**3GPP Call on modernization of 3GPP tools #2 6GSM-250047**

**Electronic, 3rd September 2025 *was 0043***

**Source: Nokia, Huawei, Apple, Samsung, ZTE, Ericsson, NTT DOCOMO INC., Xiaomi**

**Title: Pseudo-CR on Shortcomings, Pain Points and Potential Improvements to 3GPP Specification Format and Working Methods**

**Document for: Discussion, Endorsement**

**Agenda item: 5.2**

**Spec: 3GPP TR21.802**

**Version: 20.0.0**

**Work Item: FS\_6GSpecs Release 20**

**Comments**

In SA#108, a cross-TSG SID was approved in [SP-250802](https://www.3gpp.org/ftp/tsg_sa/TSG_SA/TSGS_108_Prague_2025-06/Docs/SP-250802.zip). The purpose is to have a study item across RAN/CT/SA in 3GPP in order to understand what possibilities we have to improve the current ways of handling the 3GPP specifications going forward towards 6G.

This CR adds terminology and shortcomings, pain points, and potential improvements to 3GPP specification format and working methods, applicable to objectives 1a and 1b, described in [SP-250802](https://www.3gpp.org/ftp/tsg_sa/TSG_SA/TSGS_108_Prague_2025-06/Docs/SP-250802.zip). Throughout this document, we refer to different types of CRs which fall into two broad categories: CRs which are updated with revisions through collaborative processes and CRs which are aggregations of more than one CR. Whether a CR has been updated, aggregated during one meeting, or aggregated across meetings, a CR which is intended to be merged into the next release of the specification.

Comments provided on 6GSM-250043 were incorporated into this version.

|  |
| --- |
| First objective in [SP-250802](https://www.3gpp.org/ftp/tsg_sa/TSG_SA/TSGS_108_Prague_2025-06/Docs/SP-250802.zip):   1. Identify shortcomings and pain-points of existing specification formats and working methods, and potential benefits to be targeted, considering the needs and processes across the 3GPP working groups and other users of the 3GPP specifications. 2. Where possible, study possible approaches to improve the pain points by enhancing working methods while continuing the use of current specification tools, including identifying pros and cons; 3. Identify requirements for any improvements to working methods and/or specification handling, e.g. ability to use tools in all countries and regions, ability to work on CRs locally, operability with company IT policies, etc. |

Based on the email discussion, the final version of which can be found [here](https://www.3gpp.org/ftp/workshop/FS_6GSpecs/6GSM_Meeting_01/Inbox/Post-meeting%20discussion/5.2/6GSM-25xxxx_draft_email_discussion_v009_Nokia.docx), this pCR was generated, taking into account the comments from the discussion. The solutions were split from the pain-points such that they could be reused and so that they would be easier to discuss. Pain points 16, 22, and 26 were omitted based on the comments. Many pain-points were merged for conciseness.

**Proposed Changes**

\* \* \* First Change \* \* \* \*

# 3 Definitions of terms, symbols and abbreviations

## 3.1 Terms

For the purposes of the present document, the terms given in TR 21.905 [1] and the following apply. A term defined in the present document takes precedence over the definition of the same term, if any, in TR 21.905 [1].

**Consumer:** A person who reads the specification such as an Original Equipment Manufacturer (OEM), a mobile network operator (MNO), an implementor, government regulatory body, researcher, automated text processing tool, or the general public.

**Contributor:** A person who contributes to the specification, e.g., a delegate, by submitting change requests (CRs) or TDocs.

**Editor:** A person who merges changes into the specification, e.g., a rapporteur or MCC.

## 3.2 Symbols

For the purposes of the present document, the following symbols apply:

<symbol> <Explanation>

## 3.3 Abbreviations

For the purposes of the present document, the abbreviations given in TR 21.905 [1] and the following apply. An abbreviation defined in the present document takes precedence over the definition of the same abbreviation, if any, in TR 21.905 [1].

<ABBREVIATION> <Expansion>

\* \* \* Next Change \* \* \* \*

## 4.2 Shortcomings, pain-points and potential benefits

Editor’s note: corresponds to objectives 1a/b

Table 4.2-1: Shortcomings, pain-points and potential benefits of current tools

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| # | Shortcoming / pain-point / potential benefit | Possible improvement approaches with current tools | Summary of feasibility of addressing the shortcoming / pain-point / potential benefit with current tools | Applicable WGs and users of the specification |
|  |  |  |  |  |
|  |  |  |
| 1 | **Delays in specification availability** Specifications are not available until after the TSG plenaries, sometimes very close to submission deadlines for the next WG meeting. Thus, writing new CRs is delayed or CRs are written based on an old version. There is also limited time for review of the merged spec. | Solution 1 – Distribute CR merging workload to delegates  Solution 2 – Scripting to automatically merge CRs | **Not feasible** – Solution 1 (infeasible) redistributes the load from MCC to delegates who would take on some of the work. Solution 2 (maybe feasible) is being developed for ETSI, but its feasibility is unknown. | **WGs** All groups  **Users** Consumer, Contributor, Editor |
| 2 | **CR cover page errors** Errors can be present in the cover page details such as specification number and version, date, release, CR revision, affected clauses, and changes are not reflected in database | Solution 3 – [New tool] CR conformance checking  Solution 4 – [New tool] CR auto-generation  Solution 5 – CR cover auto-generation | **Maybe feasible** – Solution 5 is available and can be useful if the cover page is generated and attached to the CR just before submission. Solution 3 and Solution 4 would solve the problem, but the tools do not exist. | **WGs** All Groups  **Users** Contributor, Editor |
| 3 | **CR content errors**   * Incorrect specification base text, e.g., from the incorrect version * Errors introduced by copying and pasting content such as additional sections * Errors introduced by copying a CR for a later release to mirror CRs for prior releases * Unused references, e.g., references that are used but not defined, or defined which are not used, and references to non-existent clauses * Incorrect use of styles, including manually applying text formatting to mimic a style * Use of different settings in the docx editor such as language can lead to inconsistent formatting and styles which diverge from the official template * Omission of trademarks, commercial references, etc. | Solution 3 – [New tool] CR conformance checking  Solution 4 – [New tool] CR auto-generation  Solution 6 – Restrict editing  Solution 7 – Training  Solution 8 – Light version of Microsoft Word  Solution 9 – EditHelp Consulting from ETSI  Solution 10 – Exhaustive analysis of docx files against 21.801 | **Not feasible -** Solutions 3, 4, and 8 require new tools, which need to be written. Today they are infeasible since they do not exist.  **Not feasible** – Solutions 7, 9, and 10 resolve parts of the issues. Solution 7, 9, and 10 are already done today and we still have the problems.  **Maybe feasible** – Solution 6 could be feasible to resolve the style errors as long as everyone uses Microsoft Word.  NOTE: It has been noted that some of the errors are due to not following the process. It should be considered which of these errors could be prevented by following rules and whether it is feasible to enforce the following of such rules. | **WGs** All Groups  **Users** Contributor, Editor |
| 4 | **CR Merging**   * Multiple changes to the same section sometimes remove specification text due to human errors. * It is not possible to easily check the effect of the implementation of more than one CR or pCR simultaneously to check for side-effects * It isn’t trivial to identify conflicting CRs such that the relevant experts, e.g., CR editors or WI rapporteurs, could discuss the appropriate resolution. * Implementation of a CR is a manual process | Solution 2 – Scripting to automatically merge CRs | **Maybe feasible** – if the tool would be available for all specification contributors and editors, then it would be possible to check the effect of CRs. | **WGs** All groups  **Users**  Contributor, Editor |
| 5 | **CR Tracking**   * It is difficult to associate a spec change with a CR, e.g., to determine where a change came from between two non-contiguous versions of a specification. For example, something changed between 18.1 and 18.7, but the exact version is unknown. * CRs which are part of mega CRs or aggregate CRs are not visible individually in the draft spec. * Changes to CRs to resolve merging conflicts, including those present in tables, ASN.1 definitions, and other text conflicts are not identifiable. * Checking implementation of CRs is a manual process * Changes in a new version of a specification are hard to track for implementors. It is particularly difficult to track an individual feature. * Missing background information on how certain test steps or approaches (pertaining to test parameters or implementation) were adopted for in conf test procedure |  |  | **WGs** All groups  **Users**  Consumer, Contributor, Editor |
