TESTBED FOR FUTURE INTERNET SERVICES

Deliverable Title	**D4.4 – Test toolkit implementation**
Deliverable Lead:	SQS
Related Work package:	WP 4
Author(s):	Franca Perrina, Jose Roberto, Jérémie Leguay, Gabriele Giammatteo, Annika Sällström, Urko Acosta, Jonathan González, Igor Fernández, Ainhoa Gracia
Dissemination level:	Public
Due submission date:	04/30/2011
Actual submission:	06/01/2011
Project Number	258142
Instrument:	IP
Start date of Project:	01/06/2010
Duration:	30 months
Project coordinator:	THALES

Abstract	This document covers the collection of statistics from each testbed. Further, the document specifies the mechanism to easily define Key Performance Indicators deriving from such statistical collection.

Project funded by the European Commission under the 7th European Framework Programme for RTD - ICT theme of the Cooperation Programme.

License

Project co-funded by the European Commission within the Seventh Framework Programme (2008-2013)

Versioning and Contribution History

Version	Date	Modification reason	Modified by
0.1	04/02/2011	Initial document	Franca Perrina, Jose Roberto, Urko Acosta, Jonathan González, Igor Fernández, Ainhoa Gracia
0.2	06/01/2011	Review	Ainhoa Gracia
0.3	09/09/2011	Conclusions	Jonathan González, Ainhoa Gracia

TABLE OF CONTENT

INDEX OF FIGURES

INDEX OF TABLES

Abbreviations

CPU	Central Process Unit
ICMP	Internet Control Message Protocol
IMS	IP Multimedia Subsytem
iRODS	Integrated Rule-Oriented Data System
KPI	Key Performance Indicators
MTBF	Mean Time Between Failures
QoS	Quality of Service
RED	Random Early Detection
WRED	Weighted Random Early Detection

EXECUTIVE SUMMARY

This document presents TEFIS platform monitoring overview. Initially, it is involved a brief introduction about monitoring, control and the importance of using appropriate Key Performance Indicators. Then, this deliverable describes the TEFIS monitoring architecture and TEFIS KPI definitions.

Moreover, a preliminary version of proposed overall KPIs for TEFIS platform is integrated. These ones will monitor the Quality of Service of the platform.

Finally, this document illustrates the specific KPIs for each TEFIS testbed. These KPIs will offer statistics about each testbed operation mode.

1. KPI monitoring system

The main goal of monitoring systems is efficient evaluation. Monitoring provides the dynamical analysis of system functionality and health. So, these systems are responsible of information collection, analysis and process. In this way, system monitoring is based on Key Performance Indicators (KPI). The KPIs are a set of metrics focused on measuring aspects that describe the system performance and shape. They must be a way to accurately define and measure them. It is also important KPI definition and stay with the same definition from year to year.

Figure 1 shows main reasons for measuring TEFIS platform performance using KPIs. Learn and improve performance is the most important. The goal is make better decisions that lead to TEFIS platform improvement. In this context, KPIs are used for continuous learning and improvement.

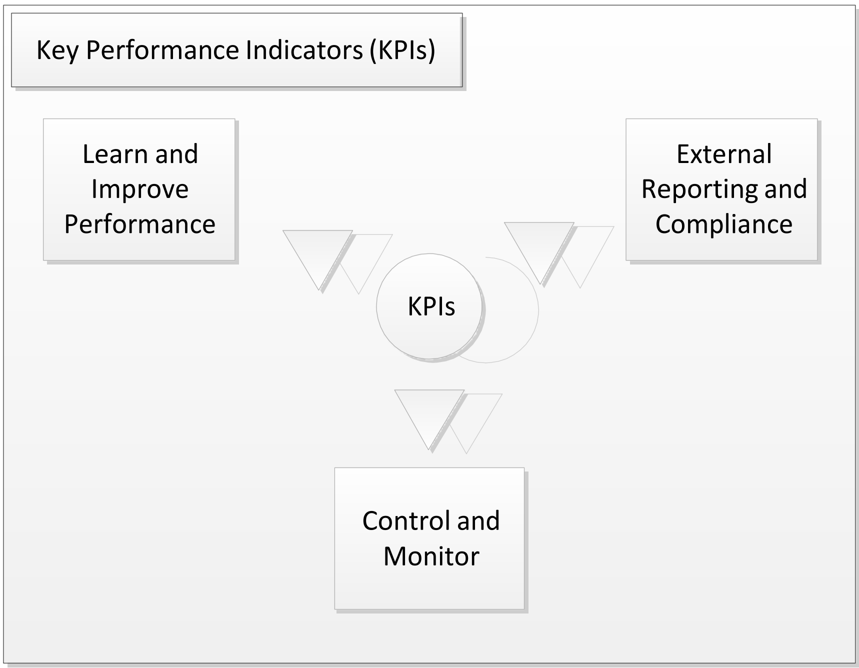

Figure 1: *Key Performance Indicators*

These KPI shall provide the TEFIS platform itself and its users, Testbed providers and experimenters, the measurement progress toward the defined goals. Figure 2 shows overall measuring performance view.

