

Feature Review Checklist				
Feature Title:			Review Date:	
Reviewer(s):			Copy Reviewer(s):	
Section	Item	Y	N	Comments/Changes
Feature	Should comply with formatting standards	<input type="checkbox"/>	<input type="checkbox"/>	
	Should have an accurate business value	<input type="checkbox"/>	<input type="checkbox"/>	
	Should have the correct system actor	<input type="checkbox"/>	<input type="checkbox"/>	
	Should describe only 1 feature, this should be reflected in the file name	<input type="checkbox"/>	<input type="checkbox"/>	
	Should describe the actor's needs consistently with the scenarios	<input type="checkbox"/>	<input type="checkbox"/>	
	Should have short and sensible title and description	<input type="checkbox"/>	<input type="checkbox"/>	
	Should have consistent domain language	<input type="checkbox"/>	<input type="checkbox"/>	
	Should be organized in the correct folders	<input type="checkbox"/>	<input type="checkbox"/>	
Background	Should comply with formatting standards	<input type="checkbox"/>	<input type="checkbox"/>	
	Should place the system in a known state	<input type="checkbox"/>	<input type="checkbox"/>	
	Should be max 4 lines	<input type="checkbox"/>	<input type="checkbox"/>	
	Should have multiple scenarios	<input type="checkbox"/>	<input type="checkbox"/>	
	Should not include technical stuff	<input type="checkbox"/>	<input type="checkbox"/>	
Scenarios	Should comply with formatting standards	<input type="checkbox"/>	<input type="checkbox"/>	
	Should have 1 and only 1 Given-When-Then section (with And/Or statements)	<input type="checkbox"/>	<input type="checkbox"/>	
	Should use declarative steps	<input type="checkbox"/>	<input type="checkbox"/>	
	Should clean up the system afterwards	<input type="checkbox"/>	<input type="checkbox"/>	
	Should place repetitive checks in a Scenario Outline	<input type="checkbox"/>	<input type="checkbox"/>	
	Should be able to run twice in a row without failing	<input type="checkbox"/>	<input type="checkbox"/>	
	Should cover happy and non-happy paths	<input type="checkbox"/>	<input type="checkbox"/>	
	Should not have And/Or within the step	<input type="checkbox"/>	<input type="checkbox"/>	
	Should <u>not</u> leave data in the folder of the running tests	<input type="checkbox"/>	<input type="checkbox"/>	
	Should <u>not</u> depend on data that is likely to change in the course of normal usage	<input type="checkbox"/>	<input type="checkbox"/>	
Tags	Should be up to date (@wip, @todo, ...)	<input type="checkbox"/>	<input type="checkbox"/>	
	Should <u>not</u> use same tag for scenario and feature	<input type="checkbox"/>	<input type="checkbox"/>	
	Should <u>not</u> be on the background	<input type="checkbox"/>	<input type="checkbox"/>	

Step Definition Review Checklist				
Feature Title:			Review Date:	
Reviewer(s):			Copy Reviewer(s):	
Section	Item	Y	N	Comments/Changes
General code	Should follow coding standards	<input type="checkbox"/>	<input type="checkbox"/>	
	Should be consistent in style and formatting	<input type="checkbox"/>	<input type="checkbox"/>	
	Should be well-structured	<input type="checkbox"/>	<input type="checkbox"/>	
	Should have clear and adequate commenting in consistent style	<input type="checkbox"/>	<input type="checkbox"/>	
	Should have properly defined variables with meaningful consistent names	<input type="checkbox"/>	<input type="checkbox"/>	
Steps	Should reuse existing step definition where possible	<input type="checkbox"/>	<input type="checkbox"/>	
	Should avoid fragile definitions	<input type="checkbox"/>	<input type="checkbox"/>	
	Should use valid & robust selectors	<input type="checkbox"/>	<input type="checkbox"/>	
		<input type="checkbox"/>	<input type="checkbox"/>	
		<input type="checkbox"/>	<input type="checkbox"/>	
		<input type="checkbox"/>	<input type="checkbox"/>	
Exceptions	Should throw exceptions in the proper place	<input type="checkbox"/>	<input type="checkbox"/>	
	Should provide meaningful exception text	<input type="checkbox"/>	<input type="checkbox"/>	

You can use generic code review checklists:

- [https://www.liberty.edu/media/1414/\[6401\]code_review_checklist.pdf](https://www.liberty.edu/media/1414/[6401]code_review_checklist.pdf)
- <https://courses.cs.washington.edu/courses/cse403/13sp/lectures/code-review-checklist.pdf>

You can also use programming language specific checklists:

Ruby:

- <https://reinteractive.net/posts/126-checklist-for-a-rails-application-code-audit>
- <http://matthewpaulmoore.com/post/5190436725/ruby-on-rails-code-quality-checklist>
- <http://www.ultrasaurus.com/2010/01/rails-security-review-checklist/>

Python:

- http://pycogent.org/coding_guidelines.html

C#:

- <http://www.codeproject.com/Articles/593751/Code-Review-Checklist-and-Guidelines-for-Csharp-De>
- <http://weblogs.asp.net/tgraham/44763>