is important to say that an automated accessibility test is very limited as it may validate user interfaces with severe accessibility problems (Braga et al., 2014).

Despite governmental legislation (Lazar and Hochheiser, 2013), it is still very common to find dismal accessibility problems. It is impossible to know how the situation would be like without this government regulation, but it is unquestionable that something is presently very wrong.

It is very important to understand this phenomenon – everybody recognize accessibility as a very important concept while at the same time it is very badly addressed -, in order to find directions to tackle the problem of lack of accessibility in software.

In this paper some facts regarding the work put into digital accessibility are depicted, some

hypotheses are formulated, and some insights are given to stimulate future researches.

2 EXPERIMENTS

In order to demonstrate the problem two experiments were made.

2.1 Searches on Paper Databases

A good indicator of how software accessibility is being addressed by the academy can be given by the number of results returned from searches made in recognized paper databases using software accessibility related keywords. Therefore, for this paper two expressions were chosen i.e. “software accessibility”, and “web accessibility”. Four searches were performed, two at Web of Science, and the other two at Science direct. There were not search constraints at all. In other words, the searches were made across the whole of the databases. Below are the results:

Table 1: Occurrences on databases of two expressions.

Expression	Occurrences at Web of Science	Occurrences at Science Direct	Date of Search
Web accessibility	724	434	6 of April, 2016
Software accessibility	23	49	6 of April, 2016

It is significant to say that the Web of Science’s universe was 132.894.950 papers. After a look at these numbers, the impression cannot be anything else than that this topic is not being appropriately considered. Since the results are so pour, it is easily possible to look carefully at the results returned from “software accessibility”. Analyzing these, it is remarkable that out of the 23 returned at Web of Science, only 15 relates to software accessibility. Some of these are about Web based applications, such as educational software, others are about a particular accessibility experiment, such as how to make a Turing machine accessible to blind users, some are the report of accessibility evaluations, and some others are regarding governmental guidelines. With Science direct, the panorama is similar, simply the percentage of results relating to digital accessibility was even lower. From the 49, only 11 are about the topic. Another fact is the most papers are from conferences dedicated to digital accessibility issue. This could be great – existence of dedicated

conferences on the topic -, but it also shows the topic’s segregation from the general software engineering discipline. Actually, digital accessibility is mostly out of the topic software engineering.

2.2 Occurrences on Books

In order to make another concrete indicator, the books Software Engineering, from Ian Sommerville (2011), and SWEBOK v3.0, from Bourque and Fairley (2014) were checked so as to find the number of occurrences of the term “accessibility”. The results were 1 in each book. Those who have already read these books, know that these results are even worse than they appear. The fact is that although the term “accessibility” is used once in each book, it does not relate to digital accessibility!

2.3 Data Discussion

These facts can have a major negative influence in spreading digital accessibility research and routines. In one hand the software engineering manuals do not talk about the topic. As a result, the topic is not appropriately taught in academic environments. And on the other hand, since there are few papers about it, this could result in any actual research becoming harder due the lack of references, discouraging researches to develop work on this topic, which has its two hands filled with problems.

3 HYPOTHESIS AND DERIVATIONS

In order to fight against this problem, it is important to find its causes and discuss them. Some reasons behind the lack of digital accessibility in general may be the following: H1 – Lack of documentation regarding software accessibility; H2 – The topic is poorly treated in scholarly environment; H3 – The academy in general thinks that everything is fine with the topic; H4 – Stakeholders do not care about the topic; H5 – Stakeholders think software accessibility requires too much effort; H6 – Stakeholders think that everything is fine with the topic.

It will be assumed that all of the given hypotheses are somehow true. This is actually fairly possible, even if not all at the same time and in every environment, but somehow they probably are all true and can be found somewhere. In fact, there are probably many more reasons, but this is our baseline at the moment.

