United Nations ECE/TRANS/WP.30/GE.1/2021/33/Rev.1

Economic and Social Council

Distr.: General
2 July 2021

Original: English

Economic Commission for Europe

Inland Transport Committee

Working Party on Customs Questions affecting Transport

Group of Experts on Conceptual and Technical Aspects of Computerization of the TIR Procedure

Third session
Geneva, 13–15 September 2021
Item 4 (d) of the provisional agenda
eTIR conceptual, functional and technical documentation version 4.3: eTIR technical specifications

Development and maintenance processes of the eTIR international system, interconnection projects and the eTIR data model

Revision

Note by the secretariat

I. Mandate

1. The Inland Transport Committee (ITC), at its eighty-second session (23–28 February 2020) approved (ECE/TRANS/294, para. 84[1]) the establishment of the Group of Experts on Conceptual and Technical Aspects of Computerization of the TIR Procedure (WP.30/GE.1) and endorsed its Terms of Reference (ToR)[2] (ECE/TRANS/WP30/2019/9 and ECE/TRANS/WP.30/2019/9/Corr.1), pending approval by the United Nations Economic Commission for Europe (ECE) Executive Committee (EXCOM). EXCOM during its remote informal meeting (20 May 2020) approved the establishment of WP.30/GE.1 until 2022, based on the ToR included in document

[1] Decision of the Inland Transport Committee para. 84 / ECE/TRANS/294 www.unece.org/fileadmin/DAM/trans/doc/2020/itc/ECE-TRANS-294e.pdf

[2] Terms of reference of the newly established Group approved by the Inland Transport Committee and the Executive Committee (EXCOM) of UNECE www.unece.org/fileadmin/DAM/trans/bcf/wp30/documents/2019/ECE-TRANS-WP30-2019-09e.pdf and corrigendum www.unece.org/fileadmin/DAM/trans/bcf/wp30/documents/2019/ECE-TRANS-WP30-2019-09c1e.pdf

ECE/TRANS/WP.30/2019/9 and Corr.1, as contained in document ECE/TRANS/294 (ECE/EX/2020/L.2, para. 5(b)).[3]

2. The ToR of the Group stipulate that the Group should focus its work on preparing a new version of the eTIR specifications, pending the formal establishment of the Technical Implementation Body (TIB). More specifically, the Group should (a) prepare a new version of the technical specifications of the eTIR procedure, and amendments thereto, ensuring their alignment with the functional specifications of the eTIR procedure; (b) prepare a new version of the functional specifications of the eTIR procedure, and amendments thereto, ensuring their alignment with the conceptual specifications of the eTIR procedure; (c) prepare amendments to the conceptual specifications of the eTIR procedure, upon requests by WP.30.

3. This document presents the development and maintenance processes of the eTIR international system. It also describes the interconnection projects and the eTIR data model. These aspects will be part of the eTIR technical specifications document.

II. The eTIR international system

A. Development processes

1. Introduction

4. This section describes the processes followed by the Information Technology (IT) experts of ECE (hereafter "the IT experts") to develop the eTIR international system so that contracting parties to the TIR Convention and the other eTIR stakeholders have a clear understanding of these aspects. Being transparent about these processes also provides an opportunity to all eTIR stakeholders to suggest proposals for improvement with the ultimate objective to have a more effective and efficient eTIR system in the long term.

2. General guidelines

5. The IT experts have taken the time to prepare, discuss and adopt their own internal guidelines related to all aspects of the development and maintenance of the eTIR international system. Proven best practices from the IT industry and experience acquired by the IT experts drive the formulation of these guidelines. Nevertheless, they are not set in stone and the IT experts will continuously strive to identify opportunities to improve them. This is especially important in an area of expertise such as Information and Communication Technology, that is evolving so quickly.

6. While preparing and improving guidelines, as well as in all decision-making processes, the IT experts are inspired and steered by the three guiding principles detailed at the beginning of this document.

7. When taking a technical decision on any aspect related to the eTIR international system, the IT experts follow the usual best practices in decision-making process, like the PrOACT model[4]. The time needed to explore and study emerging trends, approaches and possible products is invested. Possible options are then formulated, their respective advantages and disadvantages are listed and then a decision can be taken to select the best option. Decisions are documented, along with the rationale that led to this choice, to keep proper institutional memory.

8. Finally, the IT experts also recognize and take into account the Pareto principle[5] in their decision-making process to find the optimum in achieving most of the benefits in the

[3] Decision of EXCOM, ECE/EX/2020/L.2 / para. 5(b) www.unece.org/fileadmin/DAM/commission/EXCOM/Agenda/2020/Remote_informal_mtg_20_05_2020/Item_4_ECE_EX_2020_L.2_ITC_Sub_bodies_E.pdf

[4] See www.project-management-skills.com/decision-making-model.html

[5] See en.wikipedia.org/wiki/Pareto_principle

least amount of time possible. This principle is usually confirmed when it is applied in software engineering and it becomes even more pertinent in times of difficult economic situations to ensure funds are wisely spent.

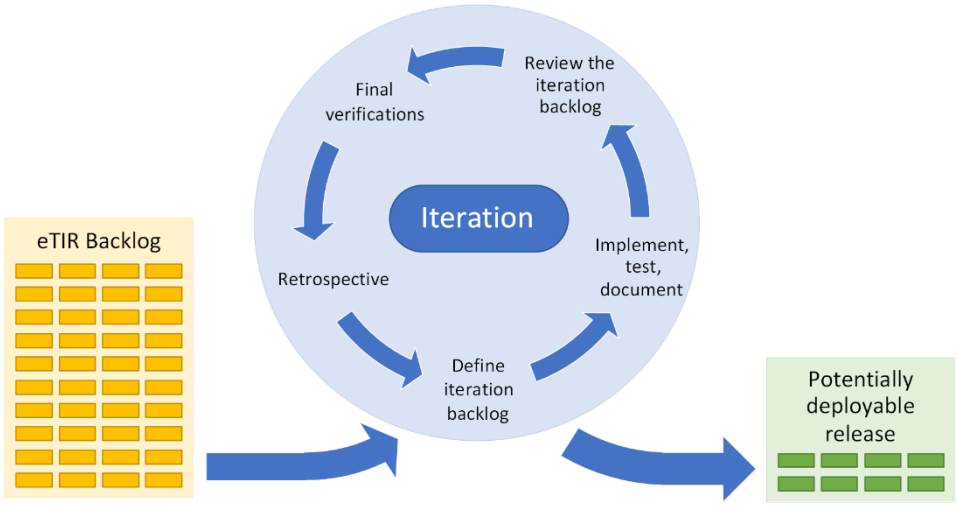
3. Development methodology

9. The development of a major information system such as the eTIR international system requires following an IT project methodology to be successful. In the short – although intense – history of IT, several paradigms and models have been proposed and extensively tested (e.g. Waterfall, V model, Prototyping, Incremental, Agile, etc.). In 2001, emerging from several new agile methodologies (e.g. eXtreme Programming and Scrum), a major breakthrough was achieved with the establishment of the Agile Manifesto[6] and its twelve principles. Since then, many IT projects were conducted using agile methodologies which offer the best chances of success for such complex endeavours.

10. ECE has chosen to follow an agile methodology close to Scrum and Kanban to develop the eTIR international system. This approach focuses on the following objectives: developing valuable and working software, being able to quickly respond to change, instil a high level of quality and above all, satisfying the beneficiaries.

11. All the work that needed to be done is broken down into tasks (hereafter referred to as "issues") and maintained in a list, called the eTIR backlog. Development is performed by iterations of several weeks. At the beginning of each iteration, the IT experts select from the eTIR backlog a set of issues to define the iteration backlog. During the iteration, the implementation, testing and documentation activities are performed on these selected issues which are then reviewed towards the end of the iteration to define the final scope of the iteration (as several unfinished issues can be removed from the iteration). After a last stage where final verifications are performed on the quality of the iteration, the output of the iteration constitutes a potentially deployable release.

Figure I
Development by iteration



12. Keeping in mind that the eTIR international system should be developed once and then properly operated and maintained indefinitely, ECE has also chosen to adopt several practices from the DevOps movement which aim at preventing issues that may arise while moving from the development phase to the operational phase of the project. These practices are the following (which are further described below): invest in automated testing, rely on continuous integration, analyse telemetry and perform blameless post-mortems.

[6] See agilemanifesto.org

4. Development guidelines

13. Standard coding guidelines and abundant IT literature[7] on the subject constitute the cornerstone of the development guidelines. The underlying technology stack of the eTIR international system is Java and the IT experts use a modern and renowned integrated development environment (IDE) to program effectively with this language and related ecosystem. This IDE also allows integrating some of the development guidelines (access to the version control system (VCS), static code analysis tool, code formatting rules).

