Dependability Certification Guidelines for NFVIs through Fault Injection

Domenico Cotroneo, Luigi De Simone, Roberto Natella
Università degli Studi di Napoli Federico II
{cotroneo, luigi.desimone, roberto.natella}@unina.it

***Abstract*—Network Function Virtualization (NFV) is an emerging networking paradigm that offers new ways of creating, deploying, and managing networking services, by turning physical network functions into virtualized one. The NFV paradigm heavily relies on cloud computing and virtualization technologies to provide carrier-grade services. The certification process of NFV systems is an open and critical question to ensure that the delivered network service provides specific guarantees about performance and dependability. In this paper, we propose potential guidelines for evaluating the reliability of NFV Infrastructures (NFVIs), with the aim of verifying whether NFVIs satisfy its reliability and performance requirements even in presence of faults. The guidelines are described as a set of key practices to be followed, in terms of inputs, activities, and outputs. These practices are intended to be conducted by companies that want to evaluate the reliability of their NFVI against quantitative performance, availability, and fault tolerance objectives, and to get precise feedback on how to improve its fault tolerance.**

***Index Terms*—NFV; Virtualization; Dependability; Certification; Fault Injection**

I. Introduction

Network Function Virtualization (NFV) is an emerging solution that is going to supersede traditional network equipment in order to reduce costs, improve manageability, reduce time-to-market, and provide more advanced services [13]. NFV will exploit IT virtualization technologies to turn network equipment into *Virtualized Network Functions* (VNFs) that will be implemented in software, and will run on commodity hardware, virtualization and cloud computing technologies located in high-performance data centers, namely *Network Function Virtualization Infrastructures* (NFVIs).

The NFVIs are subject to the stringent performance and reliability requirements inherited from telecom applications, that are even more demanding than existing IT cloud systems: telecom workloads will require extremely low packet processing overheads, controlled latency, and efficient virtual switching, along with automatic recovery from faults and extremely high availability (99.999% or higher) [16]. Thus, differing from existing telecom networks, the "*softwarization*" process of network functions raises serious reliability concerns. NFVIs should be able to be resilient against *faults* that affect IT cloud computing infrastructures, and that do not affect traditional network equipment.

The characterization and certification of the reliability of cloud computing systems, which include NFV as a whole, are both high-priority issues for telecom operators, service providers and the user community. Indeed, the assessment of dependability becomes more compelling for NFV, as denoted by the interest of standardization bodies to define reliability requirements and evaluation procedures for the cloud [4], [9], [20], for NFV [13], [14], and also the effort to drive the consistent implementation of an open and standard NFV reference platform [15].

Although these efforts, compared to other business- or safety-critical domains, there is still a lack of process certification standards that provide rigorous steps to be followed by NFV vendors/providers for producing evidences that the provided systems meet all requirements mentioned before. Indeed, European standards like the CENELEC EN 50128 [3] for railway, the RCTA DO-178B [17] for avionics, and the ISO 26262 [10] for automotive software systems, are well-known examples of set of standard procedures to be followed in order to guarantee reliability and safety. However, those standards mostly focus on development cycle and functional testing. Instead, the reliability assessment focuses more on evaluating non-functional properties such as the robustness of the system against faults and against stressful workloads.

In this paper, we present a set of potential guidelines for the systematic evaluation of reliability of NFVIs, based on *fault injection*, that is, the deliberate introduction of faults in a system during its execution. The proposed guidelines are based on quantitative performance and reliability goals and indicators, with the aim of enabling NFVI designers to achieve high confidence in the reliability of NFVIs, and providing useful feedback for improving the design of NFVIs. The proposed guidelines were adopted in the context of a Huawei Technologies Co. Ltd. pilot industrial research project. That project aimed at evaluating reliability of NFVIs utilized for Huawei's future telecom cloud network infrastructures.

The paper is organized as follows. Section II provides a background on NFVI reliability, and it defines the key reliability features and attributes. It also introduces the NFVI reliability evaluation process including concepts related to KPIs and fault model. Section III presents the practices and the activities that should be conducted during the evaluation of NFVI Reliability capabilities. In section IV we discuss related works and then conclude the paper.

II. Reliability in NFVIs

The NFV reliability evaluation aims to verify that a NFVI satisfies its reliability requirements in presence of faults. To this aim, *fault injection* is a valuable technique to evaluate the reliability of NFVIs by introducing faults into the system during its execution. In the following, we provide an overview on the entire evaluation process, where fault injection testing is the

This work has been supported by UniNA and Compagnia di San Paolo in the frame of "Programma STAR" (project "FIDASTE").

core technique adopted for assessing the NFV reliability. The process includes the following phases.

