

Enclosure 1

Assuring accessible usability in desktop, web, and mobile software:

Non-technical guidelines and acceptance testing.



Allan Milne

Senior Lecturer in Software Engineering

SSET – School of Science, Engineering and Technology

Abertay University

Assuring accessible usability in desktop, web, and mobile software:

Non-technical guidelines and acceptance testing.

Why accessible usability is important.

To be a universal and inclusive provision, software applications require to be accessible and usable without barriers. The responsibility for achieving this lies with all stakeholders within an organisation; procurement officers, commissioning/operating staff, clients/customers and support volunteers, as well as the technical developers of the system. All stakeholders should understand the role of accessible usability and their individual responsibility for achieving it.

If a product or service is provided to users through software, be it a desktop application, web site or mobile app, then there are moral, legal and business reasons why this must be presented in an accessible and usable manner.

Technical guidelines are well-established with regard to development practices for accessibility as they apply to IT staff but this may lead to non-engagement by non-technical staff who commission and operate such systems; indeed sometimes technical guidelines can create barriers to any such engagement. The guidelines and acceptance testing presented here reflect the importance of non-technical staff in providing effective, accessible and usable systems through highlighting specific areas for engagement. This document may also suggest mechanisms for such stakeholders to actually take the lead role in supporting accessible usability through the powers implicit in their procurement, commissioning and operational responsibilities.

The aims of the guidelines and acceptance testing are to ...

... contribute to an organisation meeting its moral, legal and equality obligations with regard to non-discrimination on the grounds of disability and/or age;

... enable non-technical stakeholders to actively engage in the provision of accessible & usable software systems; and

... provide specific evidence of whether or not a software system is accessible & usable.

Terminology - usability, accessibility and accessible usability.

Usability : a system is effective, efficient and satisfying in its use (ISO).

Effective – easy to learn and remember.

Efficient – quick to navigate and do the required task.

Satisfying – no frustrations or misunderstandings.

Accessibility : the ability of a user with a disability to perceive, understand, navigate and interact with the system; perhaps through the use of some assistive practice or technology.

Disability - may be visual, hearing, cognitive or motor impairment.

Assistive practice – changing colour and/or contrast; changing text font and/or size; use of keyboard only (no mouse); change of repeat key status; sticky keys; mouse resolution.

Assistive technology – screen magnifier; text to speech; screenreader; braille display; head pointer; special keyboard/switches; special control/input devices (e.g. mouth operated).

Accessible usability : the ability to do useful work in an efficient, effective and satisfactory manner through the medium of an assistive practice or technology.

Procurement guidelines.

- ✓ Include accessible usability as a non-negotiable requirement in any contract.
- ✓ Require evidence from the software provider that their developers understand and use accepted design / implementation practices compatible with creating an accessible and usable product or service.
- ✓ Include the right to undertake acceptance testing during commissioning.
- ✓ Include the right to refuse to accept the software if it does not conform to these guidelines.

Commissioning guidelines.

- ✓ Ensure the software provides a prominent statement on accessibility.
- ✓ Perform acceptance testing (see overleaf) to provide evidence of any basic accessibility and/or usability issues.
- ✓ Refuse, using the evidence from acceptance testing, to accept any product or service that does not conform to these guidelines and/or is found to be not usable or not accessible.

Operating guidelines.

- ✓ Ensure the software provides appropriate online and offline instructions and help in accessible formats.
- ✓ Monitor the accessible usability:
 - Ensure there is a usable and prominent user feedback mechanism(s) for the user to report accessibility and usability issues.
 - Have in place a process for reviewing and responding to such feedback.
 - Have in place a process and agreement with the software developer to provide resolution of identified accessibility and usability issues.

Acceptance testing for accessible usability.

As general testing criteria, a user should be able to ...

- ✓ ... perceive all aspects of the interface;
- ✓ ... understand the content;
- ✓ ... move efficiently around the screen/page using the keyboard only;
- ✓ ... know where they are; and
- ✓ ... know what to do next.

Testing process.

1. Identify a suite of typical work tasks; these should include both internal (employee-based) and external (client-based) tasks since accessible usability applies to both.
2. Identify personnel to action the testing; this might include clients with accessibility needs.
3. Action the testing by performing the identified tasks from 1 and evidencing whether or not they meet the general criteria through applying the specific tests identified below.
4. Summarise the evidence and thus make a decision on the accessible usability of the software.

✓ The user can perceive all aspects of the interface.

<i>The software remains accessible and usable when ...</i>	<i>Test by ...</i>
<p>... different devices are used with different screen sizes.</p>	<ul style="list-style-type: none"> ▪ For a desktop application or web site, change the size of the application/browser window. ▪ For a mobile app or web site, use different devices with different screen sizes. <p>Check that content is not lost on the right hand or bottom of the window/screen. Check that the text does not become unusably small on a smaller device.</p>
<p>... an increased text size is used.</p>	<ul style="list-style-type: none"> ▪ For a desktop application, increase text size in windows settings. ▪ For a web site, increase text size in the settings of the browser. <p>Check that content does not become lost, overlapping or confused.</p>
<p>... a screen magnifier is used.</p>	<ul style="list-style-type: none"> ▪ Start a text magnifier: <ul style="list-style-type: none"> ○ For iPhone, switch zoom on under 'Settings>General>Accessibility>'. ○ For windows, type in 'magnifier' in the search box. <p>Check that the content is logically positioned in that you only have to move the magnifier in one direction to move from element to element when undertaking a task.</p>
<p>... the browser/windows/device colour scheme is changed.</p>	<ul style="list-style-type: none"> ▪ For desktop applications, change the windows colour scheme to high contrast white on black. ▪ For web sites, change the browser colour settings to ignore all user-provided colours and use white text on black background. <p>All text, including headers and labels, can still be read.</p>

✓ The user can understand the content.

<i>The software provides ...</i>	<i>Test by ...</i>
... content written in clear and simple language.	<ul style="list-style-type: none"> ▪ Identify typical users to read samples of the web page content. <p>Check that the users understand the text. Check that users can identify any technical terms, acronyms and abbreviations used.</p>
... alternative text for all graphic elements.	<ul style="list-style-type: none"> ▪ Identify the graphic elements in the user interface. ▪ Move through each graphic element so that they get focus in turn. <p>Check that alternative textual description is provided for all the graphic elements when they get focus.</p>
... appropriate labels for all form elements, including buttons.	<ul style="list-style-type: none"> ▪ Identify all form elements such as text boxes, radio buttons, list boxes, buttons, etc. ▪ Move to each form element so that they get focus in turn. <p>Check that all form elements have a label associated with them. Check that a button has appropriate text associated with it when it gets focus.</p>
... audio or textual description for video elements.	<ul style="list-style-type: none"> ▪ Identify and move to an element that plays a video. ▪ Play the video. <p>Check that there is either embedded audio description in the video or an alternative textual description is available.</p>
... captioning for audio elements.	<ul style="list-style-type: none"> ▪ Move the focus to an element that plays a video. ▪ Play the video. <p>Check that there is embedded text captioning available.</p> <ul style="list-style-type: none"> ▪ Move the focus to an element that plays audio only. <p>Check that there is alternative media available either via a text description or a full transcript for that audio element.</p>

✓ The user can move efficiently around the screen/page using the keyboard only.

<i>The user interface ...</i>	<i>Test by ...</i>
<p>... provides a logical tabbing order and a clear indication of which element has current focus.</p>	<ul style="list-style-type: none"> ▪ Put the mouse away and do not use it. ▪ Press the <tab> and shift-<tab> keys. <p>Check that the <tab> key is moving through the screen/page elements in a logical order that reflects the underlying meaning of the elements and their relationship to one another.</p> <p>Check that you can always identify where the focus is.</p>
<p>... enables work to be carried out efficiently without a mouse.</p>	<ul style="list-style-type: none"> ▪ Put the mouse away and do not use it. <p>Count how many <tab> keypresses are required to perform the task(s).</p>
<p>... maps meaningful actions to the arrow keys.</p>	<ul style="list-style-type: none"> ▪ Put the mouse away. ▪ Move focus to a list or combo box. <p>Check that the up/down keys either do nothing or scroll up/down through the list values.</p> <ul style="list-style-type: none"> ▪ Move focus to a text box. ▪ Type in some text. <p>Check that the left/right keys move the cursor back/forward through the characters if a text box.</p>
<p>... maps the <spacebar> to a mouse-click on the element with focus.</p>	<ul style="list-style-type: none"> ▪ Put the mouse away. ▪ Move focus to a button. <p>Check that pressing <spacebar> activates the button as if it had been clicked on by the mouse.</p>

✓ Users always know where they are.

<i>The software user interface ...</i>	<i>Test by ...</i>
... clearly indicates where a user is within the overall application.	<ul style="list-style-type: none">▪ Move to different screens/pages within the application, web site or mobile app.▪ Bring in a 3rd party to look at this screen/page. <p>Check that the 3rd party can identify where they are within the application. A common way of showing this is via a breadcrumb trail.</p>
... clearly indicates where the user is when actioning a multi-screen/page task.	<ul style="list-style-type: none">▪ Identify a task that requires several screens/pages (if there is one).▪ Move to different screens/pages within the task.▪ Bring in a 3rd party to look at this screen/page. <p>Check that the 3rd party can identify where they are within the task. A common way of showing this is through some indicator of the form 'page 2 of 3'.</p>

✓ The user always knows what to do next.

<i>The user interface clearly identifies ...</i>	<i>Test by ...</i>
... form fields that require to be actioned.	<ul style="list-style-type: none">▪ Identify a text box that requires to be entered in a certain way.▪ Type in an invalid value. <p>Check that this invalid status is not indicated solely by text colour but that the text box label or some other element clearly identified the form element requiring to be edited.</p>
... any information, warning and error messages.	<ul style="list-style-type: none">▪ Identify examples of each of the message categories. <p>Check that the text of all informative, warning or error messages explicitly states their purpose, rather than relying on some colour change or position.</p> <p>For example “Error: the age must be greater than 21.” Is preferable to ‘ Age less than 21’.</p>