14. The IT experts use Git as the VCS of the eTIR international system and follow the usual best practices related to this product. Modifications brought to the source code are regularly committed and pushed to the central repository to be shared between all developers and prevent any loss of work in case of a workstation malfunction. Large developments are usually performed on separate branches. Finally, pushing code changes to the central repository requires prerequisites steps (detailed in the next sections) to ensure that each contribution has a high quality.

5. Logging guidelines

15. The logging service of the eTIR international system is very important as it produces the data needed for the non-repudiation system and for producing the metrics required to monitor the global health of the system. As explained in DevOps practices, these metrics (or telemetry) are the only way for the IT experts to monitor the eTIR international system in operation, and to be alerted of any issue arising and, thus, being able to efficiently resolve the issue even before being contacted by end-users.

16. The logging service generates several files, each having its own function. Each entry to a logging file is accompanied with the date and time information when it occurred and a potential severity:

- **eTIR messages**: all contents of the incoming and outgoing messages are saved into a file to store the entire communication threads between the eTIR international system and the information systems connected to it. This data is then used by the non-repudiation system and can be retrieved upon request by the contracting parties to the TIR Convention;
- **Database**: all queries to the eTIR database are saved in a file along with the time needed to perform these queries. This allows to continuously measure the performance of these queries and give indicators to the IT experts to identify and remove potential bottlenecks as well as better plan for future scalability requirements;
- **ITDB**: all calls made to the interface with the International TIR Data Bank (ITDB) are saved in a file along with the time needed to perform these queries. This allows to continuously measure the performance of these calls and give indicators to the IT experts to further optimize this interface;
- **Application**: all events occurring in the eTIR web services module are saved in a file to store the whole history of events which is used by the monitoring system to alert on any serious issue occurring in real time with the eTIR international system. This data is also used when investigating a past issue to identify its root cause.

6. Testing guidelines

17. Tests are a vital part of software engineering. IT history consistently shows that without proper attention dedicated to this aspect, software projects have a substantially higher probability to fail. Tests can either be manually or automatically executed. In the case of manual execution, the tester follows a series of steps to interact with the information system to be tested and compares the actual results he or she gets with the expected results. If they match, the test is successful and if not, it is a failure. Manual tests are the most

[7] In particular from authors Kent Beck, Martin Fowler and Robert C. Martin

obvious action a software engineer can immediately apply on a newly developed piece of software to verify if it works as expected. However, the biggest impediment of manual tests is that they rely on a person to execute them, which is not cost-effective and error prone. Moreover, they only verify the state of the system at the moment they are executed and their result (success/failure) is therefore no longer relevant when the conditions change (the source code is updated, the environment settings are updated, etc.).

18. Nowadays, in modern software engineering practices, it is acknowledged that manual tests are no longer sufficient to ensure high reliability and quality for the information system being developed. As explained in the related DevOps practices, tests now need to be automated to be executed on specific frequent events (when the conditions change, as mentioned above) to ensure that no regressions are introduced. Indeed, when implementing new features or correcting defects in the source code, software engineers always risk bringing in unwanted side-effects (like defects). In order to solve this inherent problem of software engineering, automated tests have to be implemented to verify any change being brought to the source code. It is important to keep in mind that this investment in time in implementing automated tests always pays off. Indeed, when automated tests are absent, the number of defects is much higher, and the time needed to investigate and correct them is substantially higher than the time needed to implement automated tests. In addition, regular issues experienced with systems because of defects can frustrate their users and cause severe reputational damage to the entity in charge of the system.

19. There are several types of automated tests that have their own characteristics that complement one another:

- **Unit tests**: tests written to check that a piece of software (known as the "unit") meets its design and behaves as intended. In object-oriented programming languages like Java, the unit is often an entire interface, such as a class, but could also be an individual method. The goal of unit testing is to isolate each part of the program and show that the individual parts are correct. A unit test provides a strict, written contract that the piece of code must satisfy. Unit tests are usually quick to implement and then to execute;
- **Integration tests**: tests written to check that software modules are combined and tested as a group. Integration testing is conducted to evaluate the compliance of a system with specified functional requirements. It occurs after unit testing and before validation testing. Integration testing takes as its input modules that have been unit tested, groups them in larger aggregates, applies tests defined in an integration test plan to those aggregates, and delivers as its output the integrated system ready for validation testing;
- **Performance tests**: tests written to check that a software system meets its performance requirements. This family of tests also includes tests written to simulate a given load (high number of queries) applied to the software. This type of tests is important to verify that the performance of the software does not degrade over time, in particular when new features are added;
- **Validation tests**: tests written to check that a software system meets its specifications and that it fulfils its intended purpose. Usually these tests are the most complex and costly to implement and maintain, as they involve simulating actions performed by end users on the user interface (UI) of the system. In the specific context of the eTIR international system, there is no UI, as data is exchanged automatically with the information systems of the other eTIR stakeholders using the eTIR messages. This approach allows for a very easy and efficient way to perform validation tests, as each test request message sends back a response message that can be validated to ensure the system behaves as expected;
- **Conformance tests**: similar to the validation tests, this type also includes, in the context of the eTIR system, the necessary tests to ensure that a representative set of simulated TIR transports is properly managed by sending and receiving a specific sequence of eTIR messages that are checked to validate entire scenarios. These tests can also focus on testing the information system of one particular eTIR stakeholder,

or they can also include several of them to better replicate real TIR transports following the eTIR procedure.

20. When writing automated tests, software engineers also need to ensure that most (if not all) relevant lines of the source code are covered and validated. In particular, software engineers need to ensure that all paths in the source code are covered with tests (this practice and related metrics are called “branch coverage”). In addition to an appropriate “code coverage”, software engineers also need to make sure that the assertions validating the source code are pertinent and comprehensive, otherwise the tests are not achieving their objective.

21. As described above, achieving a good code coverage is the only sustainable way to develop and maintain an information system and the IT experts have integrated this objective and related practices in the development processes. When a new feature is implemented, the appropriate number of unit and validation tests should be written to meet the code coverage objective. When a defect is corrected, one or more tests should be written to prevent the same issue from occurring again.

7. Static code analysis

22. Static code analysis consists in automatically examining the quality of the source code of a piece of software without actually executing this software. This examination is performed by a tool which is loaded with programming rules and best practices, most of them being defined over the years by the worldwide community of IT experts. Static code analysis is a very efficient way to perform a first check on the quality of the source code and an excellent complement to targeted manual code reviews performed by the IT experts on the source code.

23. While acknowledging the usefulness of this type of automated tool, the IT experts also recognize the need to jointly review the pertinence of several rules, given the specific context of the eTIR international system. As a result, the IT experts configure rules and their severity to best match this context.

24. Static code analysis is regularly performed on the entire source code of the eTIR international system and, in addition, the IT experts also benefit from the integration of this capability in the IDE they use for programming, which gives them an immediate feedback on the quality of the code they produce.

25. The objective is to progressively increase the quality of the source code and maintain it at a very high level throughout its lifecycle. This increases the reliability and maintainability of the source code and, eventually, saves time to the IT experts, which increases their productivity. This objective is performed in two phases: progressively increasing the quality of the source code and maintaining it at a high level.

26. During the first phase, the IT experts set low quality gates[8] in the static code analysis tool and correct as many issues as needed to meet these targets. Once these low targets are met, they are gradually increased, and the IT experts continue working on solving issues to meet the new targets. Once the quality gates reach a level deemed sufficient by the IT experts[9] (also by taking into account the Pareto principle), the second phase can start.

27. During the second phase, the objective is to continue developing and maintaining the eTIR international system while continuing to meet all quality gates. Additional measures can be put in place to send a notification to the IT experts if one of these quality gates is breached following the update of the source code, so that the IT experts can immediately look into this issue to solve it.

[8] A quality gate is a quantitative target set a on a particular criterion (e.g. «Less than 10 critical issues», «More than 40% of the source code covered with tests»)

[9] As detailed in the reliability requirements of the eTIR international system

8. Continuous integration (CI) pipeline

28. In software engineering, continuous integration (CI) is the practice of merging all developers' working copies to a shared mainline several times a day. This practice is not new (it dates from the 1990) and was continuously refined and expanded to finally get to the current DevOps practices, known as continuous integration and continuous deployment (CD) or CI/CD. The IT experts have chosen to focus on CI to start with, and once the appropriate maturity level is reached, they may consider also adopting the CD practice, which requires solid foundations.

29. Nowadays, the definition of CI reflects to the automation of all steps related to the integration and verification of changes in the source code of a software. CI allows software developers to get quick feedback on the quality of the code they commit to the VCS by executing all automated tests against a newly built and deployed version of the software, which contains the latest modifications brought to the VCS. CI relieves software developers from mundane, error-prone tasks related to building, testing and deploying a new version of the software, so that they can concentrate on where they have the best added value: to deliver features to the clients.

