

- Scenario Creation: Has to be added again, because currently it is not really covered
- Link use case section to the scenario pragmatics section. Define for each use case the corresponding workflow
- Unify or integrate the workflows
- Define prototypes/toolings or reference implementation for the standard without developing them. ASAM does not want to support normative software implementation. Standards are decoupled from reference implementation, because ASAM does not want to be a software maintainer. So maybe an API

Many tools (e.g. checkers, visualizers, etc.) directly support the adoption and training of ASAM standards

In future, ASAM will provide a platform for hosting complimentary, non-normative, open-source tools

- ASAM will support the development of a community around this platform
- ASAM will not become a software maintainer
- ASAM will provide gatekeeping/moderation of the platform
- Tool submission to the platform via “tool proposal” subject to approval of ASAM
- Any tools can be used as the basis for further development projects, as defined in the general ASAM project processes.

- User story vs. requirement: Are requirements necessary? Or should we stick to the user stories?
  - o Yes, both should exist, but relate to each other and be consistent. We have to derive requirements from the user stories.
- Not all user stories have to be supported from the beginning, therefore a prioritization should happen
- User stories ⇔ requirements → orthogonal
- → we have to take time-constraints into account. Relations between user stories or requirements
- User story → workflow → requirements

The steps that have to be processed during the standardization project:

- 1) Review the concept document (use case section/usage and pragmatics section)
- 2) Cluster/restructure the user stories/use case examples (→ e.g., grouping according to the feature sets)
- 3) Link all user stories/use case examples to corresponding content from the usage & pragmatics section if possible
- 4) Define the workflow for each user story/use case example
- 5) Decomposition of workflows into subworkflows
- 6) Identify redundant subworkflows/workflows
- 7) Try to create a generic workflow that is consisting of the subworkflows and capable of covering most of the user stories/use case examples
  - a. Any gaps in the workflow, that are identified in the standard should be communicated to the other working groups, so that they are able to fill the gaps.
- 8) Document non-covered user stories/use case examples and define working packages

- 9) Document the generic workflow and link all user stories/use case examples appropriately
- 10) Review and discuss key requirements and decide about a reference implementation/API/interface...
- 11) Requirement for all other points above: Reusability of subelements from a scenario should be possible (→ Siddartha)

Deadline: Mid of April (detailed version of the 11 points), End of April (Proposal ready to go to TSC)

This working package will come up with guidelines based on the user stories, it is not a normative standard, it is a guideline for the usage workflow.