

LINGI 2255 - Evaluation Criteria - *Agile sprint*

These are *-some of-* the criteria that will be exploited by your team managers in order to evaluate each of the Agile sprints. Note that the global appreciation will *not* necessarily be an exact average when taking each criterion separately.

<u>Ratings:</u>	A+	19-20	<i>nearly perfect</i>
	A	17-18	<i>very good, more than we hoped for</i>
	B	15-16	<i>good, exactly what we hoped for</i>
	C	13-14	<i>average, some good things but also room for improvement</i>
	D	10-12	<i>passable but not very good</i>
	E	10	<i>only just passable</i>
	F	<10	<i>should not succeed</i>

Document

- **[EN] English**
 - Is the English of sufficient *quality* to understand the document ?
 - Has *care* been taken to avoid grammatical and spelling errors ?
- **[RD] Readability**
 - Max. 10 pages
 - Only the mockups can be put in appendices
 - Is the report *pleasant* and *practical* to read ?
 - Front page containing all practical information (group number, etc.)
 - Is it clean and well-presented (good layout, numbered figures)
 - Every figures are referred in the text
 - Is the report *well structured* and well organised ?
 - Does it have a clear overall structure ?
 - Is the report more than just a bunch of diagrams or bullet points ?
 - Are the diagrams supported by relevant text *explaining* them ?
 - Are there an introduction, a conclusion, transitions between the sections ?
 - Is the report *self-contained*, such that it provides sufficient explanations and surrounding context to be understood by the reader ?
- **[TA] Targeting**
 - Is the report targeted to *both* the *customer* and the *manager* ?
 - The customer must be able to easily determine which part(s) of the document is(are) targeted to him, and to understand it
- **[RL] Relevance**
 - Are all the elements in the report *relevant* ?
 - Avoid noise
 - Should a non-mandatory additional element, although useful, be *motivated* in the text ?

Production

- **[SA] Software Architecture**
 - The report should *at least* come with an *up-to-date* class diagram.
 - If for some reasons, using a class diagram is not relevant in the current context of the Agile sprint, then another type of diagram should be motivated and exploited.
 - According to the current planning, does the class diagram (and other diagrams if any) cover all the requirements planned up to the current sprint ?
 - Is the *architecture* well-chosen ?
 - Is it modular and easily extensible ?
 - Are the architecture *choices* discussed / motivated ?
 - Are there *additional elements* in the report, such as activity diagrams, sequence diagrams, etc. allowing a better and easier understanding of the current architecture ?
 - Are all these elements consistent between each others ?
- **[IM] Implementation and testing**
 - Did you already think at a testing strategy for the final product and/or each specific requirement separately ?
 - Is a link to the current *implementation* provided ?
 - Is the code easy to *understand* and well-commented ?
 - Is the code written in a way that eases an external developer to extend or modify it ?
- **[TR] Traceability**
 - Are the *feedbacks* provided by the managers (from the previous reports) taken into account ?
 - Are all the *modifications* from the previous sprint well-notified ?
 - In the software architecture
 - In the development planning
 - In the requirement analysis
- **[CV] Current version of the Software [Sprint 3 only]**
 - Are all the *mandatory* requirements working ?
 - User *experience* of the software

Organisation

- **[PL] Planning**
 - Is the current planning *respected* ?
 - If there are *changes* in the planning, are these well *defined* in the report ?
 - Is the future planning realistic ?
- **[CM] Communication**
 - Clearly state the responsibility of each team member during the current sprint

- How *efficient* has been the communication between the team and the manager during the sprint ?
- How *clear* does the communication seem to be within the team ? With the
- Communication with the customer ?