30. The IT experts have put in place a CI pipeline which consists of a specialized tool in which several actions are defined and configured to execute as successive automated steps. These steps are executed every time one of the IT expert commits a code change to the VCS. These steps are the following:

(a) **Build**: the CI pipeline detects that a commit was added to the VCS and will retrieve the latest version of the source code and build the new software components affected by the code change;

(b) **First testing phase**: automated unit and integration tests are then executed against the newly built software components to verify no regression was introduced with the code change;

(c) **Deployment on the SIT**[10] **environment**: the newly built software components are deployed on the SIT environment as a fully functional instance of the eTIR international system;

(d) **Second testing phase**: automated validation tests are then executed against the new instance of the eTIR international system to continue verifying, at the highest level, that no regression was introduced with the code change.

31. If an error happens during one of the steps (for instance if even only one test fails), the CI pipeline stops and a notification of failure is sent to the IT experts on their collaboration platform. The time of execution for all steps should not take more than 30 minutes to ensure quick feedback to the IT expert who commits a change to the VCS. This CI pipeline combines several best practices described above and is an excellent way to ensure high reliability of the eTIR internationals system and increase the productivity of the IT experts.

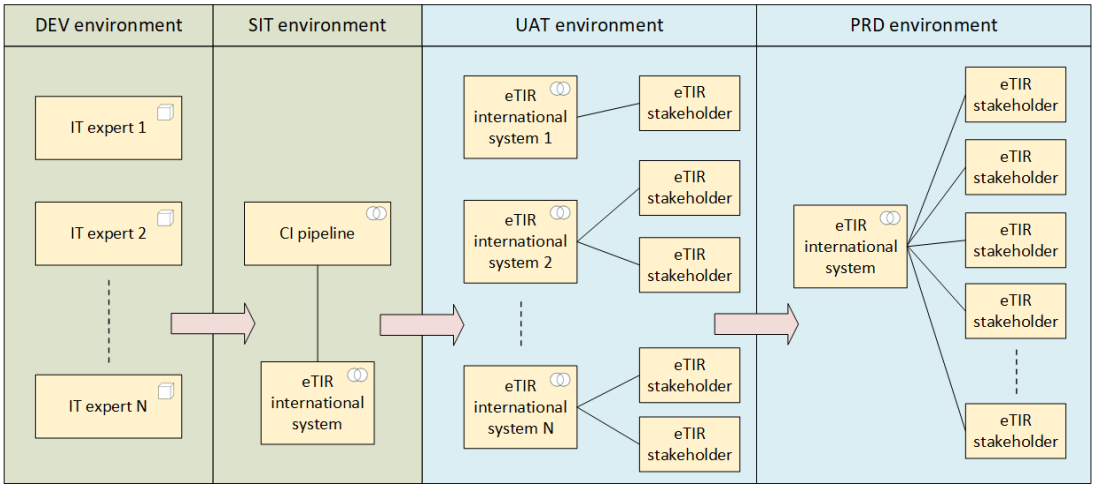
9. Environments

32. Following modern best practices from the IT industry, the IT experts have set up and configured four different environments to develop and maintain the eTIR international system in the best conditions. One of the challenges in managing several environments is to limit the number of variances between each of them to avoid defects linked to a specific environment. Specific development procedures are set up and followed by all IT experts to limit the probability of occurrence of this type of defect.

33. The figure below shows the different environments, which are then described in the following paragraphs.

[10] System Integration Testing (SIT), see next section for more information.

Figure II
Environments of the eTIR international system



34. **Development (DEV) environment**: each IT expert has his/her own workstation on which s/he can develop and test a local copy of the eTIR international system without interfering with the work of others. Once a code change has been prepared and tested, the IT expert commits it to the VCS so that it can be automatically deployed and tested on the SIT environment by the CI pipeline.

35. **System Integration Testing (SIT) environment**: this internal environment is used by the CI pipeline as a temporary location where newly instances of the eTIR international system are built, deployed and automatically tested. Once a set of code changes are validated on this environment, the IT experts may decide to build and deploy the latest version of the eTIR international system to the UAT environment.

36. **User Acceptance Testing (UAT) environment**: this environment is accessible by eTIR stakeholders to perform tests in the context of their interconnection projects. Several instances of the eTIR international system are available and each eTIR stakeholder gets access to one or more of these instances. Conformance tests of the eTIR international system and of the information systems of the eTIR stakeholders are also performed in the UAT environment. Once a version of the eTIR international system has been extensively tested in the UAT environment, it can be moved to the PRD environment.

37. **Production (PRD) environment**: this environment holds a unique instance of the eTIR international system which is only accessible to the eTIR stakeholders that have completed their interconnection project. This “live” environment is the only one used for performing TIR transports following the eTIR procedure.

10. Database guidelines

38. The eTIR database uses a database management system (DBMS) to record the information received in the eTIR messages. This component is the core of the eTIR international system and its development and maintenance should be treated with the utmost care.

39. The structure of the eTIR database was inherited from the eTIR pilot projects and the IT experts have identified several opportunities for improvement and optimization which are planned to be progressively implemented. The IT experts use a specialized tool, Liquibase, to track, version and apply database schema (structure) changes. Furthermore, this library also allows managing changes applied to the master and reference data stored in the database.

40. In the context of the eTIR system, “master and reference data” refers to data about the parties, the roles, and data used to classify or categorize the data processed and stored from the eTIR messages (e.g. eTIR stakeholders identities, country codes, guarantee types, goods classification, etc.). This data changes on rare occasions and needs to be meticulously managed.

41. Using this tool also allows to easily check which changes have been applied to the various copies of the eTIR database, present in all environments listed in the previous section. This is important to ensure that a recent change implemented to the schema or the master and reference data is consistently applied on all environments following the relevant release management procedures.

11. Issue management

42. One of the cornerstones of the adopted agile methodology is a defined and effective issue management. In this context, an issue can represent a feature request, a change request or a defect report. All changes to the eTIR data model, to the source code or to the documentation of the eTIR international system first need to be logged into the issue tracking system of ECE. This is essential to ensure proper traceability of all changes and allows verifying that only authorized changes are applied.

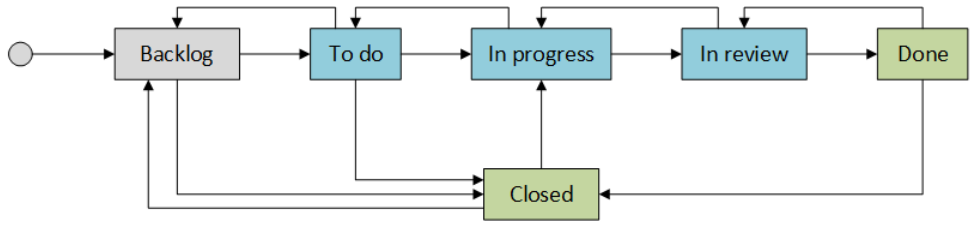
43. When logging an issue into the issue tracking system, an IT expert ensures that all necessary details are documented so that any other IT expert should be able to understand what needs to be done. This is also a prerequisite to ensure retaining institutional memory without being affected by potential turnover in ECE.

44. The IT experts have agreed on a series of activities that should be performed during the various stages of the lifecycle of any issue before it can be considered as completed. The stages are named with the different statuses of issues. This is the “definition of done” and it is defined as follows:

- **Definition of done (DOD)**: is when all conditions, or acceptance criteria,[11] that an issue must satisfy, are met. The objective is to ensure a proper level of quality and reliability of the system, at all times. The investment in time spent on all of these activities always pays off in terms of preventing defects from being deployed to the PRD environment. Having less defects prevents from spending time and stress in troubleshooting and avoids impacting the reputation of ECE.

45. A newly created issue gets assigned the “backlog” status, symbolizing its belonging to the eTIR backlog, and a priority is also assigned to it. Issues are the atomic work packages that are assigned to the IT experts by the IT coordinator when they have been selected in the iteration backlog. The following figure shows the issue lifecycle with the various statuses that an issue can take, and the following list describes them.

Figure III
Issue lifecycle



- **Backlog**: the issue has been identified and logged into the issue tracking system but is not yet selected for processing;
- **To do**: the issue has been selected to be worked during an iteration and is assigned to an IT expert who needs to complete the steps related to the "To do" stage of the DOD (see below);
- **In progress**: the issue is being processed by the IT expert who needs to complete all steps related to the "In Progress" stage of the DOD;

[11] The conditions and acceptance criteria are defined later on in the section.

- **In review**: the issue is being reviewed by another IT expert to check several aspects related to quality assurance by following all steps related to the "In Review" stage of the DOD;
- **Done**: the issue is done (implemented and reviewed) and it will be finally validated by the IT experts during regular meetings where all deployed issues on the PRD environment are finally closed;
- **Closed**: the issue is either deployed (coming from "Done") or closed as it will not be corrected or it is considered as a duplicate of another issue (coming from "Backlog" or "To do").