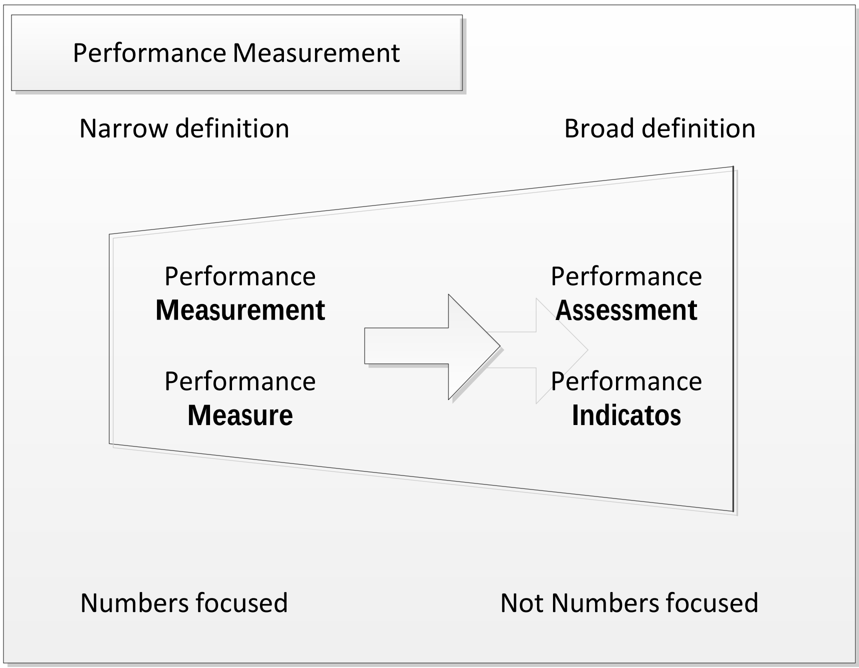

Figure 2: *Measuring Performance*

A set of statistical data, metrics, and quantifiable measurements should be collected during the execution of the experiments and during the interaction between the users and the TEFIS platform and between the components with the others.

In this context, Key Performance Indicators will be quantifiable measurements that shall reflect the critical success factors. They will differ depending on the specific characteristics of each Testbed, their needs and the information that is interesting collect to be presented to the user, and the interfaces defined to establish the communication between the components, the core services and the connectors or the TEFIS middleware and the Portal.

The Testbeds may have as one of its Key Performance Indicators the percentage of satisfaction that comes from return experimenters. A Testbed that includes a service of functional testing may focus its Key Performance Indicators on Pass/Not Pass rates of test cases execution. A Testbed that includes a team of people, who assess the features of a specific application, may have as one of its Key Performance Indicators, the percentage of feasible improvements identified during a session. A Key Performance Indicator for a Testbed that provides hardware might be number of devices repaired during the year.

On the other hand, useful Key Performance Indicators for the TEFIS platform itself could be the percentage of failure events or inactivity hours.

To explain this concept in more detail, whatever Key Performance Indicators are selected, they must reflect the TEFIS and users goals, they must be the key to its success, and they must be quantifiable, it is to say, measurable. In addition to this, the Key Performance Indicators usually are long-term considerations because the definition of what they are and how they are measured do not change often. The goals for a particular Key Performance Indicator only may change when the testbed's goals or the TEFIS platform's goals change, or as it gets closer to achieving a specific goal.

A Testbed that has defined as one of its goals "to provide the answer to the customer in the best times' will have Key Performance Indicators that measure the hours of effort involved in the development of the service and related productivity measures. However, "percent of delays of answers from the users" will not be one of its Key Performance Indicators, considering the efficiency of the team does not depend on the time the costumer needs to reply.

Besides all these issues, Key Performance Indicators must be quantifiable because, if a Key Performance Indicator is going to be of any value, there must be a way to accurately define and measure it: "attract more repeat users" is useless as a KPI for the TEFIS platform without some way to distinguish between new and repeat users.

It is also important to define the Key Performance Indicators and stay with the same definition from year to year. For a KPI of "increase experiments", you need to address considerations like whether to measure by number of executions of each experiment or by value of SLAs signed.

To sum up, these KPIs allow the following actions:

- Follow results of implemented corrective actions.
- Malfunction identification and correction.
- Have an overall and accurate vision about available resources and their using rates.

The following list describes features the KPIs should have:

- Simple and clear
- Easy and quick to make
- Relevant
- Adapted to each function

KPIs give the users and the administrators of the TEFIS platform a clear picture of what is important, of what they need to make happen, using them to manage performance. In this way, it is assured that everything done is focused on meeting or exceeding those Key Performance Indicators.

2. TEFIS monitoring architecture

TEFIS monitoring system will be based on monitoring probes hosted into TEFIS connectors. Each hosted probe will collect and packet resource monitoring information for each testrun execution. Processed data will be forwarded and saved into Experiments Data Manager.

Monitored information will be based on Key Performance Indicators (KPI) for each testbed resource. This KPI information will be described into Teagle repository. So each testbed will have defined into Teagle repository its own monitoring resources. And each resource will indicate resource type and KPI parameter list to be collected.

Figure 3 shows overall monitoring architecture. The monitoring interface will make available monitoring information for TEFIS users. There will be monitoring data available for each testrun execution.

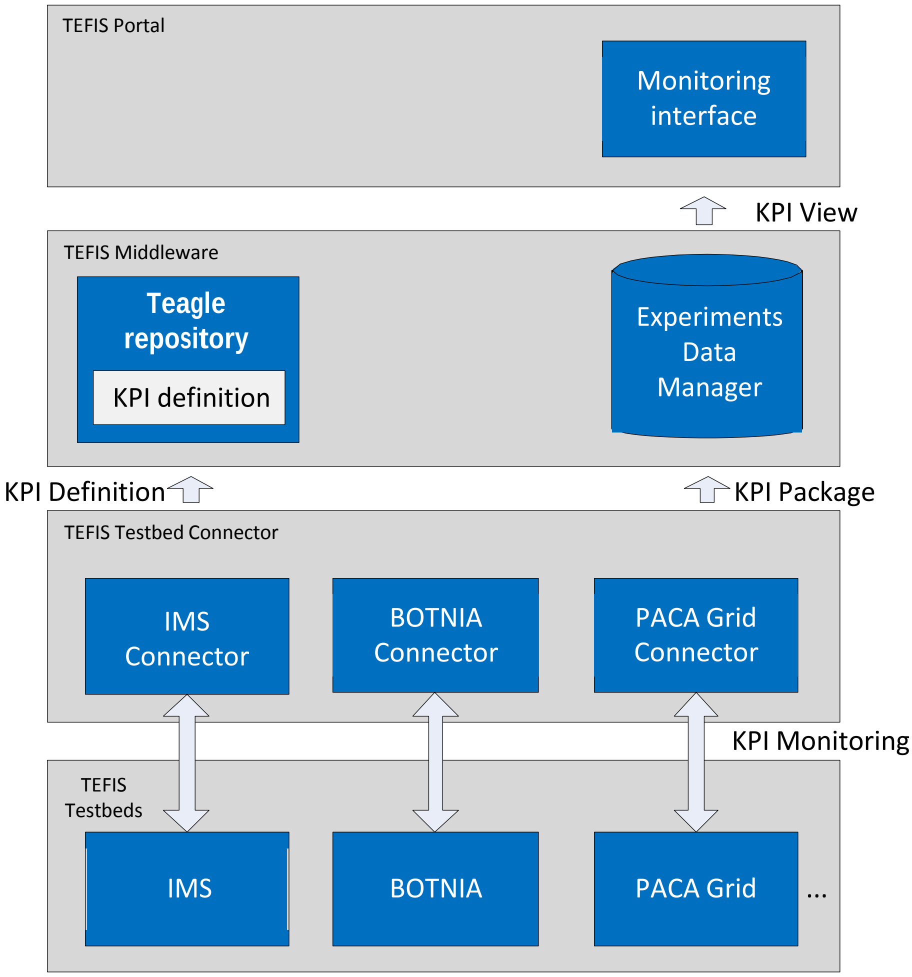

Figure 3: *Monitoring architecture*

3. Overall KPIs

For this first version of TEFIS platform some general Key Performance Indicators have been defined. Taking into account the recent deployment of the platform, this list must be considered as a preliminary version, created in order to initiate a statistic collection that could indicate more precise and necessary KPIs.

Considering that each module will monitor and will give support to its own indicators, TEFIS KPIs are focused mainly in two points. First, they will monitor use statistics, in order to ensure the correct scaling of the platform, considering both hardware and software resources. On the other hand, those KPIs will also provide information about the platform availability and failures.

These KPIs will monitor Quality of Service monitoring using polling techniques. Each TEFIS connector will send Internet Control Message Protocol (ICMP) *echo ping request* packets to each TEFIS testbed. The responses received by each connector will allow total QoS status monitoring during each testrun execution. So, each TEFIS block will have its own QoS defined by the following KPIs

- **Availability**

This KPI will measure the time percentage during which a TEFIS testbed is fully functional. This will involve TEFIS platform service availability. This measure will take place periodically during testrun execution.

Availability %	Downtime per year	Downtime per month*	Downtime per week
90% ("one nine")	36.5 days	72 hours	16.8 hours
95%	18.25 days	36 hours	8.4 hours
98%	7.30 days	14.4 hours	3.36 hours
99% ("two nines")	3.65 days	7.20 hours	1.68 hours
99.5%	1.83 days	3.60 hours	50.4 minutes
99.8%	17.52 hours	86.23 minutes	20.16 minutes
99.9% ("three nines")	8.76 hours	43.2 minutes	10.1 minutes
99.95%	4.38 hours	21.56 minutes	5.04 minutes
99.99% ("four nines")	52.56 minutes	4.32 minutes	1.01 minutes
99.999% ("five nines")	5.26 minutes	25.9 seconds	6.05 seconds
99.9999% ("six nines")	31.5 seconds	2.59 seconds	0.605 seconds

Figure 4: *Availability ranges*

- **Bandwidth**

  It is necessary to monitor the bandwidth between TEFIS blocks and TEFIS testbed. So, network communication between each TEFIS block (Teagle repository, Experiments Data Manager and TEFIS core services) and a TEFIS testbed is monitored using packet sending techniques.

- **Total of users**

  The most obvious statistic to start with is the one which will provide information about the number of users who have an account and use TEFIS platform.

- **Access rate**

  It is also interesting to monitor the number of users who access to TEFIS in a certain period of time, in order to preview future load peaks and prevent them.

- **Maximum number of simultaneous users**

  Along with the previous KPI, this is a key statistic for being able to scale TEFIS platform correctly and to avoid future overloads.

- **Access rate (per user)**

  By collecting statistics about the number of times each user accesses to TEFIS platform during a period of time, and comparing it to the number of jobs they have created, it will be possible to know the usage each user does of the platform. For example, many accesses for a single job would mean that the user defines that job in several times instead of doing it in one session. That knowledge will be useful to develop the most user-focused functionalities in each module of the platform.

- **Total of submitted jobs**

  It will be important to know which the effective use of the platform is by indicating the total number of jobs submitted to execution. The real and final use of the resources in TEFIS will be closely related to the jobs, as each one will imply, for example, new storage necessities in IRODS, new information to be saved in Teagle, or more processing capacity in the Core Services. So, again, this KPI will help in scaling TEFIS platform.

- **Number of jobs per user**

  This KPI will be vital to know what use each user does of TEFIS, and its evolution will also indicate the success of the platform. Initially, each user may only request a single job or experiment execution, but if the experience is positive, then it is probable that the same user will come back in the future.

- **Number of Tesbeds used by each user**

  The main objective of TEFIS platform is to integrate in a unique site many experiment testbeds and to provide the user a single access point to use them. This KPI will be of most importance to check whether this goal is achieved. A small number of testbeds used by a user would indicate either that TEFIS is not able to provide the necessary functionality for complex experiments, or that TEFIS philosophy or capabilities are not well understood by the users.

- **Inactivity hours**

  One of the most important goals of any technological platform is to reduce inactivity periods to the minimum, as each minute that platform is working means a huge amount of money. The same can be applied to TEFIS, but considering the particularities of it. It is obvious that the users' activity may not be continuous. In fact, it should not be much extended, because another TEFIS objective is to provide the most simple and easy experiment configuration. But if we consider the time the experiments are being executed, this will be the main time occupation. So, it will be vital to reduce the time when no experiment is being processed.

- **Clients' complaints**

  The most precise indicator of users' satisfaction about TEFIS will be their own feedback and suggestions, which should be seriously taken into account in order to provide the best possible service and to attracts the higher number of users.

Apart from above KPIs, some other technical indicators could be applied in order to detect possible failures as soon as possible, and to maximize the availability of the platform. Among others, the following ones could be pointed:

- **Percentage of failure events**

  This KPI is based on errors detected in TEFIS platform. These errors will be monitored and organized by type for the creation of statistics. In this way, it is monitored the percentage of failure of different events of TEFIS platform.

- **Platform availability**

  TEFIS platform availability is based on each TEFIS blocks availabilities. So, overall availability is defined by the monitoring of the sum of the availability of all TEFIS blocks.

- **Mean time between failures**

  MTBF monitors the predicted elapsed time between inherent failures of TEFIS platform. It's calculated as the arithmetic mean time between failures. In this case, failure is considered a lack of availability in a TEFIS block.

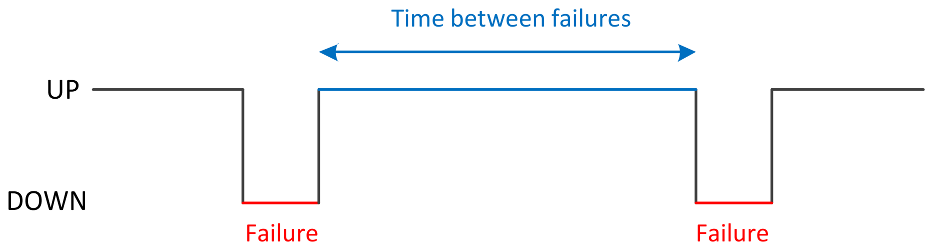

Figure 5: *Time between failures*

Those KPIs are not only applicable to the whole TEFIS platform, but also to particular and critical parts of it. For example, a correct communication between all the blocks is vital to ensure TEFIS reliability and availability. Therefore, it will be of great interest to monitor all the connections between different modules. Here is a list of the most important connections to be monitored.

- Portal – Experiment Manager
- Teagle – Testbeds
- Teagle – Core Services
- Experiment Manager – Teagle
- Experiment Manager – Core Services
- Experiment Manager – IRODS
- Testbeds – IRODS
- Testbeds – Core Services

It could also be added extra information about those connections, such as:

- **Percentage of bandwidth used**

  This KPI will be useful to scale the interface between the blocks properly. This monitoring will help bottleneck identification at network communication level.

- **Percentage of lost frames or requests**
  This indicator will suggest link failures between connected blocks. A hundred per cent of lost requests will suggest that a block is down or out of service.

4. Overall testbed KPIs

4.1 PlanetLab Testbed

Regarding PlanetLab, the pecific statistics, indicators and metrics that an experimenter could look for is the following:

- **Network path characteristics**
  - Node and paths characteristics (type of routers, load balancers, ...)
  - Measurements (traceroute, delay, available bandwidth);
  - AS-level information (prefixes, ASN and names, type of AS, ...);
  - Geolocalization information (latitude/longitude, cities, countries, continents, ...)

- **Flow characteristics**
  - Number of packets in the flow.
  - Duration of the flow
  - Average bandwidth
  - Destination in/out of PlanetLab

- **Machine characteristics**
  - CPU usage
  - Up times
  - Number of current users and slices
  - Aggregated network traffic (in/out packets)

- **Slice characteristics**
  - Number and identifier of nodes
  - Uptimes of the slice
  - Usage statistics (cpu load, network load)

4.2 PACA Grid Testbed

The PACAGrid testbed could be able to collect several types of statistics and measures that can be divided into two main categories.

The first category includes statistics and measures related to the testbed as a whole and the usage of the testbed done via TEFIS. The second set includes monitoring data related to specific experiments.

The first kind of data could give global information about the testbed status and about its usage. It has to be remarked that the testbed is accessed by TEFIS users via the TEFIS platform, but it is also used by other users that access it directly. It means that information related to the whole testbed or the usage of the platform could refer all the PACAGrid users, except if the PACAGrid TEFIS connector implements a specific filtering mechanism. Actually the following metrics could be collected (the term ‘Nodes’ refers to a computing resource in PACAGrid):

Parameter name	Type	Description
PACAGrid specific Resource Manager status	String	The status of the manager of the PACAGrid resources.
Available/Free/Busy/Down nodes	integer	Number of Available/Free/Busy/Down computing resources
Average activity/inactivity percentage	double	The percentage of the activity/inactivity of the resources from the start-up time of the system
The used node time (user)	Time	The time of usage of PACAGrid computing by a specific user
The used node time (all users)	Time	The time of usage of PACAGrid computing by all the users

Table 1: *Metrics related to the whole testbed*

The second set of data is related to specific experiments and it could be collected during experiments monitoring. When a TEFIS user involves PACAGrid in his experiment, it means that a PACAGrid job is submitted to PACAGrid as part of the user experiment. At that time, several data related to the execution of the PACAGrid job could be collected, such as the whole job duration or the duration of each step. More fine grained measures might be provided by the TEFIS connector, even if PACAGrid testbed does not currently provides this kind of data to all its users. Such kind of data would be related to physical resources consumption in the steps, such as CPU usage and Memory usage. Few examples are reported below.

Parameter name	Type	Description
Job/task execution duration	Time	The duration of the execution whole PACAGrid job or of one of its task
Job/task start/finished time	Date	The starting time/finished time of the whole PACAGrid job or of one of its task
CPU usage (Min/Max/Average)	integer	Minimum/Maximum/Average value of CPU usage (percentage) by the process or the group of processes related to a given step

Memory usage (Min/Max/Average)	integer	Minimum/Maximum/Average value (in Mega Bytes) of Memory usage by the process or the group of processes related to a given step

Table 2: *Metrics related to experiments*

4.3 ETICS Testbed

ETICS System provides the experimenter with the possibility of running builds and tests (generally called jobs) of their software's source code in a machine of its infrastructure.

The flexibility of the ETICS System in letting the experimenter to choose the checks to execute on the source code and the type of tests to run (with the possibility also to run custom test) make it difficult to identify a set of standard KPI valid and meaningful for any job. Nevertheless, it is still possible to identify few KPIs completely indipendent from the type of job executed on the system.

The following table describe KPIs that will be made available by the ETICS Connector on the TEFIS ' Experiment data manager at the end of the job execution.

KPI	Type	Description
build time	Time	The time the build job last on the ETICS system
number of components	integer	the overall number of components built during the job
build success rate	double	The percentage of components built succesfully during the build

In the future, in new releases of the ETICS System, some other KPIs such as CPU load, memory usage and disk usage may be added to the ones currently available.

4.4 IMS Testbed

The IMS is based on resources existing 9 resource instances available for TEFIS platform experiments. Each resource KPIs are defined based on each resource type properties. These KPIs describe necessary resource performance. In this way, there are defined 4 different oriented KPIs:

1. Server oriented IMS-CORE KPIs and service oriented Q-Mobile KPIs.

- **IMS-CORE**

  The IMS-CORE monitoring will be based on network interface and CPU monitoring. This point of view involves the following KPIs:

- **Availability:** Time percentage during which a TEFIS IMS-CORE is fully functional. This will involve service availability. This measure should take place periodically during testrun execution.
- **Bandwidth using:** Percentage of bandwidth consumed by IMS-CORE network interface resource.
- **Available storage memory**: Amount of free memory in IMS-CORE.

- **Q-Mobile**

  Q-Mobile service oriented resource makes mandatory available tester KPI inclusion. This service is based on these testers and available mobile devices which provide physical support.

  - **Available tester**: Amount of free testing engineer who develop Q-Mobile service tasks.
  - **Available mobile devices**: Amount of free mobile devices. They provide Q-Mobile service necessary hardware support.
  - **Bandwidth:** Percentage of bandwidth consumed by Q-Mobile network interface device.
  - **Available storage memory**: Amount of free memory in Q-Mobile storage device.