| 6 | **Data Structure Styling, Syntax and Representation**   * APIs and data structures, e.g., ASN.1, can contain formatting errors such as indentation for readability and colorization of keywords. * Code-like and data model or table representations coexist and can be inconsistent, e.g., ASN.1, OpenAPI and XML in tabular form can be inconsistent and don’t match the code representations. * OpenAPI: Different WGs (e.g. CT4 and SA5) often use the same datatypes but they are defined slightly differently. * In some groups, when modifications are made during a meeting, syntax errors can be introduced due to lack of verification. * In some groups, the contributions to the meeting can include syntax errors. * It isn’t easy to extract code | Solution 11 – Externalization of APIs and data structures  Solution 12 – Single representation of APIs and data structures  Solution 13 – Automatic syntax checking | **Feasible –** Solution 11 is already used by some WGs to store certain data structures and APIs separately from the specification, e.g., RAN4, CT4, and SA5. Solution 12 is already used in RAN2 which moved away from tabular ASN.1.  **Maybe Feasible** – Solution 11 could be adopted. Groups such as RAN2 and RAN3 include field definitions and descriptions of conditionals in field tables. A solution for storing the field descriptions, which contain normative text, needs to be devised. Additionally, many find it convenient to have the procedures and protocol definitions in the same document.  Solution 13 would require new scripting to extract text from the specification and run the syntax checker. | **WGs** All groups  **Users**  Consumer, Contributor, Editor |
| 7 | **Cross-specification reference and navigation**   * References to other specification documents are made using numbered references in square brackets and are sometimes accompanied by a section or clause number. * Navigating across a work item over multiple specifications and working groups can be time consuming. For example, a RAN2 procedure could be triggered by RAN3 procedure, triggered by an SA2 procedure. * Conformance Test specifications use heavy cross-referencing between the test spec (e.g. 38.521-1) and associated details in test point analysis (TR 38.905), MU/TT (TR 38.903), UE/UE connection diagrams (TS 38.508-1) | Solution 14 - Include the specification number and clause number in references  Solution 15 – Use hyperlinks in references  Solution 16 - Script to create, validate, and automate verification of cross-references between specifications | **Feasible** – Solution 14 at least solves the problem of accurately referencing another specification or the same specification.  **Partly feasible** – Solution 15 – while feasible, links would be to zip files hosted on the 3GPP FTP server. These would have to be unzipped, which is inconvenient, and the reference could not be to a specific clause.  **Infeasible** – Solution 16 doesn’t exist. | **WGs** All groups  **Users**  Consumer, Contributor, Editor |
| 8 | **Figures can become impossible to edit**   * Occasionally, CRs convert Visio figures to a format which cannot be edited. Sometimes Visio figures are drawn in an older format. * MSC-Generator diagrams can become corrupt | Solution 17 – Store figure source files | **Maybe feasible** – Solution 17 is feasible in the current tools because figure source files could be attached in the zip files. However, specifications can contain dozens to hundreds of figures, which could lead to error in omission of files and could be unscalable. Additionally, each attached file would have to be individually opened. | **WGs** All groups  **Users**  Contributor, Editor |
| 9 | **Quality and size of figures**   * Some specifications and TRs include many images or a large body of text which can contribute significantly to the size of a docx file, leading to high upload and download times. * Figures are resized in dimension or compressed, thereby being distorted. * Microsoft has disallowed usage of equation editor 3.0 due to security issues with equation editor. When opening equations that were originally written using equation editor are stored as rasterized pixel images. Such images have image distortion issues when resizing the scale of the view. | Solution 18 – Store images as vector graphics | **Feasible** – Solution 18 is feasible for non-equation graphics. | **WGs** All groups  **Users**  Contributor, Editor |
| 10 | **Inconsistent use of tools and formats for figures**   * e.g., Word, Visio, MSC-Generator, PlantUML, MS paint, PowerPoint, Equation editor, Latex, etc. * Use of non-cross-platform formats (Visio, objects embedded in documents using Windows OLE are not available on macOS). | Solution 19 - Support a limited number of supported formats which work across WGs and platforms | **Feasible** – Solution 19 would consider tools supported by the overall specification format, so they would be compatible.  NOTE: This could also solve pain-point 8 depending on what is chosen. | **WGs** All groups  **Users**  Contributor, Editor |