1) *Reliability Requirement Definition*: it consists in the definition of quantitative measures for reliability attributes, e.g. how to calculate fault detection, localization, recovery. To this aim, a set of *key performance indicators* (KPIs) are defined (see section III-B for further details).
2) *Fault Modeling*: it consists in the identification of the *faultload*, i.e., a set of faults to inject in the NFVI (see Section III-C for further details).
3) *Fault Injection Test Planning*: it consists in the definition of test cases that will be performed to evaluate NFVI reliability, and in the identification of the workload to be used for exercising the NFVI during the experiments. Each test case provides details about the fault to be injected according to the identified fault model (see III-D for further details.).
4) *Fault Injection Execution*: it consists in the execution of a sequence of fault injection test cases. For each test case, a fault injection experiment is performed: the NFVI under evaluation is first configured, by deploying a set of VNFs to exercise the NFVI; then, the workload is submitted to the VNFs running on the NFVI and, during their execution, a fault is injected; at the end of the execution, performance and failure data are collected from the target NFVI (See III-E for further details).
5) *Fault Injection Data Analysis*: it consists in analyzing performance and failure data produced during the execution of the test cases in order to compute KPIs (See III-F for further details).
6) *Fault Tolerance Improvements*: it consists in identification of reliability bottlenecks in the target NFVI on the basis of the results obtained during data analysis (See III-G for further details).

III. Guidelines and best practices for assessing NFVI reliability capability levels

As discussed in section II, the NFV reliability evaluation consists of six phases. Each phase requires to perform one or more activities in order to achieve a specified objective. The set of activities is referred as *practice* and it leads to one or more *output documents*, which report the results obtained for each activity. Figure 1 shows the six practices that correspond to each phase of the reliability evaluation process by highlighting the input and the expected output. These practices are discussed more in details in the following sections, which in turn provide examples of application of such practices in the context of industrial research project [5], [7].

In turn, Table I provides more details about the objectives (briefly discussed in Section II) and the output that the activities should produce.

A. NFVI reliability capability levels

The reliability evaluation practices are aimed at the assessment of **NFVI reliability capability levels**. The capability level provides insight both on the reliability of a NFVI, and on the quality of the reliability evaluation process itself. A high capability level means that NFVI designers can have a high confidence on the reliability of the NFVI. In particular, a high capability level requires that the key practices are followed, and all the activities are correctly and thoroughly performed, well-documented, and they help in the identification of actions for improving the NFVI fault tolerance.

TABLE I
Objectives and Outputs of NFVI Reliability Practices

Reliability Practice	Objective	Output
Reliability Requirements Definition	Definition of KPIs to measure reliability attributes and identification the of thresholds for the analysis of KPIs	Quantitative Objectives: list of the identified KPIs
Fault Modelling	Identification of the faultload, i.e. the set of faults to be injected	Fault Model: list of the potential faults that can affect the NFVI components
Fault Injection Test Planning	Identification of the test cases that should cover the identified fault model.	Test Case Plan: list of the test cases
Fault Injection Execution	Execution of fault injection experiments driven by the specified test plan	Raw performance/failure data collected during experiments
Fault Injection Test Data Analysis	Analysis of the results in order to calculate the identified KPIs.	Test Report: for each test case, it reports the obtained value of KPIs for performance, availability, and fault tolerance attributes
Fault Tolerance Improvement	Identification of reliability bottlenecks of a NFVI	Plan for NFVI improvements: list of corrective actions to be performed

The definitions of reliability capability levels are:

LEVEL 1: There is a lack of systematic procedures and activities for reliability evaluation. NFVI fault tolerance is evaluated by using ad hoc practices, and fault injection is not systematically conducted. The reliability evaluation is not rigorous and it is based only on the engineers' experience.

LEVEL 2: A reliability requirement analysis is performed, and NFVI fault tolerance mechanisms are identified. Performance and availability objectives are defined only on qualitative basis (e.g., they do not provide precise performance or availability objectives to be achieved, or do not consider all the services deployed on the NFVI). The fault modeling practice is solely based on engineers' experience and/or it does not analyze all components' failures and/or services deployed on the NFVI. As a result, the practice produces an incomplete fault model. Fault injection testing is performed but the experiments are not reproducible, automated, and comprehensive. Test data are just manually analyzed, and/or the causes of test case failures are not diagnosed in detail.

LEVEL 3: A reliability requirement analysis is performed, and NFVI fault tolerance mechanisms are identified. Performance and availability objectives are quantitative, but their values are not well-grounded (e.g. they are based on engineers' experience). The fault modeling is based on a systematic FMEA analysis, and covers all components and services in the NFVI, but is not based on quantitative failure information from deployed NFVIs. Fault injection testing is performed but the experiments are not reproducible, automated and comprehensive. Test data are systematically analyzed, but engineers are not able to perform corrective actions to improve NFVI fault tolerance.

