# Recommendations from the Technology Planning Group for the potential merger of DSpace and DSpace-CRIS

Technical Strategy Report (Oct 28, 2025)

This Report was authored by the Technology Planning Group for the DSpace & DSpace-CRIS Merger. The following individuals participated in this group and the creation of this document:

Andrea Bollini (4Science), Ben Bosman (Atmire), Giuseppe Digilio (4Science), Lieven Droogmans (Atmire), Damian Józefowski (PCG Academia), Katharina Kaiser (TU Wien), Emmanuel Pastor (Université de Neuchâtel), Philipp Rumpf (University of Bamberg), Kim Shepherd (The Library Code GmbH), Mark Wood (Indiana University), Nicholas Woodward (Texas Digital Library), Tim Donohue (DSpace Tech Lead), Holger Lenz (DSpace Program Manager). Additional contributions provided by Pascal-Nicolas Becker (The Library Code GmbH).

**Table of Contents**

# Preamble

This document represents the Technology Planning Group’s recommendations regarding a potential merger of DSpace-CRIS and DSpace.

Based on the discussions within this group, we have determined that the **Training Program** for Committers / core developers, as outlined in the group’s charge, may be achieved during the merger via the “series of Pull Requests” approach that we outline below (see “Codebase Merger Strategy” section). This approach provides an opportunity for Committers and core developers to analyze and learn the new code/features in pieces. Additionally, 4Science has offered to provide a brief overview of each merger-related PR in our regularly scheduled Developer Meetings, with available smaller group meetings with 4Science staff as necessary. If Governance determines a more public training program is necessary (especially for post-merger), then we recommend a separate working group be established to develop a more detailed community training program.

# 1. Executive Summary

The Technology Planning Group has, between February and October 2025, examined the potential merger of DSpace-CRIS into DSpace from a technical perspective.

**We recommend proceeding with the merger of these platforms, and feel confident that a successful merger may be achieved using existing volunteer resources.**

**We propose performing the merger via a series of GitHub Pull Requests (PRs) over the span of two major DSpace releases** (either both 10.0 and 11.0, or both 11.0 and 12.0). Each release will include a subset of DSpace-CRIS features, based on the available development time and resources.

We highly recommend beginning merger activities with 10.0 in order to retain our current momentum and resolve uncertainties about ongoing contributions within our developer community. It also ensures both DSpace and DSpace-CRIS share a roadmap beginning with the 10.0 release. **Because the development cycle for DSpace 10.0 is rapidly decreasing, we highly recommend governance come to a quick decision on the merger.**

The resulting “merged” product will still be called DSpace. Institutions can decide which features to enable/disable, allowing the system to behave more like traditional DSpace, more like DSpace-CRIS or somewhere in between.

4Science estimates that the effort required for them to port all DSpace-CRIS features to DSpace is approximately 1,600 development hours. To better understand the size of this effort, this estimate is larger than the annual contribution expectations of the Platinum providers in the DSpace RSP Program (1,000 hours or $50K equivalent). This estimate also does not include the community effort required to review / approve the created Pull Requests, nor any additional development effort required for the processing of community feedback or unifying the codebases (especially regarding the different relationship models).

Spreading this effort over two major releases makes it possible to rely on our current volunteer network of community developers and Registered Service Providers (RSPs). However, we will need to establish new groups in order to keep merger activities on track. These include establishing a “Merger Management Team” to create/manage the merger roadmap and a “pool of volunteer code reviewers” to help ensure code reviews can be completed based on that roadmap. We also recommend finding ways to incentivize all RSPs to contribute to the merger activities. The more we can take advantage of existing resources, the quicker the merger will proceed.

By spreading these changes over two releases, we will also be able to provide an easier upgrade / migration process for both DSpace and DSpace-CRIS users into this single (merged) DSpace system. Upgrading / migrating to each of these releases is anticipated to be similar in complexity to the upgrade from DSpace 8 to 9.
Because this merger will involve large amounts of new code and close collaboration within our developer community, we recognize technical or strategic disagreements could occur which cannot be easily resolved by the Merger Management Team. In that scenario, we recommend using existing community groups and procedures to resolve any conflicts. Namely, Steering and the Committers should be tasked with jointly resolving any conflicts from a strategic and technical perspective, respectively.

## 2. Recommendations

Achieving a successful merger will require quick decision making and establishing new groups and policies specific to this merger process. To that end, we’ve provided additional detailed recommendations to Governance:

- **Define a start date for the Merger Roadmap as soon as possible.** With every week that goes by, the 10.0 development cycle goes shorter. We feel Governance must make a merger decision quickly and set a “start date” in order to make the most of the remaining 10.0 development cycle. We recommend beginning merger activities as soon as possible.
- **Establish a Merger Management Team -** This team will create, manage and maintain the *merger roadmap* and, if necessary, be empowered to make adjustments to merger schedules or release plans. A draft roadmap has been provided in this report which may be used as a starting point to their work.
    - Membership: 6-8 individuals. DSpace Program Manager and DSpace Tech Lead are ex-officio.
    - Must be willing to meet at least once every two weeks, or as needed for merger activities.
    - This team will report directly to Steering and provide regular updates to Steering related to ongoing merger activities.

- **Help to build/promote a Pool of Code Reviewers** - The Merger Management Team must establish a list of community developers/technologists who are available to help with code review of merger-related Pull Requests. Members must share their expertise, so that we are able to balance review tasks among the available members. Members must also share upcoming availability to allow for prior scheduling of PR code reviews.
    - This pool will be managed by the Merger Management Team
    - Governance should help to promote this effort to community developers.
    - Ideally, this pool should include at least 15-20 developers, with a mix of Java and Angular knowledge. Developers need not be experts to join this pool, but should be comfortable with basic code reviews and testing of PRs.
    - If this pool is small, then the Merger Roadmap timelines would need to increase, extending the timeline of the merger.
- **Incentivize Registered Service Providers to be active in the merger process** - To ensure our Registered Service Providers (RSPs) are encouraged to contribute to the merger process and/or code review process, we recommend considering an increase to the “multiplier credit” in the RSP program *for all merger-related activities*.
- **If major conflicts or disagreements arise during the merger process, use existing DSpace community groups and procedures -** If technical or strategic disagreements arise during the merger process (which cannot be easily resolved by the Merger Management Team), we recommend using existing DSpace groups and procedures to find resolution. Namely, Steering and Committers should be tasked with jointly resolving any conflicts from a strategic and technical perspective, respectively, via their established voting procedures.

## Future Priorities / Considerations

Irrespective of the outcome of the decision around the potential merger of DSpace and DSpace-CRIS, the Technology Planning Group has identified the following high priority concerns and recommends that Governance help address these in a timely manner.  We feel these should be prioritized by Governance for future discussion, but are not immediately solvable as part of the merger itself.

- **Challenging Code Review Process:** The current code review processes of DSpace are highly reliant on a small number of individuals. We need to find ways to expand this group of reviewers and incentivize others in the community to help with code review and testing processes. Should the merger go forward, there may be an opportunity to see whether the “pool of code reviewers” concept can be retained as an ongoing strategy to alleviate these challenges.
- **Speed of future code development:** Possibly related to the challenging code review processes, DSpace needs to develop strategies to increase the speed of adopting new code into the system. If a merger occurs, the priority may increase because DSpace-CRIS already has more frequent minor releases than DSpace, which may influence community expectations of a merged product.

# 3. Product Vision

The merged product should result in a single DSpace platform, which combines the codebases of DSpace and DSpace-CRIS and brings both developer communities together, eliminating duplication of features and efforts. This platform will provide the features of both products in a way that institutions can decide which to enable/disable via configuration.

As much as possible, the latest features of both DSpace-CRIS and DSpace should be retained in the merged product. But, exceptions may be necessary where the systems have overlapping features.

## Challenges to the Product Vision

In early discussions of this vision, the Technology Planning Group felt there were a number of challenges we'd need to address to make the merger successful. The challenges we determined are listed below along with our mitigation strategies for each.