46. The DOD describes the following key objectives and acceptance criteria for the abovementioned stages:

- **To do**: the issue is sufficiently detailed and has enough background information so that it can be understood by any other IT expert, and a first estimation of the time needed is made;
- **In progress**: the change needed is entirely performed across all appropriate IT assets (eTIR data model, source code, documentation). All exigencies in terms of quality and reliability are met (including the verifications performed by the CI pipeline and the static analysis tool) and all applicable guidelines are followed;
- **In review**: the outputs of the tasks performed during the “In progress” stage are checked by another IT expert. In particular, the test coverage for the source code updated is verified.

12. Documentation guidelines

47. ECE maintains three types of documentation related to the eTIR international system. The first type corresponds to the eTIR specifications, for which amendment procedures are described in Article 5 of Annex 11 of the TIR Convention.

48. The second type corresponds to the internal documentation that is necessary for ECE to properly develop, operate and maintain the eTIR international system. This documentation is prepared and updated by the IT experts of ECE and is managed on a secured KMS that offers versioning capabilities to properly store institutional memory. The internal documentation contains confidential information about, inter alia:

- Development: guidelines, technical documentation, training, stakeholder’s documentation, related standard operating procedures (SOPs), etc.
- Management: team administration, meeting notes, related SOPs, etc.
- Operations: connection with contracting parties, environments, eTIR service desk, related SOPs, etc.

49. The third type corresponds to the documentation that is produced by ECE for the eTIR stakeholders to interconnect their information systems with the eTIR international system. These documents are shared with the eTIR stakeholders on the web site[12] dedicated to eTIR. These documents are produced in addition to the eTIR specifications to facilitate the interconnection projects and benefit from the feedback received during these projects. They are a way for ECE to continuously clarify various aspects of the eTIR system in a more frequent and flexible way. All these documents are always fully aligned with Annex 11 and with the version of the eTIR specifications on which they are based.

13. Version management

50. ECE manages the source code of the eTIR international system and the changes applied to the schema and “master and reference data” of the eTIR database with a VCS.

[12] See www.etir.org/documentation

ECE has selected Git as its VCS and uses an internal and secured platform as the central Git repository.

51. The IT experts follow the usual best practices from the IT industry related to Git, and especially the ones from DevOps. In particular, IT experts should frequently commit and push their code to the central Git repository, after having performed all tests locally to ensure that this would not create a failure during the execution of the CI pipeline. Each commit should feature changes related to only one issue and the comment of the commit should clearly mention the issue to which it is related and describe the substance of the changes.

52. Branches are created and used in several cases. Firstly, they can be created by an IT expert who needs to work on a complex feature that cannot immediately be committed on the master branch. Once the feature is completed and tested, the branch is merged back into the master branch. Secondly, a branch is created every time a version of the eTIR international system is released on the PRD environment, following the release management guidelines. Tags are also created when a new version of the eTIR international system is deployed on the UAT environment or on the PRD environment.

53. Regarding the version number of the eTIR international system, ECE has selected an approach which uses the following three numbers:

- **Major version number**: it is incremented when a breaking change happens on the API which allows eTIR stakeholders to connect to the eTIR international system. It may also be incremented when a substantial change is brought to the eTIR international system without changing the API.
- **Minor version number**: it is incremented in any other case than the ones that affect the major of the hotfix version numbers. When the major version number is incremented, the minor version number is reset to 0.
- **Hotfix version number**: it is only used when one or more hotfixes need to be deployed on a version which is already deployed on the PRD environment, without willing to create a new version of the eTIR international system.

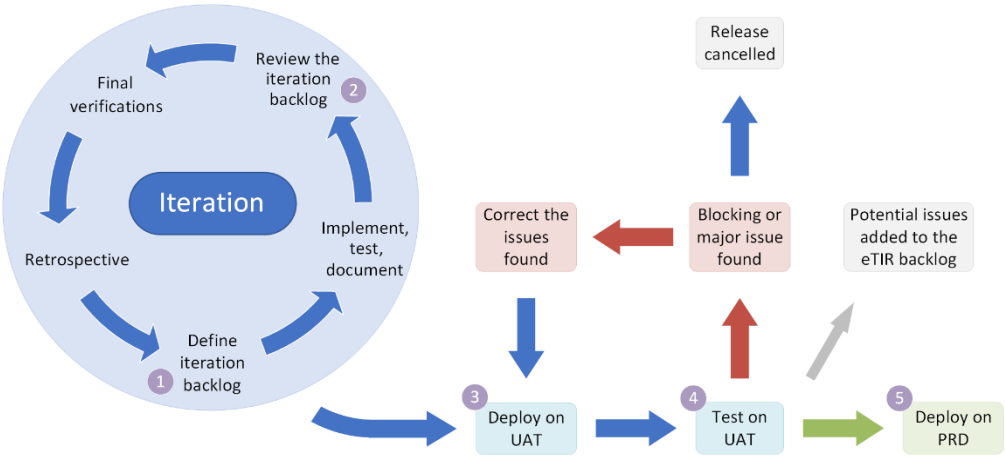
54. The major and minor version numbers, as well as the hotfix version number, if it exists, are always updated simultaneously on all software components of the eTIR international system and represent its version number under the form XX.YY.ZZ with XX being the major, YY the minor version numbers and ZZ the hotfix version number (ignored if equal to 0). Here are two examples of the version number for the eTIR international system:

- **eTIR international system 4.15**, where 4 is the major version number and 15 is the minor version number (frequent case).
- **eTIR international system 4.15.1**, where 4 is the major version number, 15 is the minor version number and 1 is the hotfix version number (rare case).

14. Release management

55. Release management is the process of managing, planning, scheduling and controlling a software build through different stages and environments; including testing and deploying software releases. In the context of the eTIR international system, it refers to the process described in the following figure and stages.

Figure IV
Release management process



(a) **Define the iteration backlog**: the IT experts select from the eTIR backlog which issues should be worked on during the iteration and determine the version number of the new release. Each release has its own, unique version number which is mandatory if the release is to be deployed on the UAT or PRD environments.

(b) **Review the iteration backlog**: the IT experts review which issues are considered as "done" and modify either the iteration length or the list of issues assigned to this version. In the end, all issues are completed, tested and documented and the quality gates are passing on the SIT environment. The release notes, which explain the changes brought by this new version, are prepared.

(c) **Deploy on UAT**: the eTIR stakeholders working on the instances of the eTIR international system are informed about the new deployment to come. Then the new version is deployed on all instances of the eTIR international system and the related eTIR databases are reset. The release notes are communicated to the eTIR stakeholders.

(d) **Test on UAT**: the newly deployed release will then be tested by the eTIR stakeholders during a given period of time agreed by all parties. The IT experts determine if a new execution of the conformance tests is needed or not. Any issue found will be raised to the eTIR service desk to be logged and categorized. If one or more blocking or major issues are found, then either they are corrected, or the current release is cancelled and a new one will be prepared which will include as a priority the issue(s) to be corrected. If these issues are corrected, the updated release needs to be deployed on the UAT environment and tested again by all eTIR stakeholders for a given period of time before being validated. Minor issues can be added to the eTIR backlog to be fixed in a subsequent release.

(e) **Deploy on PRD**: if no major issue was reported after a given period of test on UAT, the release can be scheduled to be deployed on the production environment after a proper communication to the eTIR stakeholders. The details about the new version, including the release notes, are published on the eTIR web site. Once the deployment is performed on this final environment, the eTIR service desk actively monitors the telemetry to verify everything is working correctly.

56. Then, if an issue is identified on the production environment, three cases can happen:

(a) **The issue is blocking**: the IT experts roll back the PRD environment to the previous release and inform all eTIR stakeholders accordingly.

(b) **The issue is major**: the IT experts quickly prepare a hotfix, perform all tests needed on the SIT environment and deploy it on the PRD environment to correct the issue. All eTIR stakeholders are informed accordingly.

(c) **The issue is minor**: the issue is logged and added to the eTIR backlog to be fixed in a subsequent release.

B. Maintenance processes

1. Introduction

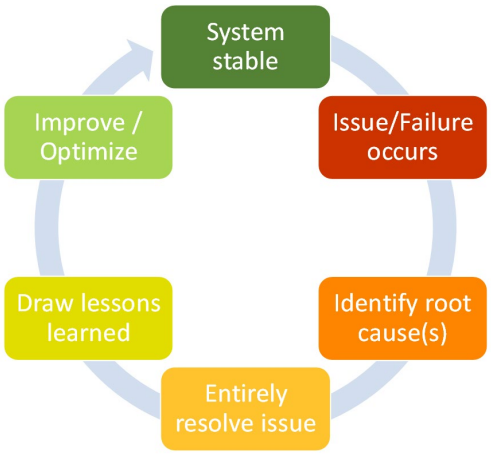
57. This section describes the processes followed by the IT experts of ECE to support and maintain the eTIR international system to ensure it functions correctly, to properly deal with issues and to anticipate and prevent possible problems in the future. This section also describes the procedure to be followed by the eTIR stakeholders when reporting an issue and informs of the internal activities performed to address it.

2. Continuous improvement

58. One of the underlying principle of the DevOps practices is about adopting a continuous improvement approach. It means that none of the outputs created (software, processes, documentation, etc.) are ever final as they can always be improved. Especially if an issue (a defect in the system, a flaw in a process, an omission or imprecision in the documentation) is raised, it should always be considered as an opportunity to improve. This principle is similar to the one used in the Deming cycle or PDCA.[13]

59. With this approach, the IT experts recognize the importance of always seizing the opportunity to learn from issues to ensure that the same issues should not happen again in the future (or, at least, that the actions taken decrease the probability for a future occurrence). In particular, it is important to take the time to identify the root cause(s) of the issue to be able to entirely correct them and improve or optimize the processes, if possible. This approach is also applied in development processes but it is especially important in maintenance processes as their main objectives are to solve and prevent issues. The main processes mentioned above are shown in the following figure. They are also further explained in the next sections.

Figure V
Continuous improvement process

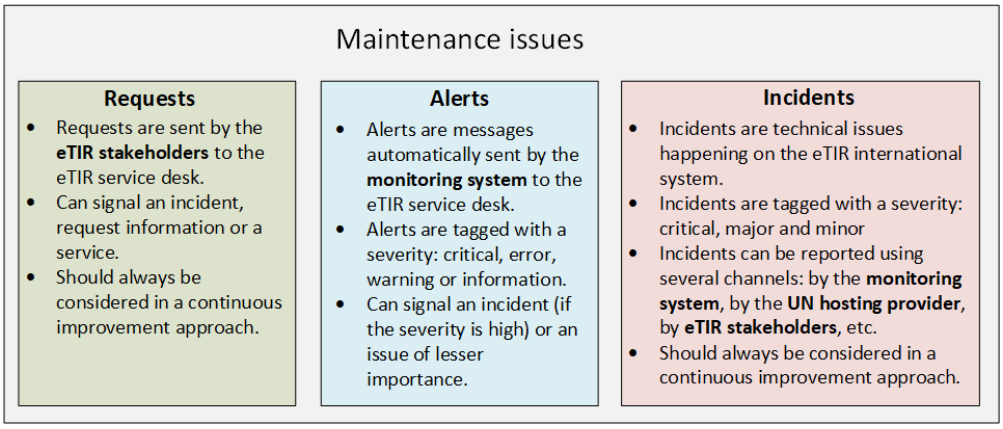


3. Issue management

60. In maintenance, there are three different types of issues that have their own characteristics and that are handled using specific procedures. The following figure describes these three types of issues.

[13] See en.wikipedia.org/wiki/PDCA

Figure VI
Types of maintenance issues



61. Requests are further described in the section related to the eTIR service desk. Alerts are further described in the section related to monitoring management. Incidents are further described in the section related to incident management.

4. eTIR service desk

62. The eTIR service desk is the single point of contact (SPOC) for eTIR stakeholders to raise any request related to the eTIR system. It is possible to do so by sending messages to its email address (etir@un.org) or through the "contact us" form of the eTIR web site.[14] The eTIR service desk is composed of the IT experts and subject matter experts of the TIR Convention of ECE.

63. Requests received by the eTIR service desk are dispatched by a (Tier-1) service desk agent to the appropriate (Tier-2) expert, depending on the nature of the request. Requests that signal an incident or a technical issue are dealt with as a priority.

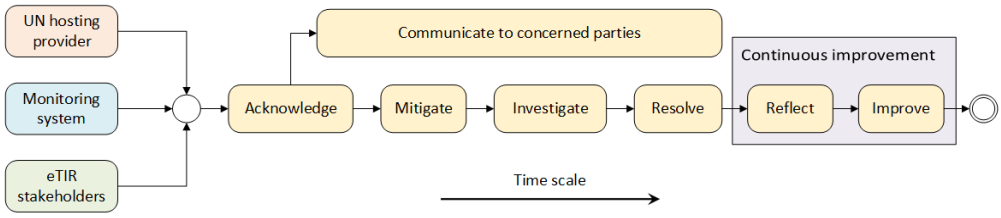
64. In the context of the interconnection projects, the eTIR service desk assists the eTIR stakeholders in connecting their information systems to the eTIR international system. These projects are closer to the development processes and, during the project initiation stage, the eTIR stakeholders define the best ways of communicating with the eTIR service desk to get information and raise any request. Given the limited resources of the eTIR service desk, the scope of its assistance is limited to providing information and guiding the experts of the eTIR stakeholders in their interconnection projects. For instance, the eTIR service desk cannot directly perform changes into the information systems of the eTIR stakeholders to connect them to the eTIR international system.

5. Incident management

65. Incidents are generally technical issues with significant consequences that need to be addressed by the eTIR service desk as a priority. Incidents have a severity associated to them which drive the type of answer that need to be given: critical, major and minor. The whole process to manage them is inspired from the Information Technology Infrastructure Library (ITIL) service management methodology and is described in the following diagram. Its stages are further described hereunder.

[14] See www.etir.org/contact-us

Figure VII
Incident management process



(a) **Acknowledge**: after having been alerted, the IT experts confirm the incident (not a false positive) and ongoing (not already solved). They define its scope (the affected components), its severity and the list of concerned parties. From that point, all actions are logged to be further analysed during the "Reflect" stage;

(b) **Communicate to concerned parties**: transparent communication to the concerned parties about the incident is essential so that information can be given of the estimated time needed to resolve the issue, as this can drive decisions of the parties to apply specific measures (e.g. fallback procedures). The IT experts decide on the content and frequency of the communication until the incident is resolved (step (e));

(c) **Mitigate**: if possible, mitigation measures are applied in order to either decrease the severity of the issue, or to temporarily resolve it;

(d) **Investigate**: the IT experts take the time needed to comprehensively investigate the incident and determine its root cause(s);

(e) **Resolve**: after the investigation, the root cause(s) are addressed and corrected and the incident must be considered as resolved before moving to the next stage;

(f) **Reflect**: the IT experts gather all data and actions performed so far to resolve the incident and have a "blameless post-mortem" meeting. The goal is to take a deeper look at the incident and figure out what happened, why it happened, how the IT experts responded, and what can be done to prevent repeating this type of incident as well as improve future responses; while assuming the responsibility of the incident collectively. An "incident report" is prepared during this meeting and follow-up actions are defined and planned accordingly;

(g) **Improve**: the follow-up actions that have been defined in both previous stages are progressively selected from the eTIR backlog as per their priority, and performed to improve the software, processes, documentation and other assets so that the probability of having the same incident happening is decreased.

66. During the "Reflect" stage, the IT experts prepare an incident report which is then stored in the knowledge management system (KMS) for institutional memory. This report contains the following information about the incident (including date and time when applicable): severity, description, services affected, how it was notified and by who, response actions performed to mitigate and then solve it, communication sent and received, results of the investigation, list of root causes, lessons learned from the blameless port-mortem and list of follow up actions.

67. With this process, the IT experts wish to achieve the following benefits: the prevention of similar incidents (or at least decreasing their probability to happen), an improvement on the average time to resolve incidents, a further reduction of downtime of the eTIR international system and an overall improved experience for the eTIR stakeholders.

6. Incidents managed by the United Nations hosting provider

68. As displayed in figure VII, incidents can be reported to the eTIR service desk by the United Nations hosting provider which hosts the eTIR international system. A service level agreement (SLA) is signed with this provider to ensure a 24/7 support of the eTIR

international system. SOPs are prepared by the IT experts for the officers of the United Nations hosting provider so that they can respond to specific types of incidents.

69. When an incident occurs, the officers of the United Nations hosting provider are notified by alerts sent by the monitoring system and they respond using these SOPs. If the response resolves the incident, they notify the eTIR service desk, for further investigation, mentioning that the incident is closed. If the response does not resolve the incident, they escalate it by contacting the eTIR service desk as displayed in figure VII, using various communication ways and procedures depending on the severity of the incident.

7. Backup and restore management

70. Backup and restore management represents the strategy and related procedures put in place to ensure that copies of eTIR related data are frequently made and can quickly be restored, in case of a data loss event. Indeed, data can be lost during several types of events, inter alia: the malfunction of a server, fire in the data centre or a cyberattack. The preparation of the SOPs are the joint responsibility of the United Nations hosting provider and ECE and they are mentioned in the SLA.

71. Data stored in all eTIR storage locations (the eTIR database, the eTIR logs and the eTIR documents) is backed up twice per day. This backed up data is securely stored in, at least, one other location than the primary site to avoid being destroyed if this site sustains a disaster. It is also not accessible from the same network to avoid being compromised by a cyberattack of type ransomware. Only the most recent and complete backups are kept and old backups are erased.

72. Finally, it should take no more than 6 hours to store the last backup in case of data loss event. Tests are regularly performed with the United Nations hosting provider to ensure that this requirement can be met.

8. Monitoring management

73. The act of monitoring an information system includes the collection of information produced by this system and the ability to produce alerts when certain events are met, so that (automated or manual) actions can be performed as answers to these events. Monitoring a system allows to proactively detect any issue that may turn into a failure and may eventually impact the availability of the system. The ability to quickly respond to these early warnings usually decreases the impact of failures and can also sometimes prevent them altogether.

74. A monitoring system is provided by the United Nations hosting provider and it is configured in collaboration with ECE to observe the resources and performance of the virtual servers, as well as the availability and performance of the various services of the eTIR international system. In particular, the list of indicators that are tracked by the monitoring system include the following metrics: CPU usage, RAM usage, percentage of disk used, processes, availability of the services, system’s response time and resource usage of applications.

75. Alerts are configured to be triggered when specific thresholds are exceeded. Alerts have a severity associated to them which drives the type of answer that needs to be given: critical, error, warning and information. Several types of answers can be activated depending on their configuration: an automated process can be executed or a communication can be sent to one or more persons (by email, SMS or phone call) to notify them of the alert so they can take action as soon as possible. The first persons notified are usually the officers of the United Nations hosting provider so they can take immediate actions by applying the SOPs prepared for such occasions. Alerts can also be sent to the eTIR service desk, depending on the urgency and importance of the issue. A comprehensive list of indicators, thresholds, alerts and related answers are jointly documented by the United Nations hosting provider and ECE, and are mentioned in the SLA.

76. In addition to tracking metrics gathered from the virtual servers and processes, the monitoring system also exploits data contained in the eTIR logs. This information, also referred to as telemetry, logged by the eTIR international system, provides valuable data

which can be used to detect any potential immediate issue with the system. It also informs about the performance of the system and gives an indication to the IT experts on related trends. It is important to track this data to ensure that target values set in the technical requirements of the eTIR international system are met.

77. Finally, it is important to take into consideration one drawback usually associated with the practice of monitoring. When initially configured, thresholds and alerts can lead to false positives or, on the contrary, they can "miss" issues that should have been detected. For this reason, the practice of continuous improvement is particularly relevant and the configuration of the monitoring system should regularly be reviewed to be optimized.

9. Patch management

78. A patch is a set of changes to a piece of software designed to update, correct, or improve it. This includes fixing security vulnerabilities and other defects. In this document, patch management refers to the strategy and related procedures put in place to ensure that all software components, including the operating systems of the underlying servers, are regularly patched to correct any recently found issues.

79. It is especially important to remove security vulnerabilities that are uncovered in existing versions of all software by the cybersecurity community. Regularly applying patches from authorized and verified sources is one of the most effective way to protect the eTIR international system from cyberattacks (see the part dedicated to the security of the eTIR system).

80. SOPs are prepared and applied on a regular basis (at least every three months) to patch the following software components, if a patch is available: underlying operating systems, frameworks and libraries (e.g. Java virtual machine) and database management systems. Regular schedules do not prevent applying important patches as needed, most of the time for security reasons. Software components are patched by the United Nations hosting provider and by ECE, depending on the responsibilities detailed in the SLA.

10. Upgrade management

81. An upgrade is generally the replacement of hardware, software or firmware with a newer or better version, in order to bring the system up to date or to improve its functionalities. In this document, upgrade management refers to the strategy and related procedures put in place to ensure that technical debt is regularly addressed and will not grow over time (see the maintainability requirements of the eTIR international system). Upgrade management differs from patch management as upgrades are new versions of software which need to be carefully tested to detect and address potential issues before they can be applied.

82. Replacing the hardware and associated firmware is the responsibility of the United Nations hosting provider. Regarding software, the responsibilities are shared between the United Nations hosting provider, which needs to plan and perform the upgrades of all software components under its purview (e.g. virtual server farm system, operating systems of the virtual servers), and ECE which needs to plan and perform the upgrades of all software components of the eTIR international system.

83. At least once per trimester, the latest versions of the underlying programming language, frameworks and libraries used to build the eTIR international system are checked. The IT experts then regularly review and document the various advantages and disadvantages to migrate a software component to one of its new versions. The following criteria are taken into consideration to decide when to plan such a migration: end of support date of the currently used version, maturity of the new version as assessed by the IT community, potential benefits regarding security and additional features.

84. When the decision is taken to migrate a software component to a new version, an internal project is launched and its associated tasks are included into the eTIR backlog to be prioritized and considered in the usual development by iteration approach. The objectives of this type of project are the following: comprehensively test the new version of the software component to detect any issues that may arise in the context of the eTIR international system, correct any major issue found, possibly take advantage of the new features brought by the new version to improve the eTIR international system, further test and validate on the UAT environment before eventually deploying a new version of the eTIR international system on the PRD environment.

III. Communication between eTIR stakeholders and the eTIR international system

85. This part describes all the technical aspects of the interfaces between the information systems of the eTIR stakeholders and the eTIR international system. The following facets are described: interconnection projects, the eTIR data model, implementation and test of the eTIR messages, requirements and recommendations applicable to eTIR stakeholders, the technical specifications of all eTIR messages.

86. The main objective of this part is to facilitate the interconnection of eTIR stakeholders with the eTIR international system (following the third guiding principle) and to reduce the risks of facing technical issues when performing this interconnection. The following sections intend to demonstrate the consistency and integrity of these interfaces by clearly specifying them, as well as expectations on every constituent part of the whole set of interfaces; while specifying the responsibilities between the parties involved.

A. Interconnection projects

1. Introduction

87. eTIR stakeholders who wish to become part of the eTIR system shall undertake an interconnection project to connect their information systems to the eTIR international system by following the eTIR specifications. ECE will assist countries in connecting their national customs systems to the eTIR international system, in line with paragraph 2 of Article 11 of Annex 11 of the TIR Convention.

88. The size of the project varies between the eTIR stakeholders and is mainly related to the following two factors:

- The differences between the data model and the procedures of the information systems of the eTIR stakeholder on one side and of the eTIR international system on the other side;
- The number of eTIR request messages to be implemented, which depends on the kind of eTIR stakeholders: 3 for holders, 4 for guarantee chains (7 including optional request messages) and 11 for customs authorities (13 including optional request messages).

2. Documentation

89. In order to assist eTIR stakeholders in the context of the interconnection projects, ECE produces additional documentation which is aligned with the eTIR specifications and with Annex 11 of the TIR Convention. The experience acquired and the feedback received during the interconnection projects allows ECE to continuously improve these documents by regularly publishing new versions, available on the eTIR web site[15].

90. Guides titled “Project guidelines” are addressed at the management team of the eTIR stakeholders. These guides describe the approach proposed by ECE to undertake the

[15] See etir.org/documentation

interconnection project. It details the project methodology, stages and milestones, communication management plan and teams involved, along with roles and responsibilities. Finally, it proposes a draft project plan in the form of a Gantt chart.

91. A common technical guide titled "Introduction to the eTIR web services" is addressed at the technical team of all eTIR stakeholders. This guide describes how to access the eTIR web services, how to implement and test the eTIR messages, and how to implement and configure the security features of the communication with the eTIR international system.

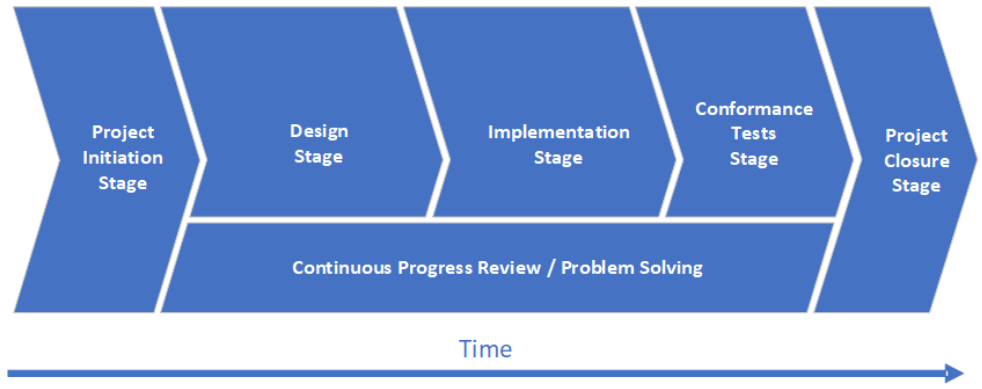
92. Finally, each pair of eTIR message has its own technical guide addressed at the technical team of the relevant eTIR stakeholders. Each guide recalls at which moment of the TIR transport these messages are used, their context and prerequisites, details the description and usages of all fields of both messages and gives examples on how to use them.

93. Most of the content of the technical guides is available in the next sections of the current part to give a comprehensive overview of the technical specifications of these eTIR messages and how to implement, test and use them. At the same time, these guides propose additional technical details and are updated more frequently than the eTIR specifications to better accompany the eTIR stakeholders during their interconnection project.

3. Stages

94. The stages proposed by ECE for the interconnection projects (which are described in detail in the "Project guidelines" document) are shown in the following figure. Their main objectives and activities are detailed hereafter where the team from the eTIR stakeholder, working on the project, is hereinafter referred to as the "project team".