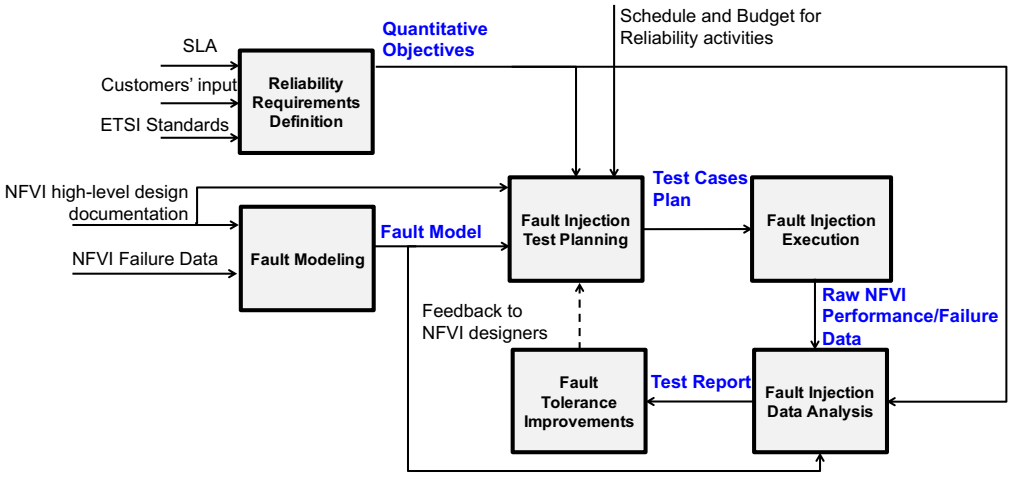

Fig. 1. Practices and items for reliability evaluation of NFVIs.

LEVEL 4: A reliability requirement analysis is performed, and NFVI fault tolerance mechanisms are identified. Performance and availability objectives are quantitatively defined, and their expected values are well-grounded (e.g., based on standards, regulations, and SLAs). The fault modeling practice is based on field data and it produces a fault model that covers all the components and/or services deployed on the NFVI. Fault injection testing is performed and the experiments are reproducible, automated and comprehensive. The results of fault injection testing are analyzed and leveraged to provide feedback on corrective actions that have to be performed in order to improve NFVI fault tolerance.

Table II summarizes how the practices are conducted for each reliability capability level. In the following, we describe more in detail each phase.

B. Reliability Requirements Definition

Despite the complexity introduced by virtualization, it is expected that a NFVI is able to provide services with the same reliability as that assured by traditional telecom systems. To this aim, designers should clearly define reliability requirements for the NFVI, and then perform reliability evaluation activities to assure that these requirements are met.

To evaluate the performance, availability and fault tolerance in presence of faults, it is required to define quantitative objectives in terms of KPIs (*key performance indicators*). The performance/availability/fault tolerance measurements will be compared to these objectives.

The inputs, activities, and outputs from this key practice are provided as follows:

- **Inputs:**
  1) Customer inputs in the form of their requirements and expectations;
  2) Service Level Agreement documents;
  3) Requirements for NFV from applicable standards (e.g., ETSI standards);
  4) Expected environmental and operating conditions of the NFVI (e.g. VNFs to be deployed, VNFs workload);
  5) Reliability requirements of non-virtualized services that are going to be virtualized;
  6) Design documentation about the services that will be deployed on the NFVI;
  7) Design documentation about the technologies to be used for NFVI (e.g., information on virtualization technologies).
- **Activities:**
  1) For each service to be deployed, identification of quantitative objectives of performance (latency and/or throughput) and availability (success response rate) to be fulfilled by the NFVI.
  2) For each service to be deployed, identification of fault tolerance mechanisms (detection, localization, and recovery) and of their quantitative requirements.
- **Outputs:**
  1) A list of quantitative objectives (for both fault tolerance mechanisms, and performance/availability at service level) based on KPIs. Each KPI is described by a name, a definition, a formula, a score, and a measurement method.

***Example**:* In our research project, we define the KPIs showed in Table III. For example, the *latency* of network functions should not exceed a maximum allowed latency goal imposed by service-level agreements. The *throughput* of network functions should not be lower than a minimum allowed throughput goal imposed by service-level agreements. The *experimental availability* of network functions (i.e., the percentage of requests/packets that are successfully processed) should not be lower than a minimum availability goal imposed by service-level agreements. Finally, the *fault tolerance mechanisms* in the NFVI should be able to detect, locate and recover from the fault, with a high probability (high coverage) and within a limited amount of time (low latency). It must be noted that such definition of KPIs is one of the possible choice among other performance and reliability measures that can be used.

