Model-based Technology of Automated Performance Testing

Prof. Dr. B. Pozin
ZAO “EC-leasing”
bpozin@ec-leasing.ru

R. Giniyatullin
ZAO “EC-leasing”
renat@ec-leasing.ru

Dr. I. Galakhov
ZAO “EC-leasing”
igalakhov@ec-leasing.ru

D. Vostrikov
ZAO “EC-leasing”
dvostrikov@ec-leasing.ru

Abstract

The Technology of the scaled-down experiment for evaluation of the performance characteristics of wide class of the systems for business-process automation is developed. The goal of the Technology is to provide adequacy of the results of particular experiment and its components: problem definition, source data and analysis of results of the experiment. The Technology is based on set of metamodels that define main components of experiments. Technology provides technique for adaptation of metamodels to the parameters of any particular automated performance testing experiment.

1. Introduction

The task of systematic estimating of the performance of information system under changes of loading or changes of system architecture is being urgent extremely in life cycle processes of information system. This task is solved by regular performance testing of the system.

At the present time is not created universal methodology of regular performance testing. In this connection the development of aggregate technology of preparation, testing and analysis of results of scaled-down load tests is being urgent. This technology provides adequacy of the problem definition, source data and results of the experiment.

On their own account experiments like this are sufficiently compound and labour-intensive. They demand effort to attract considerable resources of people and equipment.

The scientifically grounded technology of performance testing has to allow the essential decrease of labour-intensiveness, duration of planning and preparation for experiments, testing, analysis of results and consequently their cost.

The aim of this work is to develop such performance testing technology for sufficiently broad class of systems for business process automation, including the banking systems.

Systems for business process are characterized by requirements for the implementation of business processes in the regulatory time frame. Draft proposals for that systems contain a set of non-functional requirements, such as productivity, the number of concurrent end users, restrictions on the holding time of transactions in the system, or reactivity, etc., that is explicitly set the expected performance (performance characteristics) of information system.

2. Types of load tests and evaluated characteristics

An important issue when performance testing runs is to develop a common understanding of a problem by the customer and the contractor (tester). Often, the achievement of this understanding becomes like a long and laborious process, as far as required the development of common conceptual framework for both sides.

Even when a single customer needs the performance testing objectives of different experiments may differ because the customer raises different issues related to performance. Typical problems may be at least the following:

- Is the existing equipment enough to ensure specified performance of information system over the nearest period (1-2 years) at a known annual growth of load?
- Can the information system, which provides the specified capacity, at the same time to

provide the reactivity within certain limits on conditions of an increase in workload?

- Is there no sign of degradation of information system after making changes to the software?
- And several others.

In the course of execution a sufficiently large number of loading experiments revealed the possibility in principle to type a number of load testing problem definition for the «typical» tasks.

Underlying of this approach is based on three pillars:

- Classification of performance testing;
- Classification of information system performance characteristics;
- The principle of non-destructive control while preparing the performance testing.

2.1. Types of testing

Depending on the objectives of performance testing, the following types of load testing can be assigned:

- Evaluation - evaluation of performance characteristics in a single load experiment;
- Analytical - detection of dependencies (e.g., performance of computing resources) in a series of load experiments;
- Configuration - set up and optimization of load characteristics of an information system or its component parts;
- Regression - multiple periodic load testing with constant conditions for signs of degradation of the tested system recognition.

These types of tests are usually organized as scaled-down experiment. Each type of testing has especially planning (single experiment, a series of experiments, the need to store the results of the experiment for statistical processing of historical data on the last experiments). Therewith the scheme of a single experiment stables enough for assessing each type of performance characteristics of information system.

2.2. Performance characteristics classification

For planning of load experiment measurable characteristics of productivity selection plays key role, as well as on the basis of the values of these characteristics the conclusions of results of experiment are made.

The main types of performance characteristics are responsiveness, productivity and use (see table).