| 11 | **Large tables are not handled well** - MS word does not handle large tables well and large documents well - can cause MS Word to crash or operate slowly. | Solution 11 – Externalization of APIs and data structures  Solution 25 – Split large specifications into smaller parts | **Maybe feasible** – Solution 11, depending on the nature of the table’s content, it could be externalized, e.g., as in RAN4  **Feasible** – Solution 25 is used today. | **WGs** All groups  **Users**  Consumer, Contributor, Editor |
| 12 | **Collaboration on CRs**   * Limitations of change tracking include: specific changes to figures are impossible to see; change marks not shown for columns deleted from tables; change marks are not shown when merging or unmerging cells; and change marks are shown in different colors, which poses an accessibility issue. * Collaborative editing relies on downloading a document, providing input, and uploading it. There is a race condition when multiple delegates are working at the same time. * Commenting bubbles do not scale well and having too many which affect the same clause makes it impossible to read. * It is difficult to update documents when used to collect input since manual locking does not work and progress becomes slow and error prone | Solution 20 - Provide comments on CRs in a separate file instead of using bubble comments  Solution 21 – Use NWM to collect comments  Solution 22 - Extract text under review and use Git to manage reviews  Solution 23 - Use FTP to download latest inputs and upload new inputs  Solution 24 - Split the CR for review into multiple files | **Infeasible** – Solution 20 and 21 could become unscalable when commenting on a large CR with a large number of comments and responses. Solution 22 would cause the loss of content most likely and would result in an inaccurate review.  Feasible – Solution 23 is used today. Solution 24 could also work, but it has problems similar to Solution 23 and could increase confusion. | **WGs** All groups  **Users**  Contributor, Editor |
| 13 | **Specification opening and navigation delay**   * Opening a TR or TS of 100s or 1000s of pages can take many minutes or even be impossible due to crashing. * Searching with keywords can be slow. * The specifications are stored as ZIP files, which add another step to opening the specification. * There seems to be some issue in the .docx specification template itself, regarding the header that appears on top of each page which checking for every single page how the header should be filled even if there is only one outcome. | Solution 6 – Restrict editing  Solution 7 - Training  Solution 8 – Light version of Microsoft Word  Solution 11 – Externalization of APIs and data structures  Solution 25 – split large specifications into smaller parts  Solution 26 - Open specification and change to draft mode  Solution 27 - Produce 3GPP PDF version of the specification after each plenary  Solution 28 - Make all specs available in HTML | **Maybe feasible –** Solution 6, Solution 7, Solution 8 could help solve the problem, but so far these have not resulted in faster loading specs. Solution 25 works but has downsides like lack of navigability. Solution 26 works once the document is open, but crashing can occur prior to being able to switch to draft mode. Solution 28 may be infeasible because the conversion of Word to HTML could be lossy.  Solution 11 is feasible but has more impact.  **Feasible –** Solution 27 is already done by ETSI and other SDOs, but 3GPP could possibly release a version more quickly.  NOTE: No solution has been provided for how to deal with ZIP files. | **WGs** SA1, SA5, CT3, and all groups which refer to SA1 requirements  **Users** Contributor, Editor |
| 14 | **Numbering of PRs and CPRs**   * Potential Requirements (PR), Consolidated Potential Requirements (CPR), requirements are numbered manually and inconsistently within TR/TS, making it error prone for tracking or later reference. * In some groups, requirements are not numbered which makes reference to requirements very difficult - it must be done by copying the text of the requirement. This can become misaligned, if the text is corrected in the specification where it is a provision. | Solution 29 – Mandate the numbering of requirements (PR and CPR) | **Feasible** – In Solution 29, numbering just needs to be enforced. |  |
| 15 | **Automatic processing of specifications**  Access from automated text processing tools, e.g., Automata, to CRs and TSs/TRs is very cumbersome, requiring a lot of preprocessing and manual intervention |  | The docx format is not easily processed. The file format is a mix of text and binary (to store images and objects). Conversions, e.g., docx to markdown, do not produce perfect representations of the original docx. | **WGs** All Groups  **Users** Contributor, Editor |

