CDM – Release Guidelines Task Force

Background

- In mid-May 2024, a small task force was convened under David Shone's direction to develop proposals to refine the CDM release control guidelines developed under Tom Healey based on production systems experience
- The group also looked at developing a production release schedule for 2024 and early 2025
- Participants in the task force included:
  - David Shone and Brian Lynn from ISDA
  - Chris Rayner from ISLA
  - Minesh Patel and Leo Labeis from REGnosys
  - Marc Gratacos from TradeHeader
- Brian Lynn acted as group leader and editor
- These recommendations have also been circulated with a wider set of maintainers and WG leads, and feedback has been incorporated
- The following presentation describes a summary of the group's recommendations (a more detailed version will be included in the CDM GitHub documentation).

CDM – Release Guidelines Task Force

Summary Document

- Major release scheduling guidelines
- 2024 CDM Roadmap/schedule
- Change control guidelines (principles, rules, evaluation methods)
- PR classification, review, approval guidelines
- Minor/Dev Release scheduling & release approval guidelines

CDM – Release Guidelines Task Force

Major Release Scheduling and Approval Guidelines

- Objectives of defining major releases & release guidelines:
  - To communicate to CDM users and developers when profound changes will be made
  - To support planning and adoption, through consistency and predictability

- Overarching principle:
  - The Streering working group (SWG) shall define a release schedule with clearly defined scope, to provide controlled but timely enhancements to the model

- Scheduling and approving major releases:
  - No major release will be planned/scheduled without formal approval of the SWG
  - Major releases shall be planned and reviewed at the SWG at least 3 months ahead of the anticipated release date
  - It is anticipated that for at least the next several years at least one major release will be planned each year
  - Any addition to the scope/contents of (or technical change to) a planned major release requires SWG approval
  - If planned scope items for a major release are not available in time for the planned release date, the SWG will need to decide whether to slip the release date or drop the item, based on industry priorities
  - These guidelines can be overridden in exceptional circumstances by a formal vote of the SWG

CDM – Release Guidelines Task Force

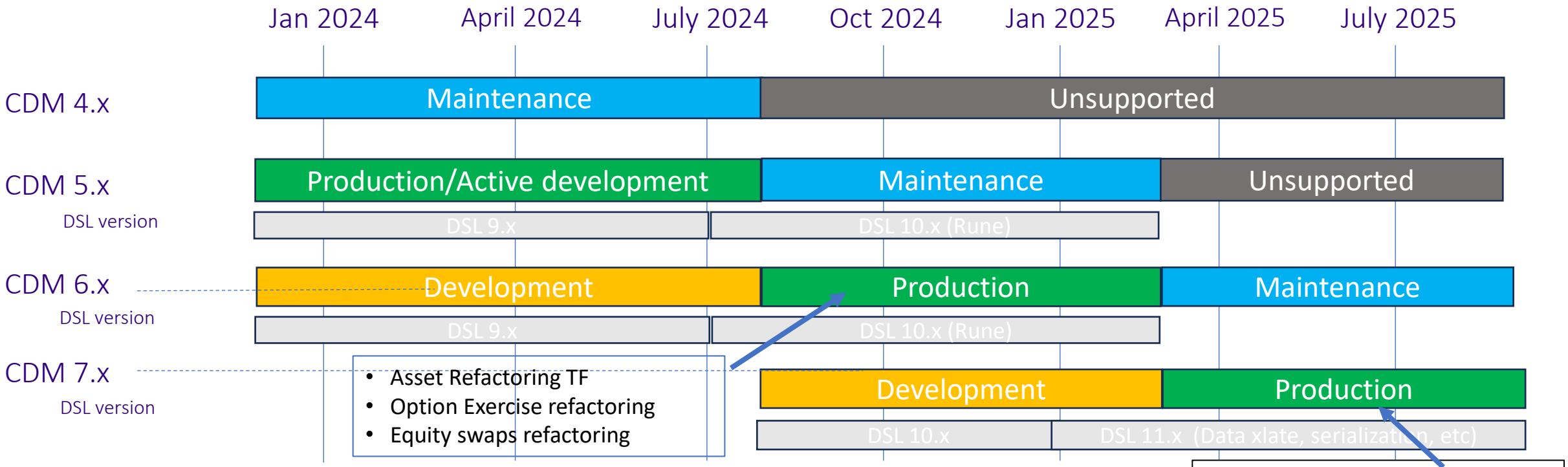
Major Release Scheduling and Approval Guidelines

- Long term planning and outreach:
  - SWG will establish plans for upcoming major releases for at least the following 9-12 months and publish them on the GitHub repository

- Types of changes allowed in major and minor versions:
  - Breaking changes can only be implemented in a major version
  - Therefore, PRs containing breaking changes shall only be approved for dev releases, for inclusion into the next major release.
  - When a major version includes breaking changes, migration guides and transition plans should be in place.
  - To facilitate this, each incremental dev release to be included in a major release should include a migration guide for any incremental breaking change it contains.