Table 1. Types of performance characteristics

Performance characteristics	Calculated index
Responsiveness	Average response time
	Average waiting time
	Average service time
Productivity	Capacity
	Generation
Utilize	Resource utilization
	Relative capacity

They in turn are divided into several calculated (or directly measured) variables; the values of each can be calculated on the basis of directly measurable values. Their structure and relationship under calculation may depend on system engineering platform and the structure of tested system.

2.2.1. Responsiveness.

Reactivity is important for systems operating in real time, i.e. for systems required the implementation of restrictions on the time of decision of certain or all of the functional tasks. In this case the operational characteristics of information systems can be the response time of the system to a user's query or the waiting time for solving the problem (for example, can be solved the problem of minimizing the waiting time of business transactions execution).

The characteristics of reactivity defined as the time between the presentations of input data for the system and the appearance of the imprints. Reactivity may be measured either in terms of end-user or in terms of machining centers. Business transaction and the physical (technical) transaction shall be measured. A transaction is a business unit of work implemented to solve a problem for the business. For example, a banking transaction may consist of a double entry transaction payment; thereby the customer is interested in execution time not only of the single input, but the entire banking operation. The implementation of banking transactions can be performed by several physical transactions, for example, appeals to the database.

Reactivity assessed or calculated on basis of the following performance measures:

- Average transaction response time;
- Average transaction service time;
- Average transaction waiting time.

Response time for a transaction is the time between the transactions beginning and the final execution of the last transaction step.

Response time (Tr) in general is made of: service time (the time actual work is done) (Ts) and waiting time (the time waiting for resource) (Tw):

$$Tr = Ts + Tw.$$

Indicators of reactivity depend on the type of system and the structure of its points of entry and exit.

2.2.2. Productivity.

For a large number of systems their integral capacity is important and measured as the number of business transactions processed by the system in time unit (productivity).

In assessing the productivity, directly measured or calculated values of the following performance measures typically used:

– Generation (*V*);

– Capacity (or an absolute capacity) (*C*).

Generation (V) is defined as the number of completed transactions for a certain period of time:

$$V_i = \frac{N_i}{T}\text{, where}$$

$i = \overline{1, n}$ - Sequence number of the time period;

N_i - The number of completed transactions;

T - Period of time.

Capacity (C) – the maximum number of completed transactions per time unit:

$C = \max(V_1, ..., \mathrm{V_n})$.

Characteristics of the productivity can be used to assess the system as a whole and its parts.

2.2.3. Utilize

Specifications of utilize are used to describe the extent in which the testing system resources are used at a given load.

To utilize the following performance indicators are referred:

– Resource utilization (utilization factor);

– Relative capacity.

Resource utilization measures how much a resource delivers service in a timely basis. The general formula is:

$$U = \frac{\sum_{i=1}^{n} t_{Si}}{T} \times 100\%\text{, where}$$

t_{Si} - Time of service of i-th transaction

n - The number of serviced transactions;

T - Timely basis.

Characteristics of resource utilization and their numbers depend on the equipment used in the system and its constituent resources: CPU, RAM, external drives, I / O channels, etc.

3. Models as means of describing the properties of performance experiment artifacts

Ensuring the adequacy of the problem, the input data and results of performance testing requires the achievement in comprehension of the key aspects of planned experiment between the customer and the contractor. In a rapidly growing business the comprehension should be achieved as soon as possible.

Key aspects of the planned experiments are:

– The problem, defining the purpose of experiment, should be linked to the requirements of the system;

– Input data, defining the object of testing and the necessity of its load, must be defined;

– Set of characteristics and indicators, which are determined the basis of test results must be formed before the start of the experiment.

With the formalization of the subject area (for class of systems) a few frames of metanotions are formed. Metanotion is the metamodel describing possible concepts that could be significant with the subsequent performance testing and evaluating the adequacy of the results.

In the metamodel must be defined methods for collecting baseline data for performance experiment, which ensure the adequacy of performance experiment to the physical operation of the system with the expected load.

These baseline data are:

– Information is about the form of performance testing (assessment, analysis, configuration, regression);

– Information is about measuring performance and productivity;

– Information is about the structure of the system in terms of ways to feeding the load and methods of measurements;

– The information is about the proposed load in a structured way.

When prepared the performance testing task there are four metamodels designed to help to make choice of required features, performance and measurement parameters that adequately characterize the process of operation of tested system:

– Requirements metamodel - characterizes the type of tested system and non-functional requirements (business rules and technical requirements) of tested information system.

– System metamodel - describes the structure of the system as a network of queuing systems (including the composition of elements of «resource» type);

- Workload metamodel - is a description of the number and types of service requirements to the system, the distribution law of service requirements in time of experiment, the rules of proceed of service requirements in the system, the entry points of service requirements in the system (logic level);
- Measurements metamodel - defines a collection of characteristics, rates and values, the input interface of requirements into the system, the method of data collection and transformation algorithms as well as criteria for evaluating the results.

When planning a new performance experiment using the concepts of metamodels the requirements model, the system model as well as workload and the measurements models are formed by selecting metanotions and their values, based on the properties of the tested system and the objectives of performance experiment. Through the use of the metamodel in the planning of a new performance experiment ensured the completeness and integrity of formed models.

For different types of performance testing and various systems these models may be vary.

Metamodels provide a unified approach to the problem definition either for the customer or executor. They offer advantages such as rapid understanding and agreement on the objectives of the experiment, the rapid development of ways to achieve these goals and a common comprehension of ways to achieve them. Metamodels are the bridge between the non-formalized requirements of customers with formal description of the performance experiment in the form of workload models. This allows simplify significantly the planning and automation of implementation of the performance experiment.

3.1. Requirements metamodel

Requirements metamodel contains the rules for formalize the requirements of performance system. Such claims do not contain information about the functions performed by the system and, therefore, are called non-functional.

Non-functional requirements may be defined by business rules of the organization, which operates the system. Regulations may specify the time frame of implementation of various processes in the system. To comply with these time limits resources of the system are directly linked to its capacity.

Depending on the specified non-functional requirements, objectives of performance experiment are formed. Also depending on this, examined characteristics are selected.

Requirements metamodel assigned to describe non-functional requirements for the system. Requirements metamodel can be presented as follows:

$R = B \cup T$, where

R - The set of requirements for the system;

B - The set of business rules;

T - The set of technical requirements.

Business rules include or are related to technological processes, corporate regulations, policies, standards, legislative acts, intra-corporate initiatives, accounting practices, computing algorithms, etc. Business Rules Group provides a definition of business rules as “the regulations which define or limit some aspects of business.“

Technical Requirements - a technical requirement pertains to the technical aspects that your system must fulfill, such as performance-related issues, reliability issues, and availability issues.

Requirements metamodel specifies description rules of verbal requirements model, containing the non-functional requirements of the system.

When choosing the specific values of the concepts or the specific rules, actually on a metamodel, there are formed model for the specific requirements of the tested system.

3.2. System metamodel

The system metamodel makes it possible to describe the structure of the system as a network of queuing systems, consisting of the elements of the type of «resource» and the linkages between them.

The system metamodel has a complicated structure and determines the rules of tested object model formation, which is described until the level of involved in the testing devices and software (components, services), with certain characteristics of performance.

In the metamodel, it is assumed that a certain class of systems can be represented by a collection of some concepts and rules of their relationship. When choosing the specific values of the concepts or the specific rules, actually on a metamodel there are formed model of the specific tested system.

The system metamodel can be represented as follows:

$\sigma = \{\{U(p)\}, \{S\}, K_S, K_U\}$, where

$\{U(p)\}$ - A set of tested object devices with the characteristics of performance;

$\{S\}$ - A set of bundled software (components, services);

K_S - The matrix of relationships between software systems, lines of which are the source, the columns are receivers, and cells indicates the existence of links between them;

K_U - The matrix of relations between software systems and devices, which characterizes the number of allocated by device resources for software system. Rows of the matrix are the software systems, columns are the computer systems. The elements of the matrix are the vectors of the allocated resources.