Table 4.2-2: Possible improvement approaches with current tools

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| # | Possible improvement approaches with current tools | Pros of possible improvement approaches | Cons of possible improvement approaches | Implementation Feasibility Analysis |
| 1 | **Increase workforce for CR merging** – In WGs where all CR merging is done by a single MCC officer today, there is the possibility to:   * Increase the MCC staff for CR merging. * Offload this task to each spec rapporteur. 3GPP could provide regular training courses for delegates that are candidates to become TS rapporteurs. | Drastic decrease in the workload for MCC officers, reduces the need to look for automated tools for merging CRs into specs. | Increase in workload for delegates. Need to ensure that all TS rapporteurs complete the task timely and respect all 3GPP drafting rules with proper and regular training. |  |
| 2 | **Scripting to automatically merge CRs** – ETSI is having a tool developed to automatically merge CRs into the specification. | Reduces human error and decreases the time to produce a merged specification. | It is unknown if such a script could be released to delegates and if it would comply with company security policies. For example, if macros are used, many would not be able to run them. |  |
| 3 | [New tool] **CR conformance checking** – write a new tool to check and report the location and nature of errors in a CR. These errors could include style errors, non-contiguous clause numbering, non-conforming figure and table numbering, and mismatches between the CR database and the CR cover page. | By using the tool pre-submission, delegates would not experience delays in submission and less time would be spent during meetings discussing styles and formatting. | It is difficult to write and maintain consistent tools based on docx due to high variability in docx file structures.  Ensuring that styles didn’t change between versions and ensuring that the correct approved styles are used could be challenging and encounter corner cases. |  |
| 4 | [New tool] **CR auto-generation** - Write a tool, e.g., an application-native script, Python script or other programming language-based tool to automatically create a CR from a modified version of a specification, including the following:   * Auto-filled cover page including the specification number, specification version, and affected clauses * Automatic inclusion of affected clauses in the body of the CR, including change marks * Additionally, the macro would need to be available for local use such that delegates could check their CRs prior to submission. | * Many error-prone fields of the CR cover page would be guaranteed to be correct. * Style errors would be prevented * The correct specification base text would be used | - Modifying additional clauses or removing affected clauses requires a regeneration of the CR  - Difficult to write and maintain consistent tools based on docx due to high variability in docx file structures  - Some companies do not allow the use of macros.  - This solution only works for the initial version of the CR. |  |
| 5 | CR cover auto-generation | The tool exists online at the time of reserving a TDoc for a CR. | The tool is only available for online use. |  |
| 6 | Restrict editing  A screenshot of a computer program  AI-generated content may be incorrect. | The feature is available in current tool used for writing CRs and the specification. | This feature is not available in all docx editors. |  |
| 7 | **Training** – (1) correct use of styles | Easy to implement through reminder by Chairs or MCC with detailed guidance. It is possible for MCC to organize training sessions and to provide tutorial documents. | Some tools other than used for processing docx files may not allow delegates full control of the styles. |  |
| 8 | **Light version of Microsoft Word** - light version would include only features used in 3GPP without any additional feature, e.g. removing the possibility of adding new styles, and many other functions. | Removing most of Microsoft Word features should help reduce file sizes, ensure consistent use of styles and formatting, and may make automatic processing of compressed docs files easier. | Microsoft is not likely to create a custom version of Word for 3GPP. |  |
| 9 | **EditHelp Consulting from ETSI** - This is done before entering change control. After change control secretary review is possible but does not scale well. | Current rules require this  [In principle all authors must use official templates and settings. In practice, no one enforces this] | Forcing adherence to rules leads to slower progress. Those who did so were called CR police - respected, not loved.  This solution relies on accountability by contributors for repeated errors, but this is disrespectful in a professional setting. So this solution is not feasible.  [In principle all authors must use official templates and settings. In practice, no one enforces this] |  |
| 10 | **Exhaustive analysis of docx files against 21.801** | Issues would be discovered that are currently hidden. | This would have a very large impact on ways of working since authors would need to strictly use MS Word according to rules. |  |
| 11 | **Externalization of APIs and data structures** - APIs and data structures could be stored externally to the specification. | Standard text editors would take on the role of enhancing readability of APIs and data structures by providing colorization, indentation, and syntax highlighting. | The specifications would be more difficult to search.  Field descriptions and procedures related to the APIs and data structures would be stored separately, requiring more careful checking.  Requires code to be separate from CRs, essentially. This reduces CR specification cohesiveness. |  |
| 12 | **Single representation of APIs and data structures** - APIs and data structures could be limited to a single representation. | The need for specifying which version of an API or data structure is authoritative would be eliminated. | Some WGs include additional information, e.g., a tabular form of APIs and data structures and would need to devise a new way of capturing the additional information. |  |