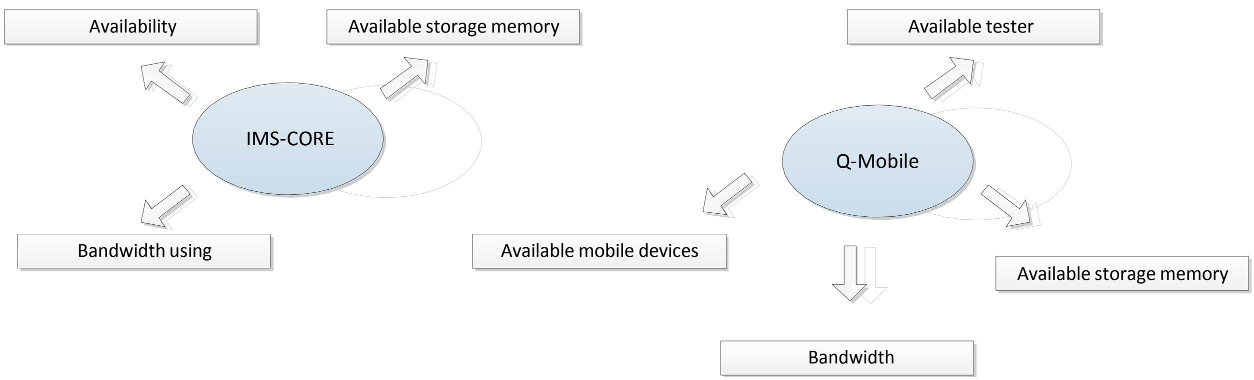

Figure 6: *IMS-CORE and Q-Mobile resources and their KPIs*

2. Test procedure oriented testplan KPIs.

- **Basic Testplan**

  Each testplan is based on a simple template of test procedures that is described by the following KPIs:

- **Development cost**: Amount of time wasted in testplan template adaptation to each testing service.
- **Availability**: Availability of testplan template into IMS testbed.

- **Customized Testplan**

  Each customized testplan is based on template of test procedures partially defined and configurable by the client.

  - **Development cost:** Amount of time wasted in testplan template adaptation to each testing service.
  - **Availability:** Availability of testplan template into IMS testbed.

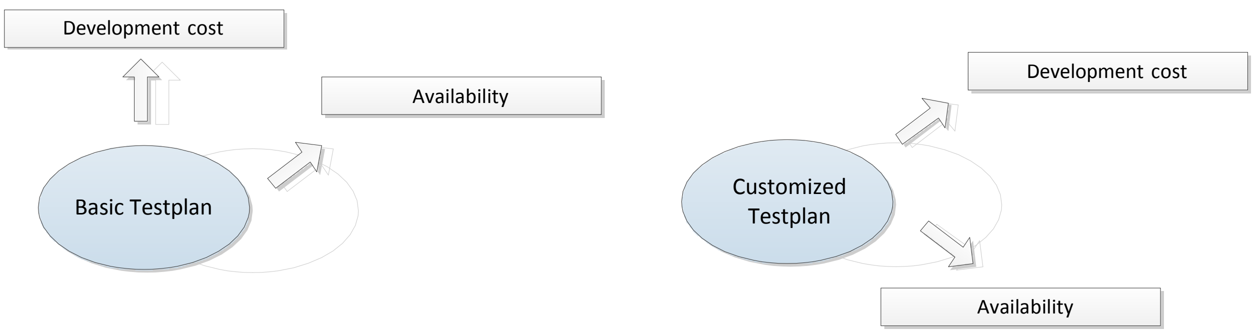

Figure 7: *Test procedure resources and their KPIs*

3. Testing tool oriented KPIs.

- **Automation Testing Tools**

  These tools are oriented to automated testing executions. So, on the one hand error control indicator is defined. On the other hand CPU overload is taken into account.

  - **Percentage of requirements tested fully:** Amount of requirements tested without problem. This indicates if the requirements have been well tested.
  - **CPU using:** CPU using diagram during Automation Testing Tool execution. This KPI helps avoiding CPU overload during automated executions.

- **Test Management Testing Tools**

  These tools are focused on test case design and functionality coverage. So, the main KPI is functionality coverage itself.

- **Coverage of application functionality:** Percentage of functionality covered by the test cases. This indicates if the coverage index exceeds the desired minimum rate. Its value depends on test type: functional, unit testing, etc.

- **CPU using:** CPU using diagram during Test Management Tool execution. This KPI helps avoiding CPU overload during test case design.

- **Requirements Management Testing Tools**
  These tools are based on requirement and test case management. So, it is necessary traceability matrix monitoring.

  - **Traceability index:** Amount of requirement does not link to any test case.

  - **CPU using:** CPU using diagram during Requirements Management Testing Tool execution. This KPI helps avoiding CPU overload during requirement management.

- **Monitoring Testing Tools**
  These tools are basically defined by device parameters. So, a set of parameters is monitored as KPI.

  - **Monitoring standard parameters:** This is a set of parameter monitoring. These parameters are defined by the user, providing the monitoring type and monitoring time interval. Performance statistics are generated based on these parameters.