C. Fault Modeling

Before starting the fault injection testing, a fault model has to be defined. This practice provides a *fault model* for a NFVI that includes the set of faults (from the physical and virtualization

TABLE II
NFVI RELIABILITY CAPABILITY LEVELS

	Requirements definition	**Fault Modelling**	**Fault Injection Planning and Implementation**	**Data Analysis and NFVI Improvement**
Level 1	Reliability requirements are not available.	Fault modeling is based only on engineers' experience	Fault Injection is not conducted	Fault injection test data are not available
Level 2	Qualitative reliability requirements are defined.	The Fault Model is simply based on anecdotal experience and/or only partially covers NFVI components and services	Fault Injection testing is performed, but tests are not reproducible, automated and comprehensive.	Test data are just manually analyzed, and/or the causes of test case failures are not diagnosed in detail.
Level 3	Quantitative reliability requirements are defined, but they are not well-grounded	The Fault Model is based on a systematic FMEA analysis, but not based on failure data from deployed NFVIs	Fault injection testing is performed, but the tests are not fully reproducible, automated and comprehensive	Fault injection test data are analyzed but corrective actions are not systematically conducted to improve NFVI fault tolerance
Level 4	Quantitative reliability requirements are defined, based on standards, regulations, and SLAs	The Fault Model is based both on a systematic FMEA analysis and on failure data from deployed NFVIs	Fault injection testing is performed, and the tests are reproducible, automated and comprehensive	Fault injection test data are systematically analyzed and provide feedback in terms of corrective actions that have to be performed in order to improve NFVI fault tolerance

TABLE III
KPIS FOR PERFORMANCE, AVAILABILITY, AND FAULT TOLERANCE ATTRIBUTES

KPI	**Definition**	**Formula**	**Evaluation**
Latency	The 50th and 90th percentiles of the empirical cumulative distribution function of the traffic processing time must be below specific latency thresholds	$L(x) = P(latency < x)$	$L(50) < X\_50$ *ms* $L(90) < X\_90$ *ms*
Throughput	The rate of successful requests/packets processed per unit of time is above a given throughput threshold	$\frac{Reqs(t_{\texttt{begin}}, t_{\texttt{end}})}{(t_{\texttt{end}} - t_{\texttt{begin}})}$	$\geq$ `threshold`
Resource Utilization	Resource utilization during a service (CPU, virtual and physical memory) should not hit the maximum	$U_{CPU_{\%}} = \frac{CPUused}{CPUmax}$, $U_{pmem_{\%}} = \frac{pmem\_used}{pmem\_max}$, $U_{vmem_{\%}s} = \frac{vmem\_used}{vmem\_max}$	$\leq 100\%$
Experimental Availability	The percentage of requests/packets that are successfully processed should be higher than a threshold	$EA = \frac{Reqs^{\texttt{success}}}{Reqs^{\texttt{all}}}$	$\geq X\%$
Risk Score	The score represents the risk of incurring in NFVI performance/availability failures when a fault occurs	$RS = \sum_{i=1}^{N} P_i \sum_{j=1}^{M} C_j \frac{F_{i,j}}{E_i}$, where • N is different types of faults injected, • M is different types of observed service failures, • $0 \leq C_j \leq 1$ is the *severity* of the failure type j, • $0 \leq P_i \leq 1$ is the relative importance of the fault type i	$\geq X\%$
Fault Detection Coverage	Percentage of cases in which VNF/NFVI component (VM or host) failures are correctly identified	$FDC = \frac{\#F_{\texttt{fault\_detected}}}{\#F_{\texttt{fault\_undetected}} + \#F_{\texttt{fault\_detected}}}$	$\geq X\%$
Fault Detection Latency	The time between the injection of a fault and the occurrence of the first fault notification	$FDL = t^{detection} - t^{injection}$	$\leq FDL\_thr$ ms
Fault Localization Coverage	Percentage of cases in which a VNF/host that actually failed was correctly identified as failed	$FLC = \frac{\#F_{\texttt{fault\_localized}}}{\#F_{\texttt{fault\_detected}}}$	$\geq X\%$
Fault Localization Latency	The time between the first fault notification and when the system identify which components have failed	$FLL = t^{localized} - t^{detected}$	$\leq FLL\_thr$ ms
Fault Recovery Coverage	Percentage of cases in which a recovery action (triggered by fault detection) is successfully completed.	$FDC = \frac{\#F_{\texttt{fault\_recovered}}}{\#F_{\texttt{fault\_detected}}}$	$\geq X\%$
Fault Recovery Latency	The time between the first fault notification and when the system concludes a recovery action	$FDL = t^{recovered} - t^{detected}$	$\leq FRL\_thr$ ms

layers) to inject, in order to assess their impact on the reliability of the NFVI and of VNFs deployed on the NFVI. If failure data from deployed NFVIs are available, then this knowledge can be adopted to further refine the fault model.

The inputs, activities, and outputs from this key practice are provided as follows:

- **Input:**
  1) Documentation about high-level design of NFVI, including virtual and physical resources.
  2) Documentation about the technologies to be used for NFVI.
  3) Failure data from deployed NFVIs (if available).
- **Activities:**
  1) Identification of components that are part of the NFVI (physical and virtualised).
  2) Identification of services provided by each component.
  3) Identification of faults that can occur for each component and service (based on the experience of engineers and developers, or on a systematic FMEA, or on failure data).
- **Outputs:**
  1) A fault model that encompasses potential faults within a NFVI, along with their expected effects, frequency, criticality, and the emulation method (for fault injection purposes).

