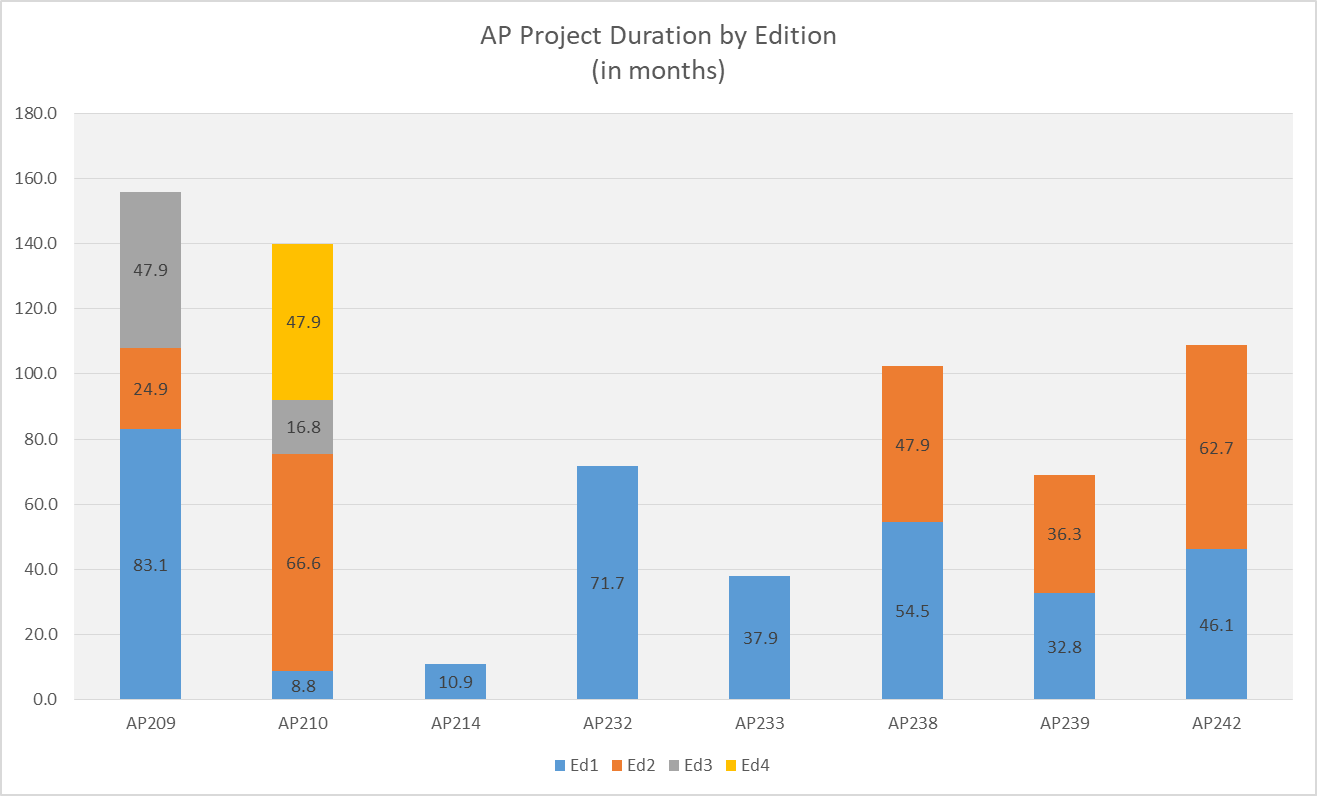
**Current State**

According to survey results of a study by the Institute of Electrical and Electronic Engineers “a majority of respondents' organizational units are using agile and/or lean methods (58%). Furthermore, lean appears as a new player, being used by 24% of respondents, mainly in combination with agile (21%).” These statistics reinforce the increased development rate of industry towards a trend of rapid incremental development as they strive towards enabling the digital threads for their enterprises.

ISO standards development follows six stages; proposal (10), preparatory (20), committee (30), enquiry (40), approval (50), and publication (60) stages (target date planner iso.org). These stages control the development process entry and exit criteria from the proposal for a new work item, through the technical development of the source models, draft review, to publication. An analysis of ISO project metrics on a sample across 8 standards and 16 edition publish cycles revealed the current average project duration is 43.5 months for the release of an edition of a standard. This calculates the time from stage 10.99 of New Project Approval to stage 60.60 International Standard Published. Additional analysis reveals this is an average project length growth of 4.5 months from edition 1 lifecycle length to that of edition 2 lifecycles.



