

## ЗАДАЧИ И МЕХАНИЗМЫ УПРАВЛЕНИЯ ИТ-ПРОИСШЕСТВИЯМИ<sup>1</sup>

**Зимин В.В.,**

к.т.н., доцент, профессор кафедры автоматизации и информационных систем,  
Сибирский государственный индустриальный университет, г. Новокузнецк  
zimin.1945@mail.ru

**Кулаков С.М.,**

д.т.н., профессор, заведующий кафедрой автоматизации и информационных систем,  
Сибирский государственный индустриальный университет, г. Новокузнецк  
kulakov-ais@mail.ru

**Зимин А.В.,**

аспирант кафедры автоматизации и информационных систем,  
Сибирский государственный индустриальный университет, г. Новокузнецк,  
zimin.1955@mail.ru

**Левченко С.П.,**

аспирант кафедры автоматизации и информационных систем,  
Сибирский государственный индустриальный университет, г. Новокузнецк  
splevchenko@bk.ru

**Аннотация:** *Материалы XVIII Международной открытой научной конференции (Lorman, MS, USA, January 2013)/  
Главный редактор, доктор технических наук, профессор, О.Я.Кравец. – Lorman, MS, USA: Science Book Publishing House, 2013.*

## TASKS AND MECHANISMS OF MANAGEMENT OF IT INCIDENTS

**V. V. Zimin, S. M. Kulakov, A. V. Zimin, S. P. Levchenko**  
Siberian State Industrial University

**Abstract:** *Proceedings of the XVIII-th International Open Science Conference (Lorman, MS, USA, January 2013)/ Editor in  
Chief Dr. Sci., Prof. O.Ya. Kravets. - Lorman, MS, USA: Science Book Publishing House, 2013.*

**В**ведение. Важными задачами стадии жизненного цикла ИТ-сервиса «эксплуатация» является задачи обнаружения, распознавания и устранения ИТ-происшествий (ИТ-происшествие – отклонение от нормативного режима функционирования ИТ-актива). При значительном количестве предоставляемых ИТ-провайдером сервисов, в силу многообразия сервисных активов, разнообразия внешних и внутренних возмущающих воздействий, влияющих на функционирование активов, число ИТ-происшествий у разных провайдеров может измеряться сотнями и тысячами в сутки, рисунок 1. Эффективное распознава-

ние и обработка ИТ-происшествий, обеспечивающие минимальное снижение удовлетворенности потребителей ИТ-сервисов, представляет сложную задачу. В статье предлагается декомпозиция этой задачи, формулируются постановки частных задач, описывается процедура распознавания ИТ-проблем.

**1. Классификации ИТ-происшествий**, ориентированная на разработку процедур обработки происшествий для каждого класса. В качестве оснований такой классификации предлагаются две характеристики ИТ-происшествия:

- *вид ИТ-происшествия*; определяет степень влияния происшествия на ключевую для клиента характеристику сервисов – доступность, позволяет сформулировать цель процедуры обработки ИТ-

<sup>1</sup> Публикация подготовлена в рамках научного исследования, выполненного при поддержке государства в лице Минобрнауки России.

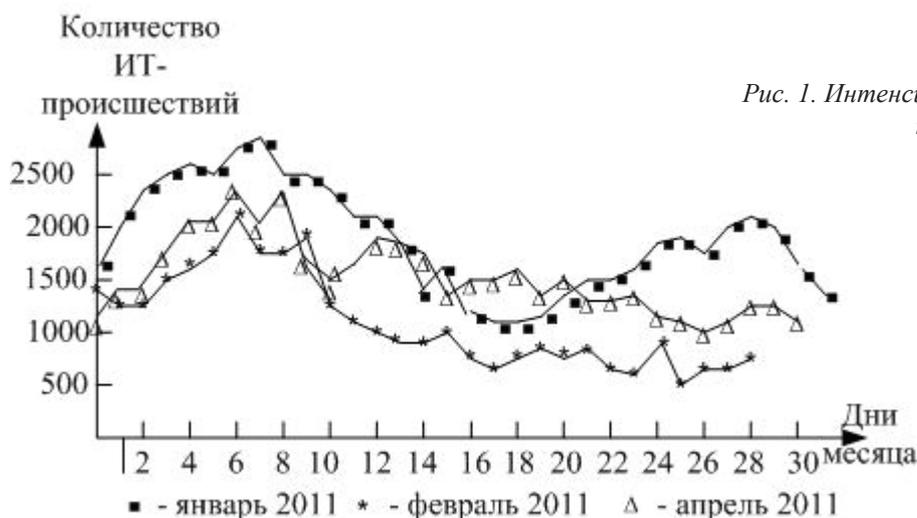


Рис. 1. Интенсивность потока ИТ-происшествий на металлургическом предприятии

происшествия, исходя из интересов потребителей сервиса;

- *тип происшествия*; определяется типом того ИТ-актива, который вызвал происшествие и который определяет специализацию ИТ-деятельности по устранению происшествия (в ИТЛ-3 [1,2] различают девять типов ИТ-активов).

