

## Anu Ramanathan's notes on the Jenkins User Handbook:

	OBSERVATIONS	ISSUE GROUPING	SUGGESTED CORRECTIONS	REASON(S) FOR CORRECTION
1.	The User Handbook link is lost in the sea of links in the left column	UX Usability Presentation Styling  Content	<ul style="list-style-type: none"> <li>Put the User Handbook and its tree of links into its own section in the left hand pane for main TOC</li> <li>I'd recommend ditching the separate "Overview" page in the TOC and have the "User Handbook" entry be a link that takes you directly into Overview</li> </ul>	This makes both the main page of the User Handbook more easily found, as well as allows users to jump directly to their topic of choice without having to read through an entire set of text
2.	The User Handbook Overview page is missing internal links in the right hand pane of the page (non-standard presentation)	Architecture Templates Standards Consistency	<ul style="list-style-type: none"> <li>The User Handbook Overview page needs to be structured the same as the pages beneath, with internal links in the page showing as a structured list tree in the right hand pane</li> <li>Strongly recommend style templates &amp; style guide (both content &amp; presentation) to enforce a consistent look-and-feel</li> <li>Ideally, have templates for each portion of the page (site/section/page titles, breadcrumbs, left column, right column, content areas, etc.)</li> </ul>	<ul style="list-style-type: none"> <li>Following a clearly defined standard for content &amp; presentation ensures consistency of look-and-feel across an entire help section (if not the entire site itself)</li> <li>Important that typography and style remain consistent for the user to learn about the product and how to interact with it, rather than puzzling out why some help pages look one way &amp; others don't</li> </ul>
3.	There is some overlap in content between the main page to the Getting Started with the Guided Tour, and the User Handbook's overview page.	Content Maintenance Architecture Standards UX Usability Consistency	<ul style="list-style-type: none"> <li>Use single-topic authoring style to ensure no overlaps in content</li> <li>Use page content insertion ("composition") in the content area to show that said content lives in different pages in the left hand tree, if it is necessary to keep the current look-and-feel of the help content</li> <li>Text inserted from another source in the TOC could be made into a link to the page in addition</li> <li>Have a section in the right hand pane of "Related Links" and put the links embedded in the page content there as well</li> </ul>	<ul style="list-style-type: none"> <li>If content is copied instead of referenced in multiple locations, maintenance becomes a nightmare, since for every single topic, we'd need to ensure that all copies are found and kept updated accurately (consistency of information across the TOC)</li> <li>Single topic authoring style ensures that for a single topic, there's only one page where the content lives, and only that topic is covered within that page.</li> <li>We can have scripting code to control which sections of said topic are displayed based on user permissions and/or where the user is in the page hierarchy and/or for other reasons</li> </ul>
4.	Dense paragraphs of text	UX Usability Presentation Styling	<ul style="list-style-type: none"> <li>Break up dense blocks of text into paragraphs, use diagrams or other content modes to improve user's ability to actually want to read &amp; understand the content</li> </ul>	<ul style="list-style-type: none"> <li>Most people are not used to reading dense blocks of text in today's hyper-connected world</li> <li>If users won't read your documentation, they won't use your product</li> </ul>

	OBSERVATIONS	ISSUE GROUPING	SUGGESTED CORRECTIONS	REASON(S) FOR CORRECTION
5.	No breadcrumbs	UX Usability Display Areas Presentation	<ul style="list-style-type: none"> <li>I'd recommend a "where in the TOC are you" and a "back to where you came from" standard set of breadcrumbs be added to the display areas that are available for documentation</li> </ul>	<ul style="list-style-type: none"> <li>Breadcrumbs can be very helpful for users to get around in documentation – sometimes when they jump away from a page, they have no idea of how to get back and end up wasting time trying to find their way back</li> <li>Having a "here's how you get back to where you came from (in the TOC)" be a breadcrumb trail, as well as the standard "here's where you are in the TOC" especially helps users who are completely new or unfamiliar with the product.</li> </ul>
6.	Help presentation	UX Usability Presentation  Philosophy of Help Architecture Write-Once- Compile- Everywhere  Content Consistency Architecture Using build commands  <b>Security</b>	<ul style="list-style-type: none"> <li>I strongly recommend that the "write-once-compile-everywhere" philosophy be applied to the information architecture and help content</li> <li>Use build commands to send output to different types of executables (DOCX, PDF, HTML, CHM, etc.)</li> <li>Following on that, for help presentation, use a single-page-architecture that does not expose your file structure to hackers (especially for the web, but SPA can be adapted for other documentation output streams as well)</li> </ul>	<ul style="list-style-type: none"> <li>I'm a big supporter of the "Write-once-compile-everywhere" school of writing regardless of if it is code or documentation</li> <li>Applying this philosophy combined with Single-Topic authoring will ensure that there is one and only one place where any specific piece of knowledge/information resides</li> <li>Both these approaches – write-once &amp; single-topic authoring – require strrrict adherence to metadata tags &amp; standards</li> <li>Having a common set of metadata tags attached to each unique piece of information will allow for quicker "See Also" links to be found for information residing within this help documentation</li> </ul>
7.	Versioning of content files and compiled outputs	Source Control  Continuous Integration (CI) & Continuous Delivery (CD)	<ul style="list-style-type: none"> <li>Implement source control to streamline documentation output with the version of the product that is in a CI/CD (DevOps) environment               <ul style="list-style-type: none"> <li>Keep help source files (content files, project file, build files, etc.) in a source control repository for traceability</li> <li>Maintain cadence with product development teams to build efficiency and ensure that help versions stay in sync with product versions (simultaneous release)</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>Along with write-once &amp; single-topic authoring, source control will streamline documentation output and provide traceability of changes in documentation requirements (maybe in response to feedback from QA &amp; production)</li> <li>Builds efficiency in an agile environment</li> </ul>