Feeding the load points and points of collection performance characteristics tend to be any connection in the matrices K_S and K_U, and which are specifically defined in the workload and measurements models.

The rules describe the object of testing are quite complex. They are divided into a description of software and hardware, as well as the relationship between them.

The level of detail of the system as the object of testing is determined by the objectives of the experiment. It may be a single software system (the system as a whole), and many software packages or applications (modules).

The description of software tools based on the principle, that the tested system is viewed as a black box with multiple inputs and outputs. Depending on the needs the system can be decomposed on software systems, which are also regarded as a black box with multiple inputs and outputs and that can be linked to each other. Communication between the blocks can be defined as the point feeding the load (specifically defined in the workload model), and points of collection attributes (specifically defined in the model of measurement).

Description of hardware should include the static characteristics of equipment, such as the number of processors, memory, and dynamic characteristics, such as the rules of the dynamic reallocation of resources. These rules define the possible joint use of certain resources, the priorities of the use of resources by one or another software system, the weighting of allocations the corresponding resources for software systems.

3.3. Workload metamodel

The workload metamodel determines the structure of the input loading stream.

The workload metamodel can be represented as follows:

$L = \{F, M, I\}$, where

F - The set of functions describing the distribution of load input into the system;

M - The multi-dimensional matrix, whose dimensions may be workload types, such as sources of load flow, the names and types of flows, and the elements are their quantity;

I - The set of interfaces for the input of load (as a link to the model of the system).

The flow of load is structured by its dynamic and static properties.

The distribution laws of flow of the load are dynamic properties of this flow. The distribution laws may be:

- Deterministic;
- Probabilistic.

Deterministic distribution laws of load over time expect having the schedules of entrance the given workload into the system. In extreme case the schedule may include the proceeding time of each request into the system. Probabilistic distribution laws of load over time suppose an indication of normal, uniform, exponential, or other law distribution.

The quantitative composition of the load flow determines its static properties. The size of the load flow can be structured on several criteria, including:

- By type of workload;
- By type of senders.

For the considered class of systems there are three basic types of loads, notably:

- Traffic;
- Communications;
- Events.

The load as the traffic created by users in a system built on client-server architecture. The most common types of traffic are HTTP-and SQL-traffic. The actions of users in the automated workplace (AWP), leads to the formation of queries (e.g., SQL-queries) to the server. In every system there are different types of automated workplace used as the senders of requests.

Another type of workload is a load of messages. Messages are used as units of information exchange between parts of large systems. Depending on the destination, each message can be divided into several types. Messages containing the documents for processing, in turn, can be divided into types according to their formats. In addition, messages can be:

- Single (containing a single document);
- Package (containing a set of documents).

Depending on the sender, each message can be encrypted and signed by one or more of the electronic digital signature.

There are workloads created by a tested system as a result of different events. These include the implementation of certain regulatory procedures executed by schedule or by the operator.

When choosing the specific values of the concepts or specific rules by the metamodel, the workload model of a specific tested system is formed. The workload model is described at the planning stage of the experiment.

3.4. Measurement metamodel

Measurement metamodel is intended to unify the description of:

- Ways to get the measured values in process of the performance testing of information system;
- Possibility in principle of measurements process formulation;
- Typical way of assessing the measures and characteristics;
- General properties of the tools to analyze the possibilities of their using to automate the measurements.

Measurement metamodel can be represented as follows:

$\Delta = \{\{|U|\}, \tau, \mu, R, \omega\}$, where

$\{|U|\}$ - A list of measured quantities for each type of device of information system or its part;

τ - Frequency and porosity of measurements;

μ - A set of estimates and their relationship with measured values;

R - Model rules and algorithms for obtaining the estimates;

ω - Standard criteria for evaluating the results.

In the process of preparing the performance experiment, the selection of specific concepts or specific rules is realized. That is, in fact, the model for specific measurements is designed by the metamodel.

3.5. Models interaction

Presented models are closely interrelated and interact. All of each model should be identified for every experiment.

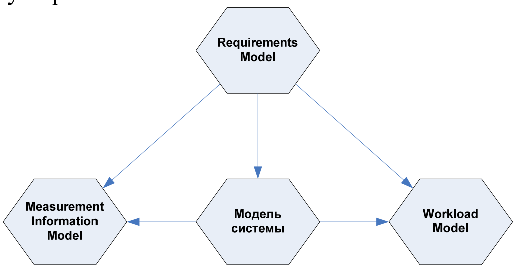

Figure 1. Models interaction

The requirements model is initiating model.

After the formation of the purposes of performance testing, we can determine:

- Test object: what part of the system will be tested - a system model;
- What are the characteristics and rates will determine and for what criteria they should be correspond - a requirements model;
- What are the workflow into the system from a controlled process - a workload model;
- What parameters should be measured and in what locations to obtain the necessary results - a measurement model.

Several characteristics of the measurement model and workflow model can be described only after the completion of the description of the system model. These characteristics, for example, include a description of load entering points and the points to collect characteristics.

4. Model-based testing process

Testing process (Figure 2) consists of the following stages:

- Definition of the goals of testing;
- Development the program and methods of testing;
- Preparation for testing;
- Feeding the load;
- Data collection;
- Interpretation and analysis of the results.

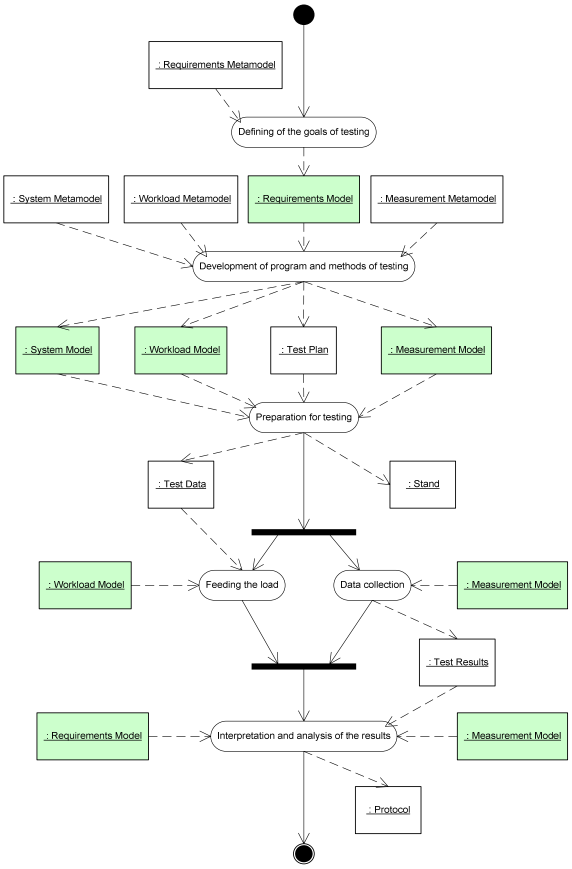

Figure 2. The process of model-based performance testing

4.1. Definition of the goals of testing

The goals of testing are defined on the basis of the customer's needs for evaluation or forecasting operational characteristics of informational system and corresponding non-functional requirements.

The customer cannot always define the goals of testing adequately. During the process of accommodation of the main goal and the process of development of the program and methods of testing, secondary testing goals and restrictions can be revealed, that are also due to accommodation. The developed requirements metamodel can help properly interrogate the customer at an early stage and adequately define the goals of testing, which reduces the number of iterations of goal accommodations and overall time of developing main and secondary goals of performance testing.

4.1.1. Definition of requirements model. At the stage of defining the goals of testing using requirements metamodel, the object of testing is analyzed and the primary requirements model is formed, which is later discussed with the customer, edited and approved.

The requirements metamodel represents verbal description of non-functional requirements. These requirements are grouped by types and objects, which they are applied to. This model also contains criteria for evaluation of characteristics and calculated indices obtained as a result of performance testing.

4.2. Development the program and methods of testing.

This is one of the most important ant difficult stages of performance testing. At this stage system model, workload model and measurement model are developed based on corresponding metamodels.

The three models are developed simultaneously and are closely connected to each other. For example, the definition of the interfaces of feeding the load and characteristics collection points is possible only after the description of the system model.

Upon completion of this stage all approved models are included in the document "The Program and Methods of Testing" that represents the plan of performance testing and is approved by the customer.

4.2.1. Development of system model. At the stage of development of program and methods, according to set goals, verbal and graphical models of the system are developed, which is later discussed with the customer, edited and approved.

System model represents the description of the object of testing, which includes software and hardware, their connections and allocated resources (storage, processor etc.)

Ошибка! Источник ссылки не найден. shows an example of system model in form of graphical scheme of interaction of software systems and computer systems.

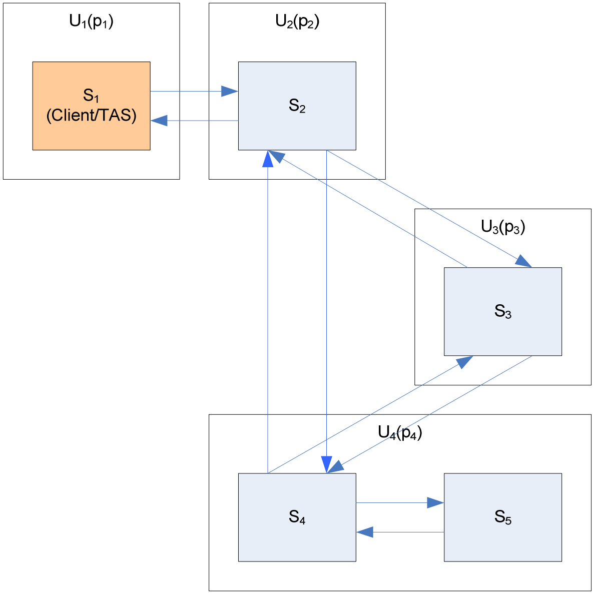

Figure 3. An example of system model in form of graphical scheme

Matrices K_S and K_U for the presented scheme look as follows:

$$K_S = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix}; \quad K_U = \begin{pmatrix} p_1 & 0 & 0 & 0 \\ 0 & p_2 & 0 & 0 \\ 0 & 0 & p_3 & 0 \\ 0 & 0 & 0 & p_4 \\ 0 & 0 & 0 & p_4 \end{pmatrix}$$, where p_i is the vector of performance characteristics, that describes the number of resources allocated on computer system for a certain software system.

4.2.2. Development of workload model. At this stage of development of program and methods, according to set goals, workload model is developed, which is later discussed with the customer, edited and approved. The values of load on the informational system, expected by the customer, and set of load sources, which will be emulated during the tests, should be considered in the model. In fact, the scenario of loading of the system, corresponding the goals of performance testing, is developed in association with the customer. As a result of detailed study of the scenario, the workload model, in which all load parameters are defined, is developed.

The workload model represents the description of test data and rules of putting them into the system. In fact, according to the workload model, source data for tool-generated load are formed.

4.2.3. Development of measurement model. At the stage of development of program and methods, according to set goals, verbal measurement model is developed, which is later discussed with the customer, edited and approved.

Measurement model represents the description of the collected characteristics, methods of their collection and interpretation.

In measurement model method and means of collecting values of measured parameters are defined. It is very convenient to use measurements metamodel in which standard mechanisms of measurements which are usually supported by the tools accessible in the market are defined. Methods of collection of values of measured parameters can strongly differ depending on set testing tasks.