На практике обычно различают три вида происшествий: *событие, инцидент, проблема* [3].

*Событие* – обнаруженное происшествие связанное с одним или несколькими ИТ-активами, которое обусловлено их некорректным функционированием, не приводящим однако к недоступности сервиса или сервисов (событие является «предвестником» недоступности).

*Инцидент* – обнаруженное происшествие с ИТ-активом(ами), которое приводит к потере доступности ИТ-сервиса для клиента, но не связано с ошибками или неэффективными проектными решениями, соответствующими принятой нормативной модели сервиса.

*Проблема* – обнаруженное происшествие с ИТ-активом(ами), которое приводит к потере доступности ИТ-сервиса для клиента и связано с наличием ошибки или неэффективностью проектного решения, реализующего нормативную модель сервиса, то есть – с некачественным проектированием.

Проблема проявляется в многократно воспроизводящихся инцидентах («проблемных» инцидентах).

Под *категорией ИТ-происшествия* будем понимать совокупность вида и типа происшествия. При девяти типах и трех видах получаем 27 категорий и соответствующих им классов ИТ-происшествий. Для каждого класса необходим специализированный механизм обработки ИТ-происшествий. Чтобы воспользоваться этими механизмами, необходимо предварительно решить задачу распознавания категории выявленного ИТ-происшествия, которая не всегда является тривиальной.

## 2. Задачи синтеза механизмов обработки событий, инцидентов, проблем.

*Постановка задачи синтеза механизма обработки событий.* Пусть  $E = \{e\}$  – множество событий с ИТ-активами на некотором плановом интервале времени  $T$ , а  $\{E_k \mid k \in \overline{1, K}\}$  – разбиение множества  $E$  на классы, где  $E_k$  – подмножество событий, связанных с активами типа  $k$ , т.е.  $E = \bigcup_{k=1}^K E_k$  и  $E_k \cap E_p = \emptyset, k \neq p$ .

Обозначим через  $\Delta t_k(e)$ , интервал времени, в течение которого актив типа  $k$ , с которым произошло событие  $e$ , сохраняет работоспособность. Пусть  $\mu(E_k)$  – искомый механизм обработки событий  $e \in E_k$ ,  $c(\mu(E_k))$  – нормативные затраты на обработку одного события  $e \in E_k$ , а  $\tau_k(\mu(E_k))$  – нормативная длительность обработки события  $e$  с помощью механизма  $\mu(E_k)$ . Введем функцию:

$$\varphi_k(\mu(E_k)) = \begin{cases} 1, & \text{если } \tau_k(\mu(E_k)) \leq \Delta t_k(e) \\ 0, & \text{если } \tau_k(\mu(E_k)) \geq \Delta t_k(e) \end{cases}$$

где  $k = \overline{1, K}$ . (1)

Тогда задача синтеза механизма  $\mu(E) = \{\mu(E_k) | k = \overline{1, K}\}$  обработки событий формулируется следующим образом:

$$\sum_{k=1}^K \sum_{e \in E} \varphi(\mu(E_k)) \rightarrow \max_{\mu(E)}$$

$$\sum_{k=1}^K |E_k| c(\mu(E_k)) \leq C^*(E). \quad (2)$$

Т.е. требуется синтезировать совокупность механизмов обработки событий для каждого типа активов, которая максимизирует количество событий, не приводящих к инцидентам и суммарные затраты на функционирование которой не превышают заданной величины  $C^*(E)$ .

*Постановка задачи синтеза механизма обработки инцидентов.* Пусть  $I = \{i\}$  - множество инцидентов, произошедших на некотором плановом интервале времени  $T$ , а  $\{I_k | k \in \overline{1, K}\}$  - разбиение множества  $I$  на классы, где  $I_k$  - подмножество инцидентов типа  $k$ , т.е.  $I = \bigcup_{k=1}^K I_k$  и  $I_k \cap I_p = \emptyset, k \neq p$ . Пусть также  $\mu(I_k)$  - искомый механизм обработки инцидентов  $i \in I_k$ ,  $c(\mu(I_k))$  - нормативные затраты на устранение одного инцидента  $i \in I_k$ , а  $\tau(i|\mu(I_k))$  - нормативная длительность устранения инцидента  $i \in I_k$  механизмом  $\mu(I_k)$ . Обозначим через  $S_i = \{s_i\}$  совокупность сервисов, потерявших доступность из-за инцидента  $i$  и пусть  $d(s_i)$  - добавленная стоимость, создаваемая сервисом  $s_i$  в единицу времени. Тогда задача синтеза механизма  $\mu(I) = \{\mu(I_k) | k = \overline{1, K}\}$  обработки инцидентов формулируется следующим образом:

$$\sum_{k=1}^K \sum_{i \in I_k} \sum_{s \in S_i} d(s_i) \tau(i|\mu(I_k)) \rightarrow \min_{\mu(I)}$$

$$\sum_{k=1}^K |I_k| c(\mu(I_k)) \leq C^*(I), \quad (3)$$

где  $|I_k|$  - мощность множества  $I_k$ .

Т.е. требуется синтезировать совокупность механизмов обработки инцидентов для каждого типа активов, которая минимизирует потери потребителей из-за недоступности сервисов вследствие происшедших инцидентов и затраты на которую не превышают заданной величины  $C^*(I)$ . При «незрелых» процессах ИТ-провайдера, величины  $d(s_i)$  могут быть оценены по значению приоритета инцидента [1]. или задачу (3) можно упростить:

$$\sum_{k=1}^K \sum_{i \in I_k} \tau(i|\mu(I_k)) \rightarrow \min_{\mu(I)}$$

$$\sum_{k=1}^K |I_k| c(\mu(I_k)) \leq C^*(I). \quad (3a)$$

В (3a) минимизируется общее время недоступности сервисов из-за происшедших инцидентов.

*Постановка задачи синтеза механизма обработки проблем.* Пусть  $J = \{j\}$  - множество проблем, выявленных на плановом интервале времени  $T$ , а  $\{J_k | k \in \overline{1, K}\}$  - разбиение множества  $J$  на классы, где  $J_k$  - подмножество проблем типа  $k$ , то есть  $J = \bigcup_{k=1}^K J_k$  и  $J_k \cap J_p = \emptyset, k \neq p$ . Обозначим через  $\mu(J_k)$  - искомый механизм устранения проблемы  $j \in J_k$ , через  $\tau(j|\mu(J_k))$  нормативная длительность устранения проблемы  $j$  механизмом  $\mu(J_k)$ , а  $c(\mu(J_k))$  - нормативные затраты на обработку механизмом  $\mu(J_k)$  одной проблемы. Пусть также  $I_j = \{i_j\}$  - множество инцидентов, вызванных проблемой  $j$ ,  $\mu(I_k|J_k)$  - механизм устранения инцидента  $i_j \in I_k$  при условии, что имеет место проблема  $j \in J_k$ ,  $c(\mu(I_k|J_k))$  - нормативные затраты на механизм  $\mu(I_k|J_k)$ , а  $\tau(i_j|\mu(I_k|J_k))$  - нормативная длительность устранения инцидента  $i_j \in I_k$  механизмом  $\mu(I_k|J_k)$ . Пусть еще  $S_{ij} = \{s_{ij}\}$  - множество сервисов  $s_{ij}$ , ставших недоступными из-за инцидента  $i_j$ , вызванного проблемой  $j$ , а  $s_{ij}(\mu(I_k|J_k))$  - восстановленную механизмом  $\mu(I_k|J_k)$  версию сервиса  $s_{ij}$ . Потери  $\delta C(s_{ij})$  потребителя сервиса  $s_{ij}$  за период устранения проблемы  $i_j$  будут

$$\delta C(s_{ij} | \mu(I_k|J_k)) = d(s_{ij}) \tau(i_j|\mu(I_k|J_k)) + \{d(s_{ij}) - d(s_{ij}(\mu(I_k|J_k)))\} * \{\tau(j|\mu(J_k)) - \tau(i_j|\mu(I_k|J_k))\}. \quad (4)$$

где  $d(s_{ij})$  и  $d(s_{ij}(\mu(I_k|J_k)))$ , соответственно, добавленные стоимости, создаваемые в единицу времени сервисами  $s_{ij}$ , и  $s_{ij}(\mu(I_k|J_k))$ . Тогда задача синтеза механизма обработки проблем формулируется следующим образом:

$$\sum_{k=1}^K \sum_{j \in J_k} \sum_{s_{ij} \in S_{ij}} \delta C(s_{ij} | \mu(I_k | J_k)) \rightarrow \min_{\mu(J), \mu(I|J)} \min_{\mu(I|J)} \sum_{k=1}^K \{J_k | c(\mu(J_k)) + I_k | c(\mu(I_k | J_k))\} \leq C^*(J). \quad (5)$$

То есть необходимо разработать такие совокупности механизмов  $\mu(J) = \{\mu(J_k) | k = \overline{1, K}\}$  и  $\mu(I|J) = \{\mu(I_k | J_k) | k = \overline{1, K}\}$ , которые минимизируют потери потребителей сервисов, вызванных проблемами  $j \in J$  и удовлетворяют ограничению на величину затрат на создание и функционирование этих механизмов. Необходимость создания механизма  $\mu(I|J) = \{\mu(I_k | J_k) | k = \overline{1, K}\}$  обусловлена тем, что устранение ошибок и неэффективных решений проектирования может потребовать значительного времени, в течение которого бизнес-процессы клиентов не должны простаивать. Отметим, что механизмы  $\mu(I_k|J_k)$  представляют собой процедуры определения и реализации так называемых «обходных» путей разрешения инцидентов, вызванных проблемами. Наиболее часто применяемым на практике «обходным» путем снижения потерь от ошибок проектирования является процедура применения предыдущей (доказавшей работоспособность) версии сервиса и/или сервисного компонента.

При «незрелых» ИТ-процессах (не формирующих сведений о  $d(s_{ij})$  и  $d(s_{ij}(\mu(I_k|J_k)))$  для оценки добавленных стоимостей можно воспользоваться значением приоритета проблемы или ограничиться минимизацией общего времени недоступности сервисов для потребителей.

Типовые нормативные механизмы обработки событий, инцидентов и проблем, являющиеся решениями задач (2), (3а), (5), приведены в ИТ-3 [3].

**3. Разработка процедуры распознавания ИТ-происшествий.** Пусть  $CI^q = \{ci_n^q | n = \overline{1, N^q}\}$

– некоторая ИТ-конфигурация [3], в отношении которой зарегистрировано ИТ-происшествие,  $ci_n^q$  – компоненты конфигурации, отраженные в базе данных конфигураций (CMDB). Если  $CI^q$  – ИТ-сервис, то  $ci_n^q$  – сервисные компоненты. Пусть  $b(c_n^q) = \{b_m(c_n^q) | m = \overline{1, M_n^q}\}$  – базовый уровень конфигурационного элемента  $ci_n^q$ , описываемый совокупностью нормативных значений характеристик  $ci_n^q$ , соответствующих штатному режиму его функционирования. Определим базовый уровень  $B(CI^q)$  конфигурации  $CI^q$  как совокупность базовых уровней ее компонентов:  $B(CI^q) = \{b(ci_n^q) | n = \overline{1, N^q}\}$ . Под ИТ-происшествием  $b^\Delta(ci_n^q)$  с конфигурационным элементом  $ci_n^q$  будем понимать зафиксированное отклонение фактических значений характеристик конфигурационного элемента от его нормативных значений, то есть зафиксированное отклонение от значений его базового уровня:  $b^\Delta(ci_n^q) = \{b_l^\Delta(c_n^q) | l \in L \subset \overline{1, M_n^q}\}$ , где  $L$  – количество характеристик конфигурационного элемента, не соответствующих базовому уровню. Определим происшествие  $B^\Delta(CI^q) = \{b^\Delta(ci_n^q) | n = \overline{1, N^q}\}$  с конфигурацией  $CI^q$ , как совокупность происшествий с составляющими ее компонентами.

*Постановка задачи синтеза механизма распознавания категории ИТ-происшествия.* Заметим, что распознавание событий является тривиальной задачей, так как сообщения о всех обнаруженных событиях содержат необходимую для определения их категории информацию (например, сообщение датчика о задымленности или о повышенной температуре в центре обработки данных; информация о не выходе на работу системного администратора; сообщение программного агента о том, что данные с магнитного диска считываются с 5-6 попытки). Поэтому далее ограничимся задачей распознавания «проблемных» и «непроблемных», