Juhani Välimäki 20.11.2017 (version 0.5)

This template is based on my work only and thus is not going to be the universal truth about the topic. Thus you ought **not to use** this **as it is**. Always adapt the Definition of done to your organization (,project,) and needs!
But at least this gives you one example. And I have seen a lot of worse DoDs in use.

DEFINITION OF DONE

1. *Analysis and requirement definition task (prework or daily work)*

(E.g. User Story elaboration, Entity-Relationship diagram (with UML class notation or other)
- it's verified against the problem description / customer's needs.
- the objective is clear for all team members
- validated by all the team members and the product owner
- shared via common repository/storage like agreed (GitHub, TFS, Moodle, Shared disk, …)

2. *UX/UI design task (Mockup/ Real UI screen/ Interaction)*

- it contains all the information needed for each functionality
- it contains all the controls needed for each functionality
- not too much information per screen (Mockup, Real UI screen)
- usability accepted by product owner (Real UI screen/ Interaction)
- it fulfills the requirement definition
- validated by another team member
- shared via common repository/storage like agreed (GitHub, TFS, Moodle, Shared disk, …)

3. *Design tasks*

(E.g. Architechture diagrams, Database diagram)
- it is clear how to implement the design
- describes exactly so that first working version can be implemented
- verified by another team member
- shared via common repository/storage like agreed (GitHub, TFS, Moodle, Shared disk, …)

4. *Implementation tasks*

- architecture has been followed (implementation is consistent with other parts)
- all the controls work properly
- no errors, no warnings
- provides the requirement functionality
- follows XYZ coding conventions: e.g. variable naming, blocks,
- follows the General requirements (from Product/Release/Sprint backlog)
- integrated to the shared project build (nightly build/weekly build)
- tested with sample data and test cases
- verified by another team member
- shared via common repository/storage like agreed (GitHub, TFS, Moodle, Shared disk, …)