CDM – Overall Release Schedule

Preliminary CDM 2024-5 Major Release Schedule



Legend

Development	New designs released in development, not yet released to production
Production	Version in production release and continuing to be updated (backward compatibly)
Maintenance	Version will be updated with fixes and critical new features only to support major business imperatives
Unsupported	Version will not be updated but remains available for download

CDM – Overall Release Schedule

CDM
COMMON DOMAIN MODEL

Release States

- Release states are defined as follows:
  - **Development –** versions that include new designs from the "main" branch that are still under development. All tests must pass but the model may continue to evolve before being released into production.
  - **Production -** the "latest and greatest" stable version that ideally everyone should upgrade to, and where enhancements compatible with the existing models will be included. We should aim for a release to be in Production for around a year if we can, to alleviate upgrade costs to consumers.
  - **Maintenance** – when a new Production version is released then the current Production version will go into Maintenance. Only critical bug fixes and changes related to critical regulatory requirements should be ported to Maintenance releases. Otherwise, functional changes would not be ported to maintenance releases. The intention would be to have only 1 version at a time in maintenance, so each time a new Production version drops, the previous Maintenance release would go to Unsupported.
  - **Unsupported/End of Life** – There will be no bug fixes or other support for the version. TBD: We may perform security scans on some more recent unsupported versions and report any identified vulnerabilities, but will not perform security remediations.
- At any point we want a maximum of 1 centrally supported development version, 1 production version, and one maintenance version.

CDM – Release Guidelines Task Force – Change Control

Overview of Change Control Guidelines

- Principles
  - We are trying to ensure rapid, smooth, and predictable evolution of the model by controlling when and how breaking changes are introduced
  - Prohibiting breaking changes within a major version should allow users to upgrade to minor versions more quickly and easily, and plan for when to implement larger changes

- Summary of Rules
  - Within successive minor releases of the same major release, data types should continue to work in newer minor releases
    - This implies a number of constraints in what can be introduced in a minor release, e.g. don't change or remove fields (full list is longer)
  - Similarly, function signatures should obey the same constraints
  - We allow some minor exceptions to these rules for newly introduced functionality that may not be fully formed
  - Functionality shall not be removed between major versions without advance notice

- Evaluation methods
  - Developers should be aware of the guidelines
  - Reviewers and WGs should double-check that the guidelines were followed
  - There should be automated regression tests to enforce the guidelines

CDM – Release Guidelines Task Force

Summary of PR approval Requirements

- PRs shall be classified into defect corrections (bug fixes to correct existing functionality) vs. enhancements (new designs or capabilities)
- There shall be an indication of whether a PR includes any backward-incompatible changes
- Approval has to be by a separate person from the submitter (enforced by GitHub)

Approval processes and requirements

Type of PR	Backward compatible	Backward incompatible
Model - bug fix	1 maintainer – separate from the submitter, preferably from a separate organization	2 maintainers; must have been reviewed by the Contribution Review WG (CRWG); if for a production version, SWG must approve; only used for recently introduced functionality
Model - Enhancement	2 maintainers; must have been approved by a WG or the CRWG	2 maintainers; must be on roadmap or approved by SWG; must have been approved by a WG or the CRWG; must go into a dev version; at least one maintainer must be from a separate organization
Technical – e.g., dependency update, change to mapping, reference data, documentation, changes to samples….	At least one; additional review up to the maintainer's discretion – e.g. might need to consult the Technology Architecture WG (TAWG)	Must be approved by the TAWG; must go into a dev version

CDM – Release Guidelines Task Force

CDM
COMMON DOMAIN MODEL

Summary of Release Approval Requirements

Type of Release	Approval requirement	Notes
Development release	1 maintainer	Scheduling is up to the maintainer
Major release (6.0.0)	2 maintainers with SWG signoff	Scheduling via SWG; Include analysis of the changes from last major release as part of the approval
Minor release (6.1.0)	2 maintainers	Scheduling is up to the maintainers, but aim to keep to around monthly and no more than fortnightly
Patch release (6.1.1)	1 maintainer	Scheduling is up to the maintainer