***Example**:* In our research project, we defined a generic fault model for NFVIs, through a failure mode effect analysis (FMEA) of the general architecture of NFVIs, and of popular virtualization technologies [7].

In particular, for each domain (physical and virtual CPU, memory, disk, and network), we identify what to inject according to the following three general fault types:

- *Unavailability*, the resource becomes unresponsive and unusable;
- *Delay*, the resource is overcommitted and slowed down;
- *Corruption*, the information stored or processed by the resource is invalid.

We specialize these general fault types for each resource, by analyzing how hardware, software, and/or operator faults can likely cause these three possible fault types [8]. In this analysis, we consider the scientific literature on fault injection and failure analysis in cloud computing infrastructures, well-known cloud computing incidents, and knowledge on the prospective architecture and products for NFVI (e.g., VMware ESXi, Docker containers), to identify a representative, complete, and acceptable set of faults. In [7], we adopted a fault model consisting in 24 fault types.

D. Fault Injection Test Planning

Fault Injection Test Planning consists in the definition of test cases (based on fault injection) that will be performed to evaluate NFVI reliability. Each test case provides details about the fault to be injected in terms of fault type, fault location, fault timing, and other attributes. Moreover, this activity will identify the workload to be used for exercising the NFVI during the experiments.

The inputs, activities, and outputs from this key practice are provided as follows:

- **Input:**
  1) Fault Model.
  2) Documentation about the high-level design of NFVI, including virtual and physical resources.
  3) Schedule and budget constraints for conducting reliability activities, and identification of responsible individuals for the activities.
- **Activities:**
  1) Identification of the workload according to the customer's expectations and to the expected operating conditions.
  2) Creation of a detailed fault injection test plan for NFVI that includes the list of test cases. Each test case is characterized by an id, by details about the fault (a reference ID to the fault model to assure traceability, fault type, fault target, fault timing, fault weight), and by the number repetitions of the injection.
  3) Allocation of available resources (materials, human resources, equipment, budget etc.). If necessary, a subset of test cases can be sampled to comply with schedule and budget constraints.
- **Outputs:**
  1) Fault injection Test Plan containing the list of test cases.

***Example**:* In our research project, according to the defined fault model, we specify a set of fault injection test cases divided in *network-related*, *storage-related*, *CPU-related*, and *memory-related*. Each of these classes is further divided in *PM-level* (physical machine) and *VM-level* (virtual machine) test cases. For example, test cases about network at PM-level evaluate the reliability of NFVI against faults affecting the network layers of the hypervisor. These faults may be induced on each physical machine in the NFVI. Table IV illustrates an example of fault injection test plan in which test cases are related to network faults at both VM- and PM-level in a scenario in which two VNFs (VNF1 and VNF2) are deployed on a NFVI. In particular, in [7], based on the particular NFVI configuration (hypervisor- and container-based virtualization), the fault injection test plan included 180 tests (60 on the physical layer, 120 on the virtual layer), for a total of 360 tests.

E. Fault Injection Execution

Fault injection experiments are executed on the basis of the defined fault injection test plan. A fault injection framework automatically executes the test cases in the plan, by executing the following steps: (1) reset and start the system under test, (2) start the workload, (3) inject a fault at the time and location specified by the test case. The fault injection tool has to collect data (e.g., performance and error logs from the system) for the off-line analysis of results. The inputs, activities, and outputs from this key practice are provided as follows:

- **Input:**
  1) Document of the fault injection test plan.
- **Activities:**
  1) Installation of the NFVI fault injection tool.
  2) Installation of the workload generator tool.
  3) Execution of the experimental campaigns. The activity includes the workload execution, the fault injection and the data collection for each test case defined in the fault injection testing plan.

TABLE IV
EXAMPLES OF TEST CASES FOR NEWORK-RELATED FAULTS AT PM- AND VM-LEVEL

Test Case ID	Fault Type	Fault Target	Fault Timing	Level
1	header/payload corruptions of network traffic frames from/to a virtual network interface	VNF1	sporadic	VM
2	header/payload corruptions of network traffic frames from/to a virtual network interface	VNF1	bursty	VM
3	header/payload corruptions of network traffic frames from/to a physical network interface	Host	sporadic	PM
4	header/payload corruptions of network traffic frames from/to a physical network interface	Host	bursty	PM
5	delays of network traffic frames from/to a virtual network interface	VNF1	sporadic	VM
6	delays of network traffic frames from/to a physical network interface	Host	sporadic	PM
7	drops of network traffic frames from/to a virtual network interface	VNF2	bursty	VM
...	...	...	...	...
N	drops of network traffic frames from/to a physical network interface	Host	bursty	PM

- **Outputs:**
  1) Raw failure data from the execution of each experiment. Data should be collected at the service level (e.g., output from the workload generator about service availability and performance) and from the NFVI (e.g., log files from VMs, from the hypervisor, and from fault tolerance mechanisms, and measurements of virtual and physical resources utilization such as CPU, memory, storage, and network).