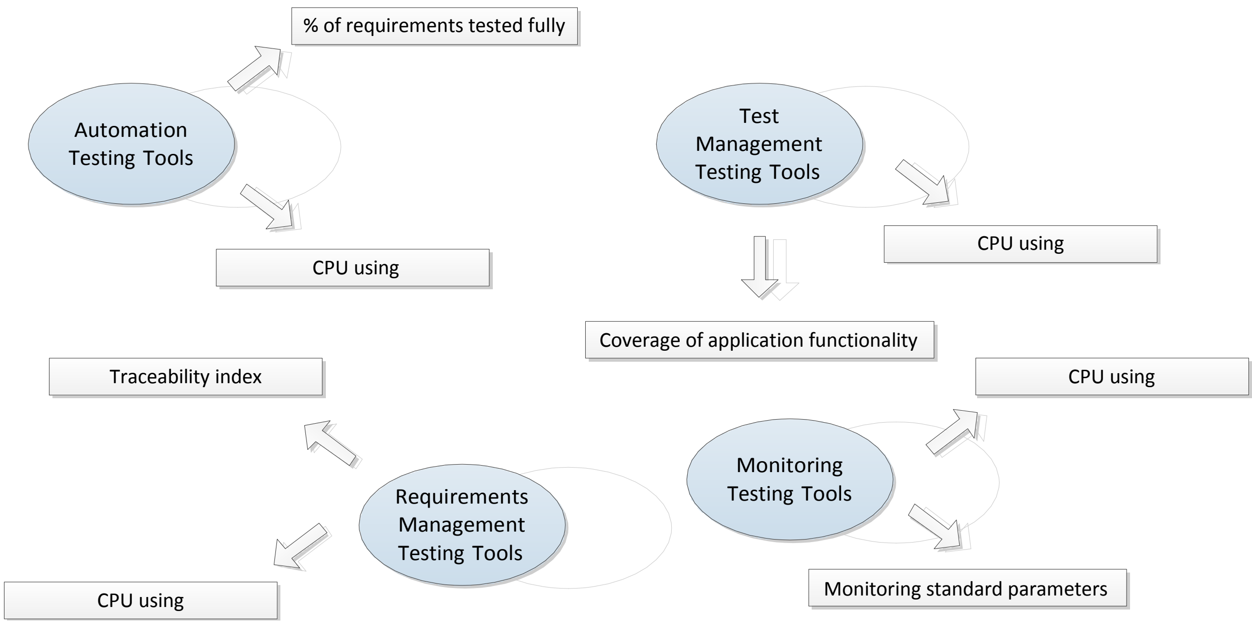

Figure 8: *Testing tools and their KPIs*

4. Human activity oriented KPIs.

- **Human Team**
  IMS testbed using is based on SQS Human Team and its service oriented availability. Therefore, these KPIs indicate team's performance:

  - **Available testers:** Amount of free and ready testing engineer.
  - **Cost of performance:** Amount of time wasted in task conditioning by Human Team.
  - **Test cost:** Amount of time wasted in task development by Human Team.
  - **Report development cost:** Amount of time wasted in report development by Human Team.
  - **Issue resolution cost**: Amount of time wasted in issue resolution by Human Team.

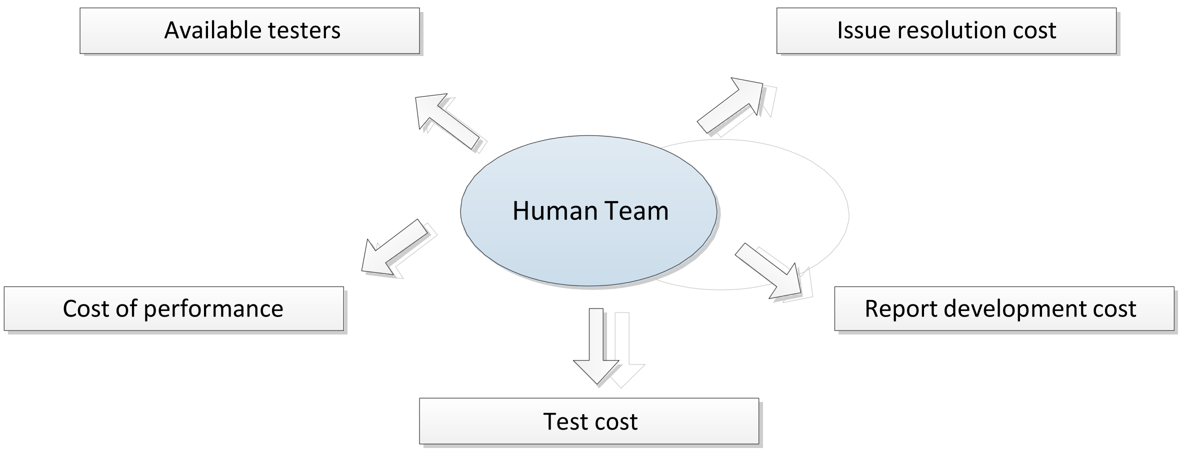

Figure 9: *Human Team and its own KPIs*

4.5 BOTNIA Testbed

As Testbed KPI´s for a Living Lab activity the following can be mentioned:

- Redesign of products/services
- Decisions for implementation of new functions
- New target user groups
- New ideas as result of user involvement
- Increased knowledge among experimenters/developers
- Established relations with new business partners

- Faster innovation process(shortened time for development) by support from end-users for decision making

4.6 KyaTera Testbed

Kyatera is a testbed that interconnect academic institutions, institutes of research and funding agencies in an enviroment for collaborative work, based on a Fiber-to-the-Lab network, dedicated to study and development of science, technologies and Future Internet applications.

The Kyatera network performance meter will evaluate certains metrics to measure network quality, targeting the problem of guaranteeing QoS on multimedia data transmission. The following indicators contain a brief description of data that are expected to be collected and measured during execution of experiments inside KyaTera testbed:

Jitter: The time variation of a periodic signal in relation to a reference clock source. In the context of computer networks, where KyaTera fits, this term is used for the measure of the variability over time of the packet latency across the network. E.g. if the network has a constant latency, i.e. has no variation in the latency, the network has no jitter.
In terms of packet jitter (a.k.a: packet delay variation, in a more precise term) it is expressed as an average of the deviation from the network mean latency. And it is an important quality of service factor in assessment of network performance.

Latency: Is the time delay between two points of a network. This is caused by the physical distance between the network points (this is a fixed amount of time, and can´t be changed or improved) and by the network quality, manufacture structure, router congestion and other factors that can enlarge the time to transmit or receive the data. This is measured in milliseconds (ms).