- **Scale of merger effort:** Can this merger be achieved in one release?
  - *Mitigation:* Perform the merger over two major releases. (See "Two-Stage Merger Strategy" section below.)
- **Amount of new code:** Will be we able to use our current code review & testing processes for all this new code? How will we learn the code well enough to be able to support it in future releases?
  - *Mitigation:* Perform the merger via a series of Pull Requests (PR), giving reviewers the ability to learn code in smaller pieces. When PRs are larger, 4Science will offer a virtual meeting to provide an overview of the PR. (See "Codebase Merger Best Practices" in the appendices.)
- **Complexity of upgrade/migration:** Will it be difficult for either DSpace or DSpace-CRIS users to upgrade/migrate to this merged product? If so, that may cause frustration after all the effort institutions had to put into migrating to DSpace 7.
  - *Mitigation:* By performing this merger over two major releases we can stage the upgrade process for both groups of users. After analysis of the feature differences, we also feel confident this upgrade will be more like an upgrade from DSpace 8 to 9. It won't be as large as the upgrade from DSpace 6 to 7. (See "Two-Stage Merger Section" section below, along with the Appendix titled "DSpace-CRIS Feature Differences".)
- **Ongoing community contributions:** We are seeing more community code contributions in DSpace 9. If the merger takes too long or pauses community contributions for too long, then those developers may get frustrated. If one entire release is the DSpace-CRIS merger, then how do others contribute to that?
  - *Mitigation:* We will not pause community contributions at all. Instead, they will be accepted in parallel to the merger activities. While the "pool of code reviewers" will concentrate their reviews on merger-related PRs, members of this pool will be encouraged to review community contributions as well (when time allows).

(See “Codebase Merger Best Practices” section in Appendencies below.) Additionally, beginning the merger process as soon as possible will reduce any uncertainty that could cause community developers to refrain from submitting their contributions.

## 4. Two-Stage Merger Strategy

We propose merging DSpace-CRIS into DSpace **via a series of GitHub Pull Requests (PRs) over the span of two major DSpace releases** (either 10.0 and 11.0, or 11.0 and 12.0). We recommend beginning with 10.0, provided that the merger decision can be made as soon as possible.

**Stage 1: First Major DSpace Release**

The first major DSpace release will provide a partial integration of both platforms, where DSpace and *many of the primary features* of DSpace-CRIS coexist. (See “Draft Merger Roadmap” below for a list of features from DSpace-CRIS that we’d hope to see in this release.)

In this release, the DSpace system will behave *by default* similar to past DSpace releases (e.g. version 9.x). However, institutions may choose to enable (via configuration) available DSpace-CRIS features.

Existing DSpace users will be able to continue to use this release in the same way as recent DSpace releases (e.g. version 9.x). But, they may optionally choose to enable DSpace-CRIS features if they desire the enhancements to entities that they provide.

Because this release will not include all DSpace-CRIS features, existing DSpace-CRIS users may choose to remain on DSpace-CRIS until all features relevant to their use cases have been added to DSpace. To support this scenario, 4Science will release a “minimal” DSpace-CRIS (based on this DSpace major release) providing maintenance / security updates to existing DSpace-CRIS users. However, new users of either platform will be recommended to use DSpace.

This release’s primary goal is to begin introducing major features of DSpace-CRIS into DSpace in a staged manner, to allow Committers and core developers to learn and review the code. It also provides an opportunity for some DSpace-CRIS users (especially new ones) to migrate into DSpace early. For a detailed list of features considered for stage 1, see “Draft Merger Roadmap” section below.

**Stage 2: Second Major DSpace Release**

The second major release will occur approximately one year after Stage 1. It will provide a unified DSpace system which achieves the use cases of both DSpace and DSpace-CRIS. All existing DSpace and DSpace-CRIS users will now be running a single DSpace platform.

Additionally, architectural differences (namely differences in the relationships models) between DSpace-CRIS and DSpace will be resolved in this release.

Existing DSpace users may continue to use DSpace similar to current releases (e.g. version 9.x), but will now have the opportunity to enable additional features previously only available in DSpace-CRIS. These features will be possible to enable individually, but many will depend on using configurable entities.

Existing DSpace-CRIS users will be able to seamlessly upgrade to DSpace in a manner similar to past DSpace-CRIS upgrades.  They will also have the opportunity to choose which features they wish to have enabled/disabled via configuration.

This release's primary goal is to establish a new DSpace platform which is flexible enough to allow institutions to decide which use cases / features they want enabled in their site. Therefore institutions will now have direct control over whether their site looks more like (current) DSpace, (current) DSpace-CRIS, or somewhere in between.

## Why merge over two releases instead of one?

There are two main reasons we are recommending staging this merger over two DSpace releases: (1) the number of feature differences is more achievable via two releases, and (2) the architecture difference in Entity relationship models will be more complex to unify.

During analysis of the DSpace-CRIS and DSpace differences, we found a total of 29 features that only exist in DSpace-CRIS, all of which would need to be merged into DSpace in some manner. To follow established code review procedures and allow  the DSpace Committers (and other core developers) to review the code behind each of these features, it makes the most sense to split these features over two major releases of DSpace. This also puts less strain on our volunteer resources, and ensures we can keep to our schedule of one major release per year.

Currently, DSpace and DSpace-CRIS have different Entity relationship models. This means that, while the basics of Configurable Entities are identical in both platforms, how those Entities are related to each other is *different*. This difference makes the merger process more complex because some Entity features of DSpace-CRIS and DSpace require different relationship models. Spreading this merger over two releases allows DSpace Committers to unify these two models and select or develop a relationship solution which works for both DSpace and DSpace-CRIS use cases.

Spreading this merger over two releases has an additional benefit to the institutions using both DSpace and DSpace-CRIS. By spreading these changes over two releases, we will be able to provide an easier upgrade / migration process for both DSpace and DSpace-CRIS users into this single (merged) DSpace system. Upgrading / migrating to each of these releases is anticipated to be similar in complexity to the upgrade from DSpace 8 to 9.

# 5. Potential Merger Timelines

Based on the estimated DSpace 10.0 and 11.0 release timelines, we've provided two potential merger timelines. *We recommend starting the merger process immediately with the 10.0 release. This is why the 10.0 timelines below are more detailed.*

**Assuming we wish to start with 10.0 Release**, then merger activities should begin in November.

- *DSpace-CRIS Pull Requests (PRs) will begin to be created immediately.*
    - *Based on the Draft Merger Roadmap (see below), approximately nine PRs will be created for 10.0.*
    - *The three largest PRs will be created in mid-November, late January 2026, and in late February 2026 in order to give reviewers more time for each.*
- Feature PR Creation Deadline: Friday, February 20, 2026
- PR Merger Deadline: Friday, March 27, 2026
- Testathon: April 6-17, 2026 *(or April 6-24, 2026 if we want a longer, 3-week testathon)*
- 10.0 Release: Tues, May 26, 2026 *(or Tues, June 2, 2026 if testathon is extended)*
- *(Note: Open Repositories 2026 will be the week of June 8, 2026)*
- *Merger activities would resume in July, 2026, where seven additional DSpace-CRIS feature PRs would be released (little by little) for inclusion in the 11.0 release.*
- Stage 2 would complete with 11.0, due in May 2027

**Assuming we start with 11.0 Release**, then merger activities could be ongoing in parallel to 10.0. But, 10.0 would need to concentrate on early porting of a few minor DSpace-CRIS features, or any community contributions which don't make the merger more difficult.

- *DSpace-CRIS PR Deadline: End of December 2026?*
- Feature PR Creation Deadline: Friday, February 19, 2027
- PR Merger Deadline: Friday, March 26, 2027
- Testathon: April 5-16, 2027
- 11.0 Release: Mon, May 24, 2027 (Stage 1)
- Stage 2 would complete with 12.0, due in May 2028

## Draft Merger Roadmap for 10.0

While it was not part of our official charge, the Technology Planning Group has drafted a *potential* Merger Roadmap that aligns with our merger strategies and **assumes that merger activities begin immediately**.