In simplified form these methods can be subdivided into methods of non-destructive control, methods of destructive control and mixed methods.

In case of non-destructive control system resources of operational environment of the IS automated testing tools are used for collection of values of measured parameters. In case of destructive control a part of measuring means are integrated into automated testing tools. In case of mixed control integration of measuring means is possible both into the system tools and into application software.

Ideally, means of collecting values of measured parameters shouldn't influence on tested system. In cases when it is necessary to take advantage of measurement means, which influence tested system (set interruptions and trigger events) separate tests are performed in order to evaluate a level of influence of measurement means on the performance of tested object. Obtained values of level of influence are taken into account further at interpretation of results.

4.3. Preparation for testing

Preliminary accomplishment of following operations is required for carrying out performance testing:

- Preparation of standa;
- Preparation of test data;
- Preparation of resources for feeding the load;
- Preparation of the means of measurement.

Preparation of the test stand is carried out according to the approved system model and includes organization of technical equipment and software and ensuring of fulfilment of additional restrictions, for example, time synchronization on all installations of the tested object, etc.

Test data for the performance testing are prepared according to the approved workload model. Test data are either generated or real data that meet the characteristics according to the workload model are used.

Interfaces of feeding the load (and appropriate tools) are customized according to the workload model.

By preparation of means of data collecting according to measurements model, suppliers and receivers are customized.

The result of the stage of preliminary operations is the test stand ready for testing (being in an initial state) and test data ready for feed.

4.4. Feeding the load

At the stage of feeding the load the prepared test data are fed to the tested system according to the laws of their allocation described in workload model.

4.5. Data collection

Data collection is done by means of measurement.

Depending on the methods of collection defined in measurements model, the data can be collected both in the course of testing as they arrive, and after performance testing passes. The first case is preferable because the data collected in process of arrival can be used for the control of passing of testing process.

Data collecting lasts until the process of testing will not satisfy the criteria of completion and moving to the next stage, defined in measurement model.

The preprocessing of the data also depends on the way of processing described in measurement model and can be fulfilled both in process of arrival of the data, and after experiment applied to the whole collection of the gathered data.

4.6. Interpretation and analysis of the results

The received results of testing are interpreted according to measurement model for the further analysis: calculation of metrics and estimation of values of operating characteristics of the IS are carried out. On completion of interpreting the results, conclusions concerning the correspondence of the IS to the goals are made taking into account the restrictions and criteria of requirements model. Results of trials in practice are drawn up in the form of the protocol of performance testing.

5. Automation of model-based testing technology

Due to formalizing of the model-based technology of performance testing, it is possible to automate many operations carried out during semi natural performance experiment.

Ошибка! Источник ссылки не найден. shows the technology of performance testing with highlight of automated operations.

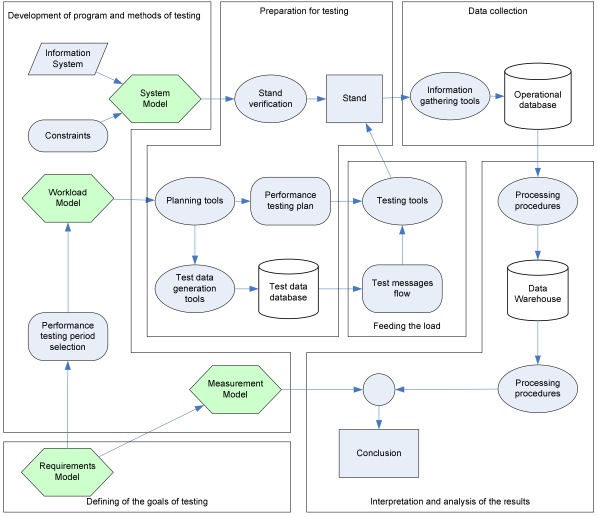

Figure 4. Technology of performance testing with highlight of automated operations

For the stage of preparation for testing the check of the test stand based on the system model is automated. Errors during stand preparation can lead to failure of performance experiment, as disability of parts of the stand may be revealed after the beginning of the experiment. Moreover, incorrect customizations of the stand can distort the results of performance testing that can lead to necessity of repetition of the experiment. The existing formalized system model has allowed automating the stand check, having reduced the time of such check. When errors in configuration of the stand and its technical equipment and software are revealed, automated check can be repeated multiple times at low labor input due to automation of this process.

