# How Agile are you? (aka. The Agility Index)

**My Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Total: \_\_\_\_\_\_\_\_**

**Email: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Twitter: \_\_\_\_\_\_\_\_**

|  |  |  |
| --- | --- | --- |
| Do You…? | Available Points | My Points |
| **Team** | | |
| Have a product owner (only 1) | 1 |  |
| Do daily stand ups at the same time | 1 |  |
| Have a scrum master | 1 |  |
| Management support of Agile | 1 |  |
| Have a product owner who maintains the backlog | 2 |  |
| Have a product owner who maintains the backlog in priority order | 2 |  |
| **During the sprint** | | |
| Sending Done’s to product owner – <http://bit.ly/gnOW0q> | 1 |  |
| Sending Done’s with Screen captures | 1 |  |
| Sending Done’s with a video recording | 3 |  |
| Keep a burn-down daily – manually | 1 |  |
| Keep a burn-down daily – automatically – and review at stand up <http://bit.ly/eROJET> | 2 |  |
| Measure Cycle Time (the time to start and end a task) | 1 |  |
| Task board (or an Electronic Task Board) to visualize the flow eg. Urban Turtle, TeamPulse, AgileZen, Stories Overview Report | 2 |  |
| Sending at least 1 ‘Done’ everyday (this indicates good sizing of tasks) Note: If sending ‘Done’s per story then it is OK every 2 days | 1 |  |
| **End of Sprint** | | |
| Have your meetings - review/retro/planning | 1 |  |
| ..... on an exact cadence eg. 2 weeks cadence does not change | 1 |  |
| Have a scribe during a meeting eg. updating the work items real time | 1 |  |
| Do you size tasks in hours/days (not recommended) or story points | 1 |  |
| Do you know your completed story units per sprint (in points) | 1 |  |
| Do you know your completed velocity (with story points) | 1 |  |
| Devs Team Room (if remote, sharing screens for most of the day – almost pair programming eg Teamviewer, Mikogo) | 1 |  |
| Pair Programming | 1 |  |
|  | | |
| **Requirements (for Project Managers/ScrumMasters)** | | |
| Mockups aka wireframes eg. Balsamiq, Sketchflow, Visio <http://bit.ly/dEEIMa> | 1 |  |
| Do you show a Roadmap eg. Hierarchy, epics, stories | 1 |  |
| Investigation tasks eg. Discovery tasks for work in future sprints (aka Spikes) | 1 |  |
| Do you give the team some bonus tasks (aka Stretch Goals) | 1 |  |
| Visualization of Requirements Progress eg. Change Management Report (like a diff on source control) – TeamPulse has it | 4 |  |
| Track issues and risks eg. ‘For the record’ this is a bad approach eg. We will not make deadline because servers are delayed eg. I have reported a bug in the datagrid to Telerik (not in emails only) | 1 |  |
| **Engineering (for the Developers)** | | |
| Using source control | 1 |  |
| Check-in policy - Using source control + associating checkins with Work Items  Consistent traceability eg. Story to Tasks to Code eg. all associated with work items ID (Note: Easier in TeamPulse) | 1 |  |
| Unit tests on at least fragile code | 1 |  |
| Unit test mind set (eg. you repro a business logic bug first, with a failed unit test, before fixing that bug) | 2 |  |
| Code coverage statistics | 1 |  |
| Test driven mind set (using BDD aka Test First) (pex/moles) | 4 |  |
| Emergent Architecture (eg. incremental design like not all tables in one go aka Vertical slicing) | 1 |  |
| Have a build in one step eg. a build.bat | 1 |  |
| Have a build server – aka continuous integration | 2 |  |
| Have ‘Continuous Deployment’ to a ‘Test Server’. You are now Kanban ready (not applicable for Windows apps) | 4 |  |
| Using test cases - Using work items + acceptance criteria + test cases (that tell a tester ‘that story’ is ready for testing) aka testing as part of the team | 4 |  |
| Check-in policy – enforcing test cases | 1 |  |
| **Testers** | | |
| Automating manual tests to basic functional tests (web tests) | 1 |  |
| Using advanced functional tests to check UI interactivity (coded UI tests or Telerik tests) | 1 |  |
| Bugs - Following a consistent bug definition (what goes in as a bug and task) | 1 |  |
| Bug triaged (eg. P1-P4) | 1 |  |
| Give feedback with balloons <http://bit.ly/fuLbJd> | 1 |  |
| **Data Schema** | | |
| Schema Master – one person who makes schema changes | 1 |  |
| Automatic schema changes by the app (eg. Create, Update, Reconcile buttons – not manual deploy schema via .sql files) | 4 |  |

**(Put your total at the top)**

Ps: Tweet your results eg I just scored XX in the #sswAgilityIndex @AdamCogan

Created by: Adam Cogan (SSW) and Joel Semeniuk (Telerik) for the Moving the Titanic SharePoint Tour

<http://bit.ly/hddwVW>