4Science estimates that the effort required for them to port all DSpace-CRIS features to DSpace is approximately 1,600 development hours. This estimate does not include the community effort required to review / approve the created Pull Requests, nor any additional development effort required for the processing of community feedback or unifying the codebases (especially regarding the different relationship models).

The draft roadmap encompasses September 2025 - September 2026. As of this writing, we are currently “on schedule” because several smaller DSpace-CRIS features have already been ported to DSpace and are under review. These include the “Audit Framework” and the “emails to fixed recipients” features. These ports are merger-independent and will proceed regardless of the merger decision.

The draft roadmap defines a series of 18 Pull Requests which encompass all 29 DSpace-CRIS and DSpace Differences . It assigns timelines for each of the Pull Requests, with the following high level release schedule:

- **DSpace 10.0 Release** (12-19 features from DSpace-CRIS)
    - *Should include* these twelve features from DSpace-CRIS (via five PRs): Basic Hierarchical Metadata, Granular Edit/Security Privileges, Control over when linked Entities are created, multilingual support for content, Custom URL for Researcher Pages, “Edit Item” page provides a submission-like editing display, Authors can download restricted bitstreams, Shared Submission Workspaces, Audit Framework *(already in progress)*, Bulk import from Excel file, emails to fixed recipients *(already in progress)*, submission forms and workflow linked to collections from User Interface.
    - *Time permitting, may also include* these seven features from DSpace-CRIS (via four PRs): Customize Entity/Item display via an Excel Spreadsheet, Popup details about related entities, “Download” button instead of file link, Metadata Extraction from PDF files, Edit homepage news / header / footer, Edit user agreement, Share links via social networks/email.
- **DSpace 11.0 Release** (10 features from DSpace-CRIS, plus any that miss 10.0)
    - *Will include* any of the above features which miss 10.0
    - *Will include* resolution of architectural differences (namely unification of the relationship models)
    - *Will also include* these additional ten features from DSpace-CRIS (via seven PRs): Enhanced Statistics, Create a sorted list of “Selected Publications”, “Export” item pages to Citation formats, Advanced Duplicate Detection, Correction Requests, “My Processes” menu for users, Metric Framework and integration with external metrics, Subscriptions to Items, Configuration options for Homepage, Set relations reciprocally

**WARNING: This draft roadmap is meant only as a representation of how the merger might proceed if approved.** However, it’s very likely this plan will require modifications by the Merger Management Team based on the final merger approval date. Therefore, the above 10.0 and 11.0 release details *may not be accurate*, and each of these releases may include a slightly different feature set to those listed above.

The detailed draft roadmap can be found in this spreadsheet: https://docs.google.com/spreadsheets/d/1fxOA8KA8hWQ4Nu0aApW5pnk9-MWONHu3MPUIgmGvoMo/edit?gid=0#gid=0

# Appendices

## Codebase Merger Best Practices

As detailed above, we propose to achieve this two-staged merger via a series of DSpace-CRIS GitHub Pull Requests (PRs), scheduled over these two major releases. This strategy will allow DSpace Committers and developers to more easily review and learn the new code PR by PR.

However, because of the scale of code differences, we must establish some “best practices” specific to these Pull Requests (PRs) for all developers to adhere to.

Below are the best practices that the Technology Planning Group recommends for all development activities that occur during this merger period.