Regarding H1, it is a fact that there is a lot of documentation regarding software accessibility. For example, World Wide Web Consortium (W3C) (2016) has a lot of information and guidelines on how to build an accessible Web interface. As for native software, it is possible to consult very well structured documentation regarding accessibility for Windows (Microsoft - Guidelines, 2016), Mac OS X (Apple Inc., 2016), Android (Developer.android, 2012), iOS (Apple Inc. 2012), GNOME interface (The GNOME Project, 2014), etc. That documentation comprises not only of basic information, but also on how to use the specialized accessibility application programming interfaces that exist for those who want to build graphical user interface components from scratch. This information is available online, free of charge, at the official developers' Websites. Also, there are several governmental guidelines and some ISOs that try to offer directions regarding the building of an accessible software and these are, again, online and available at official sources. But in this particular case, just the governmental information and guidelines are free of charge, not the ISOs.

Although the information exists, it may be still an issue, and this leads us to other possibilities, as following: H1.a – Developers do not know about this information; H1.b – Developers do not understand the information; H1.c – Developers are demotivated by the information complexity.

As for H2, it is a fact that accessibility does not have a big role in academic curriculums. The author's experience indicates that accessibility is poorly mentioned in general computer sciences courses. It is possible to speculate about the possible reasons, as following: H2.a – The absence of information in the used bibliography; H2.b - Teachers and/or people with decision capacity are not aware of the absence; H2.c – Teachers and/or people with decision capacity are not trained on the topic; H2.d – Lack of time to dedicate to the topic.

As for H3, it is possible that many people who take part in the academy in general think that everything is OK regarding digital accessibility. That can be due to several reasons, stated as following: H3.a – Wrong perception due to the existence of governmental legislation regarding the topic; H3.b – Wrong perception given that it is possible to find the accessibility logo while surfing the Web.

Regarding H4, it is a fact that a lot of software is totally inaccessible. This is also very true of the corporate Websites (Gonçalves et al., 2012) (Gonçalves et al., 2013). Therefore, we can speculate about some reasons for this, as following: H4.a – Stakeholders think disabled people won't use their

software.

Some possibilities to explain H5 can be derived from the ideas stated at the H4 expansion: H5.a – Stakeholders think disabled people are not an interesting target; H5.b – Stakeholders think that it is too expensive to implement accessibility; H5.c – Stakeholders think that disabled people do not are not a source of revenue.

As for H6, some reasons may be derived: H6.a – Stakeholders think accessibility is done automatically, without special request or care; H6.b – Stakeholders order accessibility features but these are badly implemented.

4 DISCUSSION AND INSIGHTS

There are many things that can be done, by all agents, to improve accessibility in software. Education is surely a major key stands (Fuentes et al., 2012). If students do not learn why accessibility is important, and how to implement it, nothing can be done. For this, general learning manuals must address it – e.g. those of software engineering, informatics' principals and user interface design. This however, should be regarded as a possible solution for future and upcoming software developers. But now, it is mandatory that the currently active teachers and people with decision capacity get a proper formation about digital accessibility, in order to make them truly conscious about this problem, and provide them with abilities on this topic. With this, it would be possible to embed digital accessibility into other fields of computer science, so as to be possible to teach them side-by-side.

A step forward would be to make sure developers are shown the available documentation regarding accessibility, so as these may discern what is required for each type of project. Probably, many developers are ill prepared to even open a governmental regulation document or, even worse, that of an ISO, not realizing that, for the most of the cases, that documentation is in fact useless. With a better understanding, other advantages of accessibility will become apparent to developer's, such as automated software testing (Microsoft, 2016), easier software evolution, and even some totally unexplored areas – to the best of my knowledge –, such as system integration.

Another very important issue is to make the stakeholders understand that a disabled person can be a user and/or an important client of his software or platform, even in the most unexpected situations. For example, a blind person can easily need to consult a

Website of a car company – e.g. work duties or he can own a car driven by other person. Also, it is important to understand the advantages of accessibility in the longer term – as stated before –, and explain these to stakeholders.

These ideas are maybe not the perfect solution, but, it is my conviction, that the implementation of even a single one of these, would represent a big step forward.

