

# Agile Review

This check list could be used prior to a project starting (to determine if the Agile Process is suitable and how it might be applied) and during a project to assess the practices in use. The objective is not to provide a “write/wrong” assessment but to stimulate discussion on how the project is operating and where it could be improved, perhaps as part of a retrospective/end of iteration.

The review is broken down into a number of sections. Each section describes a number of Agile features. Some features that could appear in more than one section have been limited to just one to prevent duplication.

## 1. Requirements Gathering

<i>Agile Feature</i>	<i>Project Statement</i>
Did the project have a Discovery/Prestudy Phase? What artefacts were produced, who reviewed them, what feedback was given to the team?	
Who is the main business sponsor?	
How often does the project engage with the main business sponsor?	
Does the project have access to other business representatives?	
How does the project engage with these other representatives?	
How are changes (new features or unwanted features) identified?	
Non functional requirements – have been identified, documented – where visible?	
Application Architecture? Defined up front or determined on the fly?	
What Elements of the project are fixed? Features Time Resource/Cost	
Are the business benefits of the project clearly understood and at what point does the project become too expensive to produce a return on the investment.	

## 2. Iteration Planning

<i>Agile Feature</i>	<i>Project Statement</i>
Is the Project using stories?	
Granularity of stories (too high, to granular)	
How are stories managed? (Cards on board/wall, software tool)	
Key Performance Indicators Used:	
Velocity  (Include stories completed, retired, added, new estimates – are technical stories included? Are only completed stories counted towards velocity)	
Bug Count	
Code Quality (subjective or objective measure (Copy Paste Detector))	
Number of Builds/Good Builds/Build Time	
Unit Test Code Coverage	
Regression Test Suite run & % Passing	
Others?	
How are stories prioritised? (Business priority?)	
Does the BA record additional information on the story card and story management system? (e.g. programmers/QA engineers notes, key points made during conversations).  Lesson learned from projects; story notes are good.	
Does the BA (or story manager) work on requirements for an iteration ahead of the iteration?	
Prioritisation of Technical Debt?  (Is technical debt recorded, if so as a story or bug?)	
Typical Iteration Length – do they vary, if so why?	
How are Stories Estimated?  Spikes, prior work on other projects or this project? Comparison between stories of equal complexity.	
Are the minimal features required or a usable	

<i>Agile Feature</i>	<i>Project Statement</i>
delivery known? How arrived at/validated? Critical path through system (wide coverage shallow feature set or narrow coverage with many features?)	
For modules/features yet to come, are screen shot story boards used to facilitate workshops and brief programmers etc?	

### 3. Communication

<i>Agile Feature</i>	<i>Project Statement</i>
Are daily stand ups held? (Length)	
Planning Game Are they held? Who Attends? (whole team ideally + Customer) Typical Duration?	
End of Iteration/Retrospective?	
Use of Wiki and Project Forum to share data and other information	
Use of notice boards?	
Huddles at start of a story (BA/Customer, Code and Test Engineers)	
Does the QA Engineer and BA review stories during coding and again prior to hand over for formal QA.	
Reporting to business and other interested parties (e.g. Project Office). Format and value?	
JAD Workshops with users?	
Email? (does everybody have it, do they need it?)	
Ability to access reference materials on business and technical subjects (e.g. via Google or JavaWorld)	
Communication to related teams (e.g. production) and other programme teams. How and frequency.	
Do the team ever use Quality Circles to review progress.	
Is the team located in the same workspace (if in	

<i>Agile Feature</i>	<i>Project Statement</i>
the same building)	
If the team is located in different locations/timezones, how is interactive communication achieved?	

### **3. Code & Build**

<i>Agile Feature</i>	<i>Project Statement</i>
Is a source management system used? (CVS, Subversion)	
Automated Build System?	
Who owns and maintains the build system?	
Does the build system produce a ready to deploy application?  (Important so allow anybody to take and use any build; particularly for projects where there are limited options to put iteration deliverables into production)	
Are builds version numbered?	
Can a build be recreated several days after being delivered?	
How often does the build run?  (How long does it take?)	
How often do engineers check in?	
Prior to check in, Do engineers have to <ul style="list-style-type: none"> <li>a) Ensure their code is synchronised with other team members?</li> <li>b) Ensure that is passed all units tests &amp; compiles locally?</li> </ul>	
Is pair programming used (intelligent or all the time). If not, how is knowledge of the code base shared to reduce dependencies on one person?	
Are any coding standards/conventions used?  (If so how monitored and where are they set out (e.g. Wiki)).	
How are areas for refactoring identified?	
How is refactoring managed ? (e.g. adhoc as programmers find it or by a story card included as	

<i>Agile Feature</i>	<i>Project Statement</i>
a technical debt story?)	
Is the application design continually reviewed?	

#### 4. Test

<i>Agile Feature</i>	<i>Project Statement</i>
Is the QA Engineer involved in the planning game?	
Does the QA Engineer discuss/review unit tests with the coding Engineers?  (Part of code quality/assessing value of unit tests, communication on how features have been implemented) If so, how recorded (ISO/CMMI certification)	
Are regression tests included the automated build process?	
How often are they run outside the build?	
Does the BA or Customer Representative review work before being handed over to QA for test?	
Does the BA and/or Customer Representative help devise Regression Tests	
How are defects recorded?  (On card and in Bug Tracking System)	
Do Engineers take bugs in preference with new stories?	
Are manual QA Scripts maintained	
Are QA Artefacts (Test Data, scripts, notes etc) stored in a repository?	
Who conducts UAT?	
Are UAT written by the business users?	
Do users review the software on a frequent basis?	
Are there any performance targets for the projects deliverables?	
How and by whom are deliverables measured for performance?	

#### 5. Release & Project Documentation

<i>Agile Feature</i>	<i>Project Statement</i>
<p>Are release and install instructions produced for each release?</p> <p>Who is responsible for producing these?</p>	
<p>Are any user instructions/help materials produced, if so by whom and when (during the iteration?)</p>	
<p>Are releases packaged into an installer for ease of deployment by users?</p>	
<p>Does the install package/release include any tools for ease of upgrade from earlier versions?</p>	
<p>What documents are produced by the project?</p> <p>Why are these produced (company standards, somebody will use them – a project should only product material that will be of value to somebody).</p> <p>End of Projects Docs: Functional Overview, Business Processes used in the software, ERD, Use Cases, Architecture Diagrams, Configuration Diagrams, plus all release notes and end user docs.</p>	
<p>Are the contents of the Wiki/Intranet page for the project considered project documentation?</p>	
<p>After the project has been completed, who will maintain the deliverables and what documentation/hand over process will they require?</p>	