*Best Practices for “Merger” Pull Requests submitted from DSpace-CRIS to DSpace:*

- PRs should be individually testable, as this makes it easier to verify individual features have been successfully ported to DSpace.
- PRs should not be overly large, as this will allow for an easier code review. All PRs should be less than 20,000 lines of code. Less than 10,000 lines of code is ideal.
    - Developers of large PRs (>10,000 lines) should offer reviewers a virtual meeting to provide an overview or demo of the PR. This will allow the reviewers to quickly get up to speed on the PR, understand the code design, and clarify any questions they may have.
    - We expect there to be no more than two large PRs (per DSpace GitHub repo) open at a single time.
    - Alternatively, where possible, larger PRs can be split into several smaller “sub-PRs” which make sense thematically *but do not need to build or pass tests individually*. This approach would allow the sub-PRs to be code reviewed by different reviewers (splitting up the work and allowing it to be done in parallel). Testing the sub-PRs may not always be possible, but can be done via a grouped larger PR (or a parent sub-PR).
- Reviewers **must** be assigned immediately after each PR is created. Per current policies, each PR should be assigned two reviewers. One reviewer is acceptable if that reviewer is a Committer.
    - This will require an available pool of code reviewers. We also must track the expertise of available reviewers, so that we can more effectively match PRs with appropriate reviewers.
    - Optionally, it may also be helpful to have an available pool of *testers*, i.e. individuals who are willing to help test the functionality of PRs, but may not be comfortable with a code review.

    - To encourage Registered Service Providers (RSPs) to do more code reviewers, we recommend the Governance consider increasing the "multiplier credit" for code reviews (and all merger activities) in the RSP program.
- Reviewers **must** submit a review as quickly as possible.
    - Assigned reviewers must provide a time estimate of when they expect to complete their code review as a PR comment.
    - If an assigned reviewer realizes they cannot meet their original estimate, they should add a follow-up comment to the PR to provide a new estimate.
    - If an assigned reviewer can no longer complete the review, they can ask to be unassigned at any time. At that point, the review will be reassigned.
    - If a review is late or their PR seems "stalled", the PR developer should notify Tech Lead (Tim) or "Merger Management Team". At that point, the PR may be reassigned to a different reviewer as necessary.
- Reviewers may request code changes. But, if the change request has a significant impact on the current PR (or future PRs), then that request may be moved to an issue ticket for future cleanup.
    - If the PR developer feels a request will have "significant impact", they should detail the reasons why in a PR comment. At that point, the reviewer can either accept that analysis, ask for clarification or ask for a second opinion from the Tech Lead (Tim) or "Merger Management Team". The final decision may be made via agreement between the developer & reviewer. Disagreements will be resolved by the "Merger Management Team".
- Reviewers **may not reject** the merger of any individual DSpace-CRIS PR. They can only request current or future code cleanup.
    - That said, if a reviewer feels that an issue *must be addressed* before the release, they can request it be marked as a **"release blocker"**. Issue tickets with a "blocker" status must be addressed prior to the next major release. The developer may choose when to address the "blocker" issue, but ideally it should be addressed in a follow-up PR as soon as reasonably possible.
    - If multiple Committers from different organizations object strongly to an individual PR, they can still **veto** the PR. A veto will require the PR to be immediately updated or withdrawn.
- PR Developers **must** be quickly responsive if a PR encounters "merge conflicts" or bugs are found in the PR itself. Slow responses from developers may impact the review process.
- Ideally, all DSpace-CRIS PRs should be merged within 2-3 weeks (after the PR has been opened).
- If development disagreements arise, these will first be brought to the "Merger Management Team". If a technical decision needs to be made, then they should bring this disagreement to the DSpace Committers (or a subset thereof).

*Best Practices for all other Community Pull Requests (to avoid impacting the merger):*

- No PRs may modify the current Entity models for DSpace or DSpace-CRIS. Those models must remain stable during the merger.

- No PRs should introduce new features which only work for one entity model or the other. New features must not depend on a specific entity model (either DSpace or DSpace-CRIS). New features must be fully compatible with both DSpace and DSpace-CRIS.
- We encourage **early PR creation** from everyone in the community. Because of the merger process, we may not have enough time to review PRs that come in later in our release process. The earlier you submit your contribution the better.
- Developers creating PRs should consider trading PR reviews with another community developer. Committers may not have as much time for reviews as they will be busy with the merger process.
- If development disagreements arise, these will first be brought to the “Merger Management Team”.  If a technical decision needs to be made, then they should bring this disagreement to the DSpace Committers (or a subset thereof).