***Example**:* In our research project, we developed fault injection technologies for performing the test cases identified during test plan. The fault injection suite include a set of tools, to be installed both on the virtualization layer of an NFVI node, and on its virtual nodes. Furthermore, we perform an experimental analysis on a NFV system running a virtualized IP Multimedia Subsystem (IMS). We deploy the IMS on two NFVIs based on different virtualization technologies: a commercial hypervisor-based virtualization platform (VMware ESXi), and an open- source container-based solution (Linux containers) [7].

F. Fault injection test data analysis

The purpose of the fault injection test data analysis is to analyze the collected data to (i) evaluate KPIs, and to assess whether the reliability requirements were met; (ii) identify test cases that expose a degradation of NFVI performance and availability. Failed test cases have to be diagnosed in detail to determine the root cause of the failure and corrective actions to mitigate a NFVI failure. Such information is helpful at giving recommendations to improve NFVI fault tolerance and to enhance its failure detection and recovery mechanisms. The inputs, activities, and outputs from this key practice are provided as follows:

- **Input:**
  1) Raw NFVI performance and failure data from fault injection experiments.
  2) Documents produced during requirements analysis, fault modelling, and test planning.
- **Activities:**
  1) Analysis of performance KPIs for each test case, to identify which test cases exhibited a performance failure.
  2) Analysis of availability KPIs for each test case, to identify which test cases exhibited an availability failure.
  3) Analysis of data on resource utilization, to identify which test cases lead to an overload of NFVI resources.
  4) Analysis of data on fault tolerance mechanisms, to identify test cases where faults were not detected, located, and/or recovered.
- **Outputs:**
  1) Test Report that records, for each test case, the occurrence of any performance/availability failure, the occurrence of resource overload, and the coverage of fault detection, and recovery mechanisms.

***Example**:* In our research project, we analyzed different sources of data, i.e., the log files from the hypervisor, output from the workload generator, measures about virtual and physical resources utilization of the such as CPU, memory, disk and network devices. From that analysis, we computed the defined KPIs in order to obtain useful feedbacks, for example about which type of fault impact mostly on the performance rather than fault tolerance mechanisms of VNFs and of the NFVI.

G. Fault Tolerance improvement

The Fault Tolerance improvement practice consists in introducing corrective changes in the NFVI, based on the results obtained from fault injection testing. It involves the definition (on the basis of a detailed diagnosis of failed test cases), implementation, and evaluation of corrective actions (on the basis of the re-execution of test cases that were previously failed), and preventing the occurrence of identified vulnerabilities in future products.

- **Input:**
  1) Test report with results from fault injection testing.
- **Activities:**
  1) Diagnosis of the experiments where faults affected performance and/or service reliability.
  2) Identification and implementation of corrective actions to improve NFVI fault tolerance.
  3) Evaluation of the effectiveness of corrective actions at improving reliability, by executing again fault injection experiments.
- **Outputs:**
  1) An improvement plan with documented corrective actions
  2) Implementation of the improvement plan on the NFVI
  3) Evaluation of corrective actions through reviews and through an updated fault injection test report, to be obtained by re-executing fault injection experiments.

Example: In our research project, we performed fault injection experiments on an IMS system based on an NFVI, by using as fault tolerance mechanism active replicas. For example, what we have found is that it is not sufficient to simply provide high-availability, replicated architectures, since the occurrence of faults quickly consumes redundant resources (e.g., active replica and hosts) and reduces performance and reliability. Thus, a potential improvement is adopting more complex fault tolerance strategies, by actively allocating more resources (e.g., using on-demand cloud computing resource) in the case of faults or adverse conditions, and/or by reconfiguring and recovering the failed resources. For more details see [7].

H. Summary

As previously discussed, the NFVI Reliability evaluation consists in six key practices. Each practice requires to perform a set of activities, as listed in Table V. In details, the table includes the following columns:

- **Activities**: the column is divided into three sub-columns that provide a textual description of the activity *Description*, the practice during which the activity is performed *Practice*, and the reference of the section of this paper that provides more details about the activity *Ref.*.
- **Levels**: the reliability capability level (from 1 to 4) at which the activity is performed.
- **Output**: it is divided into two sub-columns (*Description* and *Ref.*) that provide a description of the output document that must be produced at the end of the activity, the reference of the paper that describes how the produce the output document;
- **Notes**: It may contain additional information about the activity.

The table V specifies the activities that must be performed to achieve each capability level. For each activity, the table indicates how each capability level (from 1 to 4) is achieved: in general, the highest capability is achieved when every activity is performed in full coverage (for instance, the fault model and the test plan should be thorough and should cover all components in the NFVI).