At the stage of preparation for testing planning of performance experiment and generation of test data based on workload model is automated. At testing planning time and quantitative parameters of test data flow, and also scenarios of feeding the load are defined. The instrument of automation of planning of performance experiment allows entering appropriate input data about forthcoming test into previously defined screen forms. On the basis of this information the instrument of the automated generation prepares the test data. The automated preparation of the test data provides their complete correspondence to a current state of tested system. Thus, it is guaranteed that all test data will be accepted by the system and will be handled the same way as the real ones.

The stage of feeding the load is automated using testing tools that can send different types of loads in accordance with the distribution laws. The load in the form of messages filed from a prepared base of test data. The load as traffic may be submitted in the form of HTTP-, and SQL-traffic. The composition of the traffic is generated on the basis of automated analysis of code of queries, which are sent by automated workplaces of tested system by execution different operations of any user. The load as events is formed through emulation of actions, which the operator makes. Emulation is made thought testing tools.

At the stage of collecting data the basic information about the values of measurable indicators is automatically collected. This information includes such indicators as loading of CPU, using of memory, time of incoming transaction into the system, the time of the response reception from the system, etc. All of the data stored in the operational database for further processing.

At the stage of interpretation and analysis of results based on the measurements model the multi-step automated processing of raw data is performed. At first, measurements on the base of rules and algorithms of measurements model are formed. Estimated measurements are placed in a data warehouse for subsequent analysis. Based on the data warehouse the analytical reports are formed containing estimates of the values and criteria for evaluation in accordance with the measurements model and requirements model. Due to the presence of the repository it's possible to analyze not only results of passed performance experiment but also a comparative analysis of several regression tests.

The described technique is applied for four years. The volume of experimental work over the past 4 years as follows (summary):

- Amount of stress experiments, more than 40;
- The number of generated messages: more than 75 million;
- The average number of messages in a single experiment: about 2 million.

Table 2 presents data on the average duration of work.

Table 2. Duration of work phases

Phase	Duration
Planning of performance testing	1 week
The collection of baseline data for performance testing	2-3 days
Generation of the flow of messages for performance testing	1 day
Checking the correctness of the generation	1 day
Performance testing	is dependent on the experiment plan
Processing the results	3-4 hours
Documenting the results	1 week

The table shows that the developed technology and means of its automation make it possible to prepare and conduct the performance testing within a reasonable time frame for the customer.

6. Conclusion

During the testing of major systems, working on expensive technology, it is important to minimize the risks such as shift the timing of performance testing, repeat testing, etc.

Such risks occur, usually, due to a lack of common understanding of the performance testing purposes by customer and contractor, and errors in planning the experiment.

Applying the developed technology, it is possible to reduce such risks, and decrease labor intensity and duration of performance testing, as hence to reduce their value.

7. References

[1] Kostogryzov A., V.Panov, B.Pozin, V.Sablin. Mathematical modeling of processes in systems life cycles in compliance with standarts requirements of ISO/IEC 15288 and ISO/IEC 12207, Spincose, Montreal, Canada, 2003.

[2] Козлов А.Н., Позин Б.А., Технология комплексного нагрузочного тестирования банковской системы, Международная научно-практическая конференция «Реинжиниринг бизнес-процессов на основе современных информационных технологий системы управления знаниями», 2006 г. (РБП-СУЗ-2006), Москва, с.130-131.

[3] ISO/IEC 1539 “Information technology – Software engineering – Software measurement process”