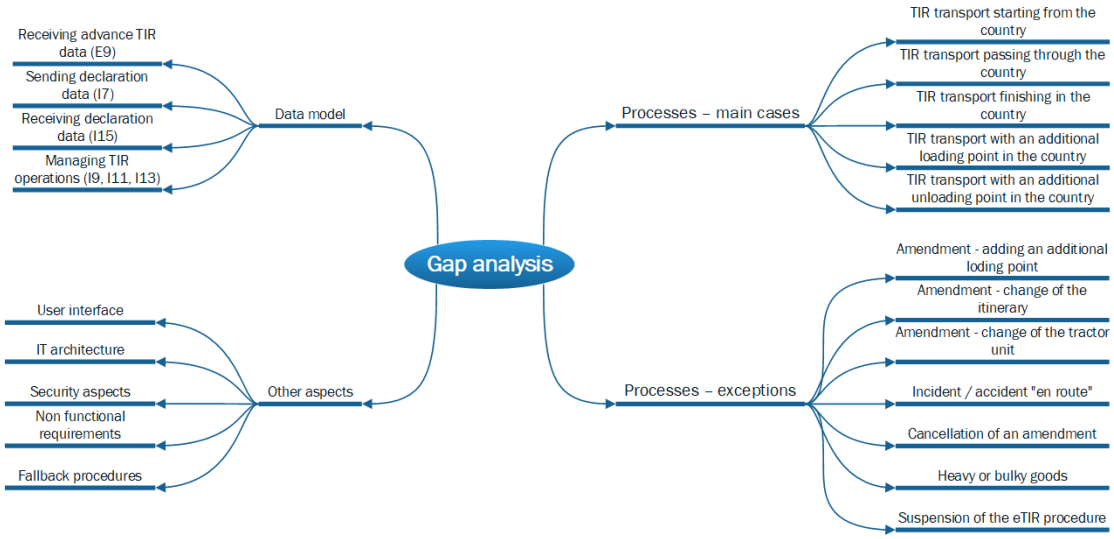
Figure VIII
Stages proposed for interconnection projects



95. During the **project initiation stage**, the project team is formed, decides on a project methodology, agrees with ECE on the collaboration arrangements, and start drafting a project plan.

96. During the **design stage**, the project team studies the eTIR specifications and the documentation applicable to them. They perform a gap analysis between several aspects of the information systems of the eTIR stakeholder on one side, and the eTIR international system and the eTIR procedure on the other side. The aspects considered during this gap analysis depend on the kind of eTIR stakeholder and, as an example in the case of customs authorities, these aspects are detailed in the following figure.

Figure IX
Aspects of the gap analysis for customs authorities



97. The objective is to identify the gaps between both sides and to find ways, in collaboration with ECE, to resolve them. Following this study, all changes that need to be applied to the information systems of the eTIR stakeholder are detailed and the relevant documentation is produced to describe to the IT experts of the project team (or the consultants) how to implement these changes. The end of this stage is an opportunity to update the project plan with the insight collected by the project team.

98. During the **implementation stage**, the project team implements and tests the changes that need to be performed on the information systems of the eTIR stakeholder, the interface to connect to the eTIR international system and the relevant eTIR messages. This stage is realized in close collaboration with ECE and by connecting to the User Acceptance Testing (UAT) environment of the eTIR international system.

99. During the **conformance tests stage**, the project team and ECE perform a comprehensive set of tests between the information systems of the eTIR stakeholder and the eTIR international system, to verify the conformity of the developed solution with the eTIR specifications. In the case of customs authorities, these tests simulate possible TIR transports and cases foreseen in the TIR Convention which can happen in the context of the eTIR procedure. If anomalies are found, the project may need to return to the implementation stage for correction or additional rework. Once all tests are successful, the project team deploys in production the new version of their information systems and train the relevant staff on how to use it.

100. As shown in figure VIII, the **continuous progress review stage** is a stage parallel to the others and it represents the continuous collaboration between the project team and ECE during the project. ECE stands ready to answer any question on the TIR Convention, the eTIR procedure and the eTIR specifications; help on the technical aspects and liaise with the project team to find solutions for identified gaps and resolve any issue that may arise.

101. During the **project closure stage**, the project team performs the usual project closure activities, draws the lessons learned from the project and hands over the results to the operations team.

4. Issues

102. During the project initiation stage, a communication management plan is discussed and agreed between ECE and the eTIR stakeholder and it includes how to raise, discuss and resolve issues that may occur during the project. These issues include how to solve the gaps identified during the design stage. If certain issues cannot be solved at the level of the project, they can be brought to the attention of TIB. In particular, the eTIR stakeholder may wish to bring a proposal to amend the eTIR specifications to be discussed by TIB.

B. The eTIR data model

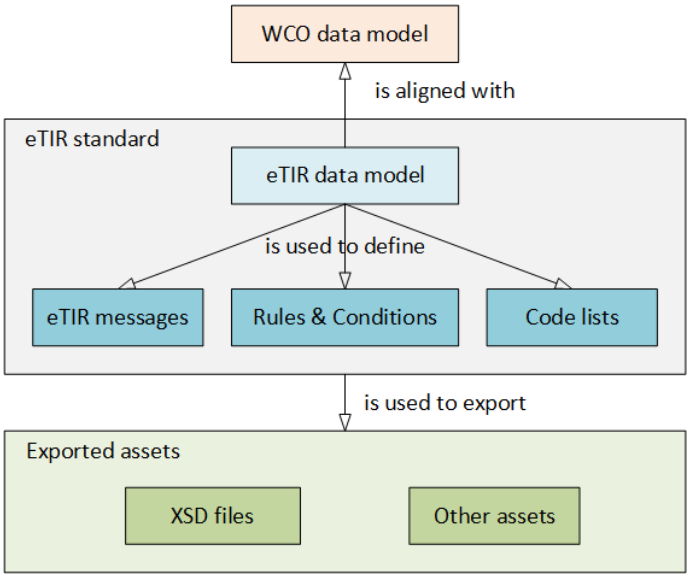
1. Introduction

103. The eTIR data model defines the various components of the messages allowing the information systems of the eTIR stakeholders to exchange information in the context of the eTIR procedure. This section details these components and their technical specifications.

2. Structure

104. The structure of the eTIR data model is shown in the following figure and its components are detailed thereafter.

Figure X
Structure of the eTIR data model



105. The **eTIR messages** define the structure of the data exchanged between the information systems of the eTIR stakeholders. They follow the structure of the eTIR data model and describe, in the form of a tree structure, the information needed in the context of the message. The nodes of the tree structure are called the classes and the leaves of the tree structure are called the attributes. An attribute is an elementary (atomic) piece of information and is part of a class. A class groups together one or more attributes and/or other classes related to the same subject. A field is the generic term to represent either a class or an attribute.

106. The **rules and conditions** define business rules that are translated as constraints on specific attributes and classes of the eTIR messages. The rules and conditions of the eTIR specifications are defined in the functional specifications document.

107. All fields have a **status** which defines whether the field is required in the message (status: “R”), optional (status: “O”), or if its status depends on the evaluation of a condition (status: “D”). In the latter case, a condition is also applied to the field.

108. The **code lists** define specific codes that are used in particular attributes of the eTIR messages. Using codes removes the problem of languages when transferring information and should always be preferred when selecting an item from a defined list. The descriptions of the codes can be translated, thus becoming multilingual. Furthermore, using a code list allows the system receiving the message to validate that attributes bound to a code list are using proper codes from the list. On the other hand, code lists should not be used when an attribute should contain an identifier, a numeric value or a free text value. The list of code lists is available in annex VI.F of this document.

109. All fields have an “eTIR name” which is specific to the eTIR data model and, therefore, might differ from the name given in the World Customs Organization (WCO)

data model. This name allows for a better understanding of the purpose of the fields. In order to remain compliant with the WCO data model, the names of the XML elements corresponding to these fields are not changed and remain the ones from the WCO data model. The following specifications are applied to the eTIR names:

- Attributes that represent an identifier are named “identifier”;
- Attributes that represent a code have their name ending with “, coded”;
- Attributes that represent a date have their name ending with “ date”;
- Attributes that represent a date and a time have their name ending with “ date time”;
- Attributes that represent either a date or a date and a time have their name ending with “ date time”.

110. The eTIR data model is developed and maintained by ECE, based on amendments approved by TIB, and it is then used to export several types of assets. These assets are used either to be integrated in documentation (like the eTIR specifications or the technical guides), or to be used in the context of the interconnection projects. Among them, XSD files are technical assets that contain the XML schemas definitions of each eTIR message and are defined in annex VI.E of this document.

3. Inheritance

111. The eTIR data model is entirely based, and fully aligned, with the WCO data model. This decision has set guidelines for the structure of the eTIR messages in particular and it yields several benefits, the most important being the similarity, consistency and interoperability between the eTIR data model and the data models of the national customs systems which also base theirs on the WCO data model. As a result, the interconnection between these national customs systems and the eTIR international system will be greatly facilitated, which will result in a small number of gaps between systems and thus, less time and cost to interconnect both systems.