Regarding future research, this should focus on identifying the best stages of software development methods to implement, formally, accessibility procedures. Also, a system integration using the application programming interfaces of accessibility is an area where a major advantage can be found, therefore, a research is planned to understand its feasibility.

REFERENCES

- Apple Inc., 2016. *Accessibility programming guide for OS X: about OS X accessibility*. Retrieved from <https://developer.apple.com/library/mac/documentation/Accessibility/Conceptual/AccessibilityMacOSX/>
- Apple Inc., 2012. *Understanding accessibility on iOS*. Retrieved from https://developer.apple.com/library/ios/documentation/UserExperience/Conceptual/iPhoneAccessibility/Accessibility_on_iPhone/Accessibility_on_iPhone.html
- Bourque, P. and Fairley, R., 2014 eds., *Guide to the Software Engineering Body of Knowledge*, Version 3.0, IEEE Computer Society; www.swebok.org.
- Braga, H., et al., 2014. Applying the Barrier Walkthrough Method: Going Beyond the Automatic Evaluation of Accessibility. *Procedia Computer Science*, 27, 471-480.
- Developer android, 2012. *Accessibility | android developers*. Retrieved from <http://developer.android.com/guide/topics/ui/accessibility/index.html>
- Fuertes, J., et al., 2012. Including Accessibility in Higher Education Curricula for ICT, *Procedia Computer Science*, 14, 382-390.
- Gonçalves, R., et al., 2015. AccessWeb barometer - A web accessibility evaluation and analysis platform. presented at the *INTERNET 2015 - The Seventh International Conference on Evolving Internet*, Malta.
- Gonçalves, R., et al., 2012. Accessibility levels of Portuguese Enterprise Websites: Equal Opportunities for All?. *Behaviour & Information Technology*, 31, 659-677.
- Gonçalves, R., et al., 2013. Enterprise Web Accessibility Levels Amongst the Forbes 250: Where Art Thou O Virtuous Leader?. *Journal of Business Ethics*, 113, 363-375.
- ISO 9241-11, 1998. *Ergonomic requirements for office work with visual display terminals (VDTs): Part 11 Guidance on usability*.
- Lazar, J. and Hochheiser, H., 2013. Legal aspects of interface accessibility in the US. *Communications of the ACM*, 56(12), 74-80.
- Microsoft, 2016. *Guidelines for designing accessible apps - Windows app development*. Retrieved from <https://msdn.microsoft.com/en-us/library/windows/apps/hh700407.aspx>
- Microsoft, 2016. *Using UI automation for automated testing*. Retrieved from [https://msdn.microsoft.com/en-us/library/aa348551\(v=vs.110\).aspx](https://msdn.microsoft.com/en-us/library/aa348551(v=vs.110).aspx)
- Moreno, L., et al., 2011. Toward an Equal Opportunity Web: Applications, Standards, and Tools that Increase Accessibility. *Computer*, 44(5), 18–26.
- Nielsen, J., 2002. *Foreword*. In: *Maximum Accessibility: Making your Website more Usable for Everyone*. Boston, MA: Addison-Wesley Professional, xix–xxi.
- Rømen, D. and Svanæs, D., 2012. Validating WCAG versions 1.0 and 2.0 through usability testing with disabled users. *Universal Access in the Information Society*, 11, 375-385.
- Sánchez-Gordón, M. and Moreno, L., 2014. Toward an Integration of Web Accessibility into Testing Processes. *Procedia Computer Science*, 27, 281-291.
- Sommerville, I., 2011. *Software Engineering* (9th ed.). Boston: Addison-Wesley.
- The GNOME Project, 2014. *Guia de acessibilidade para desenvolvedores GNOME*. Centro de programadores GNOME. Retrieved from <https://developer.gnome.org/accessibility-devel-guide/>
- W3C, 2016. *Accessibility - W3C*. Retrieved from <https://www.w3.org/standards/webdesign/accessibility>