## DSpace-CRIS Feature Differences

During discussions of this Technology Planning Group, we documented a list of the architectural and feature differences between DSpace-CRIS and DSpace.  Those differences have been posted publicly on the wiki at:
https://wiki.lyrasis.org/display/DSPACE/DSpace-CRIS+and+DSpace+Differences

## Potential Merger Scenarios Considered

During discussions of this Technology Planning Group, the following merger scenarios were discussed before ultimately deciding on the merger scenario described above. This appendix is only included to provide additional context to our decision making process.

1. Adopt DSpace-CRIS as DSpace: No porting of DSpace-CRIS into DSpace, but adoption of DSpace-CRIS as the official DSpace
    a. Requirements: Review of DSpace-CRIS code different from the DSpace codebase
        i. Some more information / visibility about DSpace-CRIS QA/review processes to date would help with this requirement (also in scenario #3)
2. Port DSpace-CRIS into DSpace: Submit DSpace-CRIS code back to DSpace via one or more Pull Requests.
3. Adoption of DSpace-CRIS as DSpace, but trim DSpace-CRIS by removing code to simplify the review process.
4. Incremental “CRISification” of DSpace to make scenarios like “Port DSpace-CRIS into DSpace” and also DSpace-CRIS releases – since they involve merging DSpace code in – easier.
    a. E.g. rewrite relationships to be compatible with DSpace-CRIS, even if it is not porting code directly to achieve this
    b. This may help if it is decided to try and spread a merger over multiple releases

   c. As a precedent, recall the services “prep” work done in DSpace 5 prior to v6, even though most implementations didn’t arrive until v6
   d. This may read a lot like #2, but is more selective towards large architectural differences, which could then enable more code to become compatible and therefore able to go through normal PR, or fast-tracked merge processes
   e. This can also deal with the complexity of migration, by ensuring for each part, the consequences of CRIS users and DSpace users can be taken into account
   f. This can also deal with the Speed of future code development. Instead of merging the changes as-is, the focus can be put more on using the Service work and allowing both the current functionality and future functionality to be added or replaced more easily, even without requiring a new release
   g. Features and changes can also be classified in core vs non-core

5. Controlled Convergence of DSpace and DSpace-CRIS
   a. Gradual Merging Process: Accept a slower merging process to minimize disruption while ensuring steady progress toward convergence.
   b. Parallel Maintenance: Both codebases continue to be maintained by fixing critical bugs, but with limited evolution and no new feature development that are not in the convergence roadmap.
   c. Convergence Roadmap:
      i. DSpace and DSpace-CRIS teams jointly agree on features for the final merged version
      ii. One codebase is designated as the Primary codebase for the final solution
      iii. All new development is evaluated against compatibility with this chosen codebase
      iv. Teams avoid implementing completely new features unless necessary
   d. Transition Strategy:
      i. The primary codebase evolves according to the roadmap
      ii. The non-primary codebase should minimize significant changes but If really needed it can implements features that are in convergence roadmap
      iii. New users are directed toward the designated primary codebase solution
      iv. Resources gradually shift toward enhancing the chosen primary codebase

6. Merge DSpace-CRIS into DSpace over two major releases: an “intermediate” release and a “unified” release. *(This was the most discussed scenario and it eventually morphed into the merger strategy recommended above.)*
   a. First major release would be a “side-by-side” release, where the entirety of DSpace-CRIS would sit in the same codebase as DSpace.
      i. Would be released as a single DSpace release. Some features/concepts (especially relationships model) may be duplicated in the codebase, and would need to be unified in the next release.

    ii. Sites could then choose which system to use (ideally via a set of configurations), but would not have the full ability to pick and choose individual features.

b. The second major release would be a full “unification” of both platforms into a single platform.  This release would also involve unifying the relationship model, and selecting or developing a relationship model which met the use cases of both DSpace-CRIS and DSpace.

    i. Once unified, sites can choose exactly which features they want enabled in their DSpace. The system could act more like DSpace-CRIS, like DSpace or somewhere in between.