112. During the development of the eTIR data model, it may be needed to request additions or updates to the WCO data model. In this case, ECE submits Data Maintenance Requests (DMR) to the Data Model Project Team (DMPT) sessions organized by WCO to ask them to amend the WCO data model to reflect the needs for the eTIR data model. As these changes are included later on, when the new version of the WCO data model is published, ECE is using and publishing meanwhile these changes as extensions to the WCO data model.

4. Updating the eTIR data model

113. The eTIR data model continues to evolve across the subsequent versions of the eTIR specifications. Many proposals for amending the eTIR data model (including the eTIR messages, rules and conditions and code lists) are submitted by ECE, based on the feedback gathered during the development of the eTIR international system. Working with the project teams of the eTIR stakeholders during the interconnection projects is another important source of amendment proposals. Finally, contracting parties and the guarantee chain can also propose amendments.

114. All these amendment proposals are submitted to TIB for consideration. TIB decides whether these amendments should be approved and if it is the case, requests ECE to include them in one of the next versions of the eTIR specifications. As for the source code of the eTIR international system, the eTIR data model is also versioned using the Version Control System (VCS) Git. This allows to easily develop and maintain several versions of the eTIR data model concurrently.

115. Keeping track of all the changes brought to the eTIR messages is done by means of “change logs”.. These “change logs” contain the following information: class and attribute impacted, nature, date and reason of the change, and indication in which version of the eTIR specification this change was applied. All changes are then aggregated by eTIR message and exported using reports to be made available in the section “Overview of changes” of the eTIR functional specifications document.

116. Originally, the version numbers of the eTIR data model had been decorrelated from the version numbers of the eTIR specifications. For version 4.3 of the eTIR specifications, the version number of the eTIR data model has been adapted to create a more obvious link. The following table shows the correspondence between the version numbers of the latest eTIR specifications, the eTIR data model and the related WCO data model on which they were based.

Table 1
Correspondence of the version numbers

Date of publication of the eTIR specifications	*eTIR specifications version number*	*eTIR data model version number*	*WCO data model version number*
4 March 2011	3.0	0.1	3.2.0
10 November 2013	4.0	0.2	3.3.0
25 November 2014	4.1	0.3	3.5.0
27 November 2017	4.2	0.4	3.7.0
To be determined	4.3	0.43	*(planned)* 3.11.0

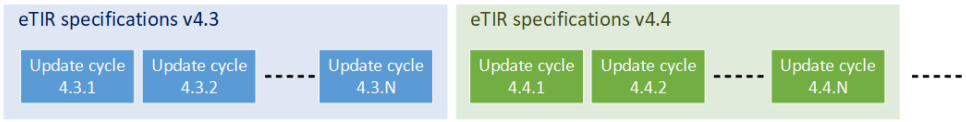
117. The version numbers of the eTIR international system are currently independent from these version numbers above, and are governed as specified in section E.13 of the part of this document dedicated to the eTIR international system[16].

5. Updating the code lists

118. There are currently 14 code lists managed internally as part the eTIR standard and 11 code lists managed externally by other entities. Some of these external code lists are updated regularly like the ones managed by UN/CEFACT which are revised twice a year. As a result, several code lists may need to be updated more frequently than the eTIR specifications. In order to allow this, a third "update" number is added to the current version number of the eTIR specifications (for instance: 4.3.1, 4.3.2, 4.3.3, etc.) and will be indicated in the metadata information of all messages exchanged between the eTIR stakeholders for them to know which versions of the code lists is used in the message.

119. The updates of the code lists is managed during so-called "update cycles" within the versions of the eTIR specifications. Depending on the length of validity of a version of the eTIR specifications, there can be one or more of these update cycles, given the fact that the first update cycle starts with the beginning of a version of the eTIR specifications, therefore, with a given list of versions for the code lists. The update cycles have to be subsequent with one another, which means that only one version of code lists is valid at a given date and time to avoid issues in interpreting historical values. The following figure shows the relationship between both notions.

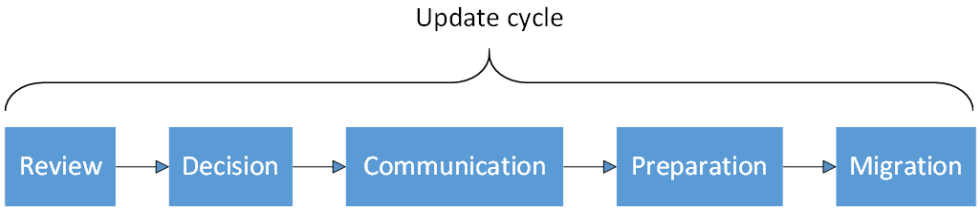
Figure XI
Update cycles in the versions of the eTIR specifications



120. The update cycles consist of a sequence of phases and the length of each of them, as well as the overall length of the cycle, is defined by TIB. The following figure shows the phases involved in an update cycle, which are then described below.

[16] See section II.A.13 of document ECE/TRANS/WP.30/GE.1/2021/33

Figure XII
Phases of an update cycle

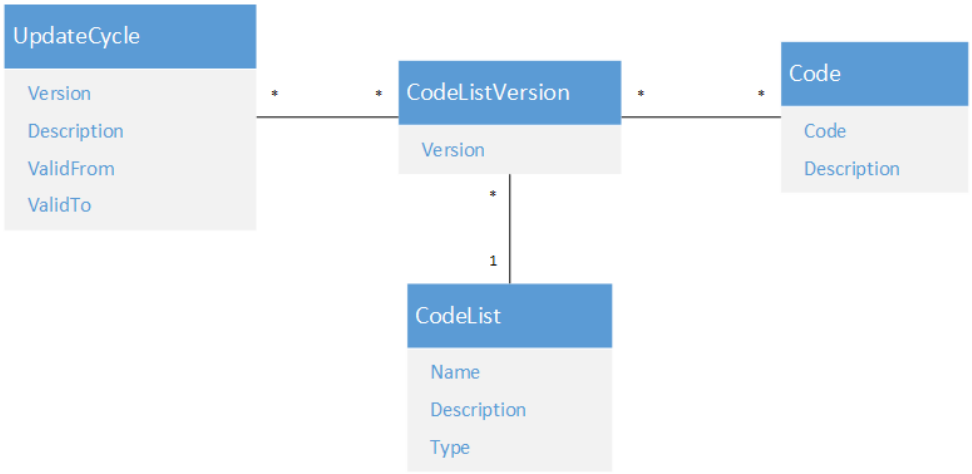


1. **Review**: ECE reviews the latest versions available of the external code lists and prepares a proposal to update one or more of these code lists. ECE can also use this opportunity to update one or more of the internal code lists.
2. **Decision**: ECE presents the proposal to TIB which decides on the list of code lists to update, assigns a new version of the eTIR specifications to it by incrementing its "update" version and decides on the start date and time of the new update cycle.
3. **Communication**: ECE informs all eTIR stakeholders about the decision of TIB related to the new update cycle and related information (new version of eTIR specifications, new versions of the code lists and start date and time).
4. **Preparation**: ECE upgrades the eTIR international system to include the new version of the code lists and configure the system to switch automatically to the new update cycle (using the new version of eTIR specifications and the new versions of the code lists) at the given start date. All eTIR stakeholders are expected to do the same and they can test the new version of their information systems with a dedicated instance of the eTIR international system on the UAT environment.
5. **Migration**: At the given start date and time of the new update cycle, the current update cycle automatically finishes and all eTIR stakeholders (including the eTIR international system) are switching to the new version of the specifications which uses the new version of the code lists.

121. During the migration phase, there are several ways for the eTIR stakeholders to smoothly operate the switch to the new update cycle. One of these options consists in manually deploying a new version of their information system at the start date and time. This new version should have been previously tested in collaboration with ECE during the preparation phase. Deploying a new version of their information system might require taking it off-line for a given period.

122. A second option consists in automating the switch to the new update cycle to avoid the period of unavailability and the manual processes. This can be done by deploying a new version of the information system in advance which includes a test based on the date and time to use the current or the future version of the eTIR specifications and code lists in messages. This can also be done by implementing the notion of update cycles in their database. As an example, the following figure shows how these notions are implemented in the eTIR database.

Figure XIII
Update cycles and code lists class diagram



123. As depicted in the above class diagram, an update cycle is linked to specific versions of code lists. Code lists can be of two types: internal (defined in the eTIR specifications) or external. Most of the time, only a few codes differ from one version of a code list to another. Therefore, the design of the structure should allow to link one code to several versions of code lists to prevent unnecessary duplication of codes.

124. With this approach of managing versions of code lists, users of the eTIR system will get an update of the code lists used by the business community on a regular and predictable basis. The update cycles, code lists and their codes will be managed in the eTIR international system to allow for proper management of code lists over time, including being able to query information related to TIR transports that happened in the past with previous versions of code lists.