Packet loss: occurs when one or more packets of data travelling across a the network fail to reach their destination. Can be caused by: signal degradation over the network due to multi-path fading, packet drop because of channel congestion, corrupted packets rejected in-transit, faulty networking hardware, faulty network drivers or normal routing routines, finally packet loss probability is also affected by signal-to-noise ratio and distance between the transmitter and receiver.
This can be on of the greatest problems in medical image analysis, in a way that a missing packet can result in a wrong diagnostic. This is measured in number of packet lost, i.e. a natural number.

Router congestion: Besides making decision as which interface a packet is forwarded to, a router also has to manage congestion, i.e. when packets arrive at a rate higher than the router can process. To manage this problem three policies are commonly used in the Internet: tail drop, random early detection (RED), and weighted random early detection (WRED). Tail drop is the simplest and most easily implemented; the router simply drops packets once the length of the queue exceeds the size of the buffers in the router. RED probabilistically drops datagrams early when the queue is exceeds a pre-configured size of the queue until a pre-configured max when it becomes tail drop. WRED requires a weight on the average queue size to act upon when the traffic is about to exceed the pre-configured size, so that short bursts will not trigger random drops.

Hop count: This is a measure of distance across an IP-based network. It is a count of the number of routers an IP packet has to pass through in order to reach its destination.
Hopcount is usually not used by itself, since any in between router or cable may have or be subject to varying data throughput (bandwidth), load, reliability (especially of cable), and latency.
Hopcounts are often useful to find faults in a network (Time to live), or to discover if routing is indeed correct.This is measured in number of hop used to reach the final node, i.e. a natural number.

Path bandwidth: Bandwidth can be divided in sub-metrics;

- Bandwidth Capacity
  This is the maximum amount data per time unit that the link or path has available, when there is no competing traffic.

- Achievable Bandwidth
  This is the maximum amount of data per time unit that a link or path can provide to an application, given the current utilization, the protocol and operating system used, and the end-host performance capability and load

- Available Bandwidth
  This is the maximum amount of data per time unit that a link or path can provide, given the current utilization

This metric is expressed in bits/second or multiples of it (kilobits/s, megabits/s, etc.)

As at the time of this document the KyaTera testbed isn't finished, we couldn't provide all software that will be used to measure this metrics, but we can provide a list of some open source software that could be part of the final implementation of this testbed:

Jitter/Latency/Packet Loss
http://iperf.sourceforge.net/
Iperf was developed by NLANR/DAST as a modern alternative for measuring maximum TCP and UDP bandwidth performance. Iperf allows the tuning of various parameters and UDP characteristics. Iperf reports bandwidth, delay jitter, datagram loss.

Hop Count
ping
traceroute (tracert on windows)

Path Bandwidth
http://www.spin.rice.edu/Software/pathChirp/
pathChirp is an active probing tool for estimating the available bandwidth on a communication network path. Based on the concept of "self-induced congestion," pathChirp features an exponential flight

pattern of probes we call a chirp. Packet chirps offer several significant advantages over current probing schemes based on packet pairs or packet trains. By rapidly increasing the probing rate within each chirp, pathChirp obtains a rich set of information from which to dynamically estimate the available bandwidth.

http://www.cc.gatech.edu/fac/Constantinos.Dovrolis/bw-est/pathload.html
Pathload is a tool for estimating the available bandwidth of an end-to-end path from a host S (sender) to a host R (receiver). The available bandwidth is the maximum IP-layer throughput that a flow can get in the path from S to R, without reducing the rate of the rest of the traffic in the path.

http://www.cc.gatech.edu/fac/Constantinos.Dovrolis/bw-est/pathrate.html
Pathrate is an end-to-end capacity estimation tool. It uses packet-pairs and packet-trains (a.k.a. `packet dispersion' methods), in conjunction with statistical techniques, to estimate the capacity of the narrow link in the path.

CONCLUSIONS

This document involves a brief overview of the TEFIS platform monitoring capabilities. First, an introduction about monitoring has been presented, highlighting its necessity on a system like TEFIS and the importance of a correct KPIs selection. However, a reflexion about the optimum KPIs has been necessary in order to obtain the best results from this monitoring system. Even being this in an early stage, it has been crucial to define a general architecture, considering the actual TEFIS platform structure and trying to maintain high compatibility with it, so that its future integration does not suppose an excessive impact.

The suggested general KPIs respond to the necessity of having tools for achieving an efficient evaluation of the platform, therefore being able to measure objectively the system performance and shape. Even if those KPIs may somehow gather measures that could be deduced from the different testbeds hosted in TEFIS, not all of the needed information is feasible to be extracted that way, so the sense of these general KPIs is fully supported.

However, much crucial information could be loosed if the system only relied on the overall KPIS. The heterogeneity of the different testbeds, along with the considerable number of them that could be part of TEFIS, make it impossible to contemplate every possible metric or data that could be interesting to monitor. Therefore, each testbed will also propose and implement its own set of KPIs, specifically thought and designed for the functionality provided by that. So, this document also includes information about the KPIs that the currently existing testbeds will control.

In conclusion, it is important to emphasize the benefits that a correct set of KPIs could bring to TEFIS platform, in various and different fields, such as service level or users' satisfaction. In a similar way, knowing vital information as soon as possible will make it possible to fix or improve TEFIS platform and increase its success.