IV. Related work

In the context of certification and reliability assessment of NFVI we need to refer on the ability of an NFVI at tolerating faults. Several fault-tolerance solutions can be leveraged for providing insights to NFVI providers on the fault-tolerance of their infrastructure. In general, fault tolerance mechanisms are adopted in a number of business- and safety-critical domains, including the telecommunication, automotive, avionics, and space domains. Even if there is no individual standard document that specifies how fault tolerance should be designed and evaluated, there are several books and articles that describe how fault tolerance mechanisms have been designed and evaluated in many application domains [1], [2], [11], [19].

Several international standards for software reliability and safety encompass the injection of faults in a system in order to assess its behavior and to measure the efficiency (coverage, latency, etc.) of fault tolerance mechanisms. Among the most significant we can cite:

- the **ISO 26262** [10] standard for automotive safety which prescribes the use of *error detection and handling mechanisms* in software, and their verification through fault injection, by *"corrupting hardware or software components"*;
- the **NASA standard 8719.13B** [12] for software safety recommends fault injection to assess system behavior with respect to *faulty off-the-shelf software components*;
- the **DO-178B** and **DO-178C** [17] requirements for avionic safety recommend that *"robustness test cases should demonstrate the ability of the software to respond to abnormal inputs and conditions"*;
- the **ISO/IEC Systems and software Quality Requirements and Evaluation** (SQuaRE) [18] standard define an evaluation module, the **ISO/IEC 25045**, dealing with the assessment of the *recoverability* of software systems in the presence of accidental faults. The evaluation framework established by *dependability benchmarks* is today partially integrated in this standard. Recoverability is defined as the *ability of a product to recover affected data and re-establish the state of the system in the event of a failure*.

It is important to note that safety standards do not impose any specific measure or procedure for evaluating fault tolerance, because they are not meant for a specific system. Instead, they are meant to provide general guidelines that apply to a broad family of systems. Therefore, these general guidelines need to be tailored for the specific system.

Focusing more directly on NFV, the document **ETSI GS NFV-REL** [14] identifies use cases, requirements and architectures that will serve as a reference for the emerging NFV technologies, including resiliency requirements that the emerging NFV architectures will have to meet. In our previous study, we briefly summarize them [6]. The ETSI addresses the problem of NFV resiliency by recommending design practices, including failure detection and isolation, automated recovery from failures, prevention of single points of failure in the architecture, and so on. Unfortunately, that document mostly proposes generic solution and practice, and does not include any references for the certification process of NFV systems.

V. Conclusion

Providing a dependable network service based on NFV paradigm is a difficult and complex task, as it involves several critical decisions about design and configuration of fault tolerance mechanisms, as well as selecting COTS virtualization and management technologies. Despite its recent introduction, NFV has already gained significant market traction, with hundreds of products from as many competing vendors. Since network services are subject to carrier-grade requirements, we need precise tools for verifying and assuring their reliability.

In this work, we have presented a set of guidelines for paving the way of process certification for evaluating dependability of NFV infrastructures. Like other critical domains, the purpose of this work has been to provide rigorous steps and best practices to be performed in order to increase trustworthiness in delivered NFV-based services.

TABLE V
ACTIVITIES AND OUTPUT OF THE NFV RELIABILITY EVALUATION

Activities				Levels				Output	Notes
	Description	**Practice**	**Ref.**	**1**	**2**	**3**	**4**		
1	Identification of quantitative requirements about performance (latency, throughput) and availability (success response rate) for each service	Reliability Requirements Definition	III-B	No	Partial	Partial	Full	A list of quantitative performance and availability objectives	*Partial*: only a subset of services and metrics are covered. *Full*: all services and metrics are covered
2	Identification of quantitative requirements about fault tolerance mechanisms for each service	Reliability Requirements Definition	III-B	No	Partial	Partial	Full	A list of quantitative fault tolerance objectives	*Partial*: only a subset of services and metrics are covered. *Full*: all services and metrics are covered
3	Fault modeling of each virtual and physical component in the NFVI	Fault Modeling	III-C	No	Partial	Partial	Full	Fault Model	*Partial*: only a subset of components are covered, without using failure data. *Full*: all components are covered, using failure data
4	Identification of the workload according to the customer's expectations and to the expected operating conditions.	Fault Injection Test Planning	III-D	No	Yes	Yes	Yes	Fault injection Test Plan	
5	Identification of available resources (human resources, equipment, budget, etc.) for reliability evaluation	Fault Injection Test Planning	III-D	No	Yes	Yes	Yes	Fault injection Test Plan	
6	Creation of detailed fault injection test plan for NFVI that includes the list of test cases	Fault Injection Test Planning	III-D	No	Partial	Partial	Full	Fault injection Test Plan	*Partial*: only a subset of components are covered, without using failure data. *Full*: all components are covered, using failure data
7	Installation of the NFVI fault injection tool	Fault Injection Execution	III-E	No	No	Yes	Yes	Raw failure data from the execution of each experiment.	
8	Installation of the workload generator tool	Fault Injection Execution	III-E	No	No	Yes	Yes	Raw failure data from the execution of each experiment.	
9	Execution of the experimental campaigns. The activity includes the workload execution, the fault injection and the data collection for each test case defined in the fault injection testing plan.	Fault Injection Execution	III-E	No	Manual	Automated	Automated	Raw failure data from the execution of each experiment.	
10	Computation of performance KPIs (latency, throughput, resource utilization) during fault injection experiments.	Fault injection test data analysis	III-F	No	No	Yes	Yes	Test Report	
11	Analysis of availability KPIs (experimental availability, risk score) during fault injection experiments.	Fault injection test data analysis	III-F	No	No	Yes	Yes	Test Report	
12	Analysis of fault tolerance KPIs (Fault Detection, Fault Location and Fault Recovery) during fault injection experiments.	Fault injection test data analysis	III-F	No	No	Yes	Yes	Test Report	
13	Diagnosis of the experiments where faults impaired performance, availability, and/or fault tolerance.	Fault Tolerance improvement	III-G	No	No	Yes	Yes	Improvement Plan	
14	Identification and implementation of corrective actions that will improve NFVI fault tolerance	Fault Tolerance improvement	III-G	No	No	No	Yes	Improvement Plan	
15	Evaluation of the effectiveness of the corrective actions through re-execution of fault injection experiments.	Fault Tolerance improvement	III-G	No	No	No	Yes	Improvement Plan	