*Figure 1: Average project duration by edition (data based on ISO.org project metrics). Note these metrics are based on a status range from stage 10.99 (new project approval) to 60.60 (international standard published). Note this is a truncated view and does not include the white paper authoring process which can add as much as an additional year to project duration.*

This trend of project length growth for standards is at odds with the industry’s need for faster releases of incremental functionality. ISO’s 2016-2020 strategic directions include six tenants two of which are (1) “Develop high-quality standards through ISO's global membership”, by ensuring we effectively (2)“ Engage stakeholders and partners ”. In order to support industry demand and maximize the benefit of standards the quality and length of development of standards must be addressed.

***Development Lifecycle Management***

There are two primary contributing factors causing an extended development cycle:

1. Failure to gain consensus [International Agreements]
2. (2) Resource commitment/management.

In the course of STEP development once a new work item is approved or a defect raised, most teams use a system such as Bugzilla for coordination. Bugzilla was originally designed as a general purpose bug tacking and testing tool (this is from wiki, find a better source) however in STEP development it has been stretch beyond that scope. In existing STEP development application it is used as a requirements management, collaboration /consensus tracking, issue management, and task management system in addition to for version control of source models. This leads to much confusion, maintenance, and misuse of Bugzilla. The comment field is used for documenting proposed solution and consensus management. This undisciplined and unintended approach with Bugzilla results in extended durations to determine solution concept consensus and failure to identify lack of consensus until issues are assigned and often times under development. The result is extended development cycles and many times rework during the committee stage 20.

Resource commitment for the performance of development work is also a primary obstacle.

1. Volunteer staff
2. Lack of resources (and shared by multiple projects)
3. Unreliable SOW/time estimates
4. Virtual distributed team

***Quality Management***

Another issue facing the current development lifecycle is that of quality/completeness concerns of the published standards. CAx-IF, which is a joint testing forum between AFNet, PDES, Inc. and prostep ivip tasked with testing CAx STEP translators quality, has continually reported issues of implement-ability of the standards. These issuesfall into three areas and can be categorized as data quality issues, incomplete solutions, and overtly complexities/implementable solutions. Metrics from Bugzilla show 420 quality/completeness concerning bugs have been reported by the CAx-IF against STEP. 

While measures have been taken combat complexity and streamline solutions, such as the transition from monolithic to modular architecture, the problem still persists.

There are two primary root causes contributing to the quality/completeness issues of standards:

1. Lack of knowledge
2. Lack of adequate toolchain

A robust knowledge of the STEP data model is necessary to ensure enhancements and defect resolutions have a complete end to end solution. The integrated nature of the elements of the STEP data model, ARMS, MIMs, and IRs, mandate that parallel changes take place in each of these elements to ensure continuity of the model. A lack of understanding of integration points leads to (1) inelegant solutions and (2) data quality issues. While quality checks exist to catch data quality issues these checks are not infallible. Other issues, such as the elegant or clumsy nature of solution concepts, require analysis by human interpretation and rely solely on the knowledge level of the human resource performing the work. Understanding of the development process and toolset are also factors that affect code quality. **INPUT METRICS FROM DEVELOPER** **SURVEY.**

Issues with current development lifecycle [in terms of business impact]

* + Issue 1: Development Time Length
    - Failure to gain consensus [International Agreements]
      * Poor requirements management
      * Poor documentation on agreement and priorities
      * Poor process instruction and compliance:
    - ~~Volunteer staff~~ Resource commitment/management
      * Volunteer staff
      * Lack of resources (and shared by multiple projects)
      * Unreliable SOW/time estimates
      * Virtual distributed team
  + Issue 2: Quality/Completeness of Standard
    - Lack of knowledge
      * Knowledge of data model
      * Knowledge of the development process
      * Knowledge of the tools used for development
    - Lack of adequate toolchain
      * Lack of automation
      * Technology obsolescence

References

<https://ieeexplore.ieee.org/abstract/document/6475408>

<https://www.iso.org/files/live/sites/isoorg/files/archive/pdf/en/iso_strategy_2016-2020.pdf>

<https://www.iso.org/files/live/sites/isoorg/files/developing_standards/docs/en/Target_date_planner_4_ISO_standards_development_tracks_2017.pdf>

<https://www.cax-if.org/>