| 13 | [New tool] **Automatic syntax checking** - Syntax checking  Reference checking, e.g., checking that a datatype of a parameter exists as an intrinsic to the language or as defined elsewhere in the API or data structure specifications. | The volume of CRs with syntax and consistency errors would be reduced. | Many errors cannot be identified without checking across all content associated with the entire target specification, or across multiple specifications. |  |
| 14 | **Include the specification number and clause number in references** – When available, include enough information to find the information referenced. |  |  |  |
| 15 | **Use hyperlinks in references** – manual process | Reader could immediately navigate to a reference without having to search each document. | The specifications are stored as zip files on the 3GPP server and in the 3GPP portal. If links were provided in the specifications, they would be to zip files, which would have to be downloaded, extracted and opened. |  |
| 16 | [New tool] **Script to create, validate, and automate verification of cross-references between specifications** | Completeness and accuracy of references can be improved (if a solution is feasible). | If a solution is feasible, it will require extensive parsing of many specifications which may take time and may turn out to be prone to errors. Validation will be complex. |  |
| 17 | **Store figure source files** - For editable figures, store at least the source file, named in accordance with the figure number, alongside the specification. | - Standalone editors could be used to modify figures  - The figure source would never be lost.  - Because captions and headings are never reused, file naming consistency should be feasible. | - Any time a figure is edited, an extra step needs to be taken to ensure that the figure source file is updated.  - Errors could occur in naming of the source files.  - This disaggregation of content removes one benefit of the use of MS Word, namely that a single file contains all content in a simple way. |  |
| 18 | **Store images as vector graphics** | Scalable Vector Graphics (SVG) are scalable without distortion and are smaller in file size. |  |  |
| 19 | **Support a limited number of supported formats which work across WGs and platforms** | Limited changes to the current way of working. |  |  |
| 20 | **Provide comments on CRs in a separate file instead of using bubble comments** | Using a separate WORD document for collecting comments is a widely used technique in WGs, and allows responding to comments efficiently | When using a draft FTP folder, comments from multiple delegates may collide in time and separate branches of the same file are created, which are then cumbersome to merge. Delegates have to carefully provide the reference to the clause and line of the spec they are commenting on. |  |
| 21 | **Use NWM to collect comments** | Using NWM for collecting comments is a technique used by some TSGs/WGs. | Using NWM avoids collision of comments, but NWM is not convenient for commenting on equations or figures. Delegates have to carefully provide the reference to the clause and line of the spec they are commenting on. |  |
| 22 | [New tool] **Extract text under review and use Git to manage reviews** | This would help reviewing text-based large parts of specs where typically a large number of comments are received during the review of a draft TS. Delegates would be able to see spec text and provide a comment directly over that spec text without using another file. | This requires separating (during review time) some parts of the specs which are only text-based for separate commenting, since text-based file formats may not be suitable to parts of specs that contains other types of objects (equations, figures, etc). |  |
| 23 | **Use FTP to download latest inputs and upload new inputs** | It is based on existing tools. | Simultaneous editing and updating is not possible nor traceable. Coordination is very difficult. |  |
| 24 | **Split the CR for review into multiple files** – this can include splitting the ASN.1 from a spec like 38.331 for review. | It is based on existing tools and does help speed up work and reduces some contention. | Other problems are created, as changes to different parts of the spec can diverge. This approach does not scale to large files. |  |
| 25 | **Split large specifications into smaller parts** | Faster opening | Decreased locality of content, more difficult to maintain, read, etc. |  |
| 26 | **Open specification and change to draft mode** | - Loading and editing times could be reduced.  - Change marks are still visible in draft mode | - It is more difficult to check bubble comments in draft mode  - Some document editing tools have been known to crash prior to being able to enter draft mode.  - Need to ensure correct insertion of figures in WORD so that they are still visible in draft view. |  |
| 27 | **Produce 3GPP PDF version of the specification after each plenary** | - Quicker access to a version of the specification which isn’t impacted by the slowness of some WYSIWYG editors.  - Processing for converting all WORD docs into PDF is done only once for each spec version, | - Cannot be used for producing CRs.  Need to ensure that conversion does not lose any information.  Requires additional storage on FTP server. |  |
| 28 | **Make all specs available in HTML** | Faster opening, faster search.  Processing for converting all WORD docs into HTML is done only once for each spec version, presumably by MCC. | Cannot be used for producing CRs.  No navigation panel in html.  Need to ensure that conversion does not lose any information.  Requires additional storage on FTP server. |  |
| 29 | **Mandate the numbering of requirements (PR and CPR) -** In some WGs, requirement numbering is done already. Maintenance of numbering is done the normal way - through careful alignment of provisions whenever they are introduced or corrected. | It is possible to refer to requirements without copying the text of the requirement. This reduces the risk of misalignment between specifications. |  |  |
| 30 | **Include WI and meeting number with editor’s notes -** For example, Editor’s Note: [FS\_AIML-air\_core, RAN2#129] | **-** It would be easier to find the delegate(s) responsible for the editor’s note  **-** It would be easier to identify stale topics to resolve. | - Slightly more work and introduces meeting-related details into the specification, which isn’t ideal. An alternative would be to find the CR which introduced the Editor’s note. |  |

\* \* \* End of Changes \* \* \* \*