REFERENCES

[1] J. Arlat, M. Aguera, L. Amat, Y. Crouzet, J.C. Fabre, J.C. Laprie, E. Martins, and D. Powell. Fault Injection for Dependability Validation: A Methodology and Some Applications. *IEEE TSE*, 16(2), 1990.

[2] A.T. Bouloutas, S. Calo, and A. Finkel. Alarm correlation and fault identification in communication networks. *IEEE Communication*, 42(234):523–533, 1994.

[3] EN50128 CENELEC. 50128. *Railway applications-Communication, Signaling and Processing Systems-Software for Railway Control and Protection Systems*, 2011.

[4] Cloud Watch HUB. Cloud certification guidelines and recommendations.

[5] D. Cotroneo, L. De Simone, AK Iannillo, A. Lanzaro, and R. Natella. Dependability evaluation and benchmarking of network function virtualization infrastructures. In *Proc. NetSoft*, pages 1–9. IEEE, 2015.

[6] D. Cotroneo, L. De Simone, AK Iannillo, A. Lanzaro, R. Natella, Jiang Fan, and Wang Ping. Network function virtualization: Challenges and directions for reliability assurance. In *Proc. ISSRE*, pages 37–42. IEEE, 2014.

[7] D. Cotroneo, L. De Simone, and R. Natella. NFV-Bench: a Dependability Benchmark for Network Function Virtualization Systems. *IEEE TNSM*, 14(4):934–948, 2017.

[8] DBench project. *DBench Final Report*. http://www.laas.fr/DBench/, 2004.

[9] European Union Agency for Network and Information Security. Cloud computing certification.

[10] ISO. Product development: software level. *ISO 26262: Road vehicles – Functional safety*, 6, 2011.

[11] Kaustubh R Joshi, Matti A Hiltunen, William H Sanders, and Richard D Schlichting. Probabilistic model-driven recovery in distributed systems. *IEEE TDSC*, 8(6):913–928, 2011.

[12] NASA. NASA Software Safety Guidebook. *NASA-GB-8719.13*, 2004.

[13] NFV ISG. Network Functions Virtualisation (NFV) - Network Operator Perspectives on Industry Progress. White Paper, 2013.

[14] NFV ISG. Network Function Virtualisation (NFV) - Resiliency Requirements. Technical report, ETSI, 2014.

[15] Christofer Price and Sandra Rivera. Opnfv: An open platform to accelerate nfv. *White Paper. A Linux Foundation Collaborative Project*, 2012.

[16] Quality Excellence for Suppliers of Telecommunications Forum (QuEST Forum). TL 9000 Quality Management System Measurements Handbook 4.5. Technical report, 2010.

[17] RTCA. DO-178B Software Considerations in Airborne Systems and Equipment Certification. *Requirements and Technical Concepts for Aviation*, 1992.

[18] RTCA DO RTcA. ISO/IEC 25045. *Systems and Software Engineering - Systems and Software Quality Requirements and Evaluation (SQuaRE) - Evaluation module for recoverability*, 2010.

[19] M. Serafini, A. Bondavalli, and N. Suri. On-line diagnosis and recovery: On the choice and impact of tuning parameters. *IEEE TDSC*, 4(4):295–312, 2007.

[20] A. Sunyaev and S. Schneider. Cloud services certification. *Communications of the ACM*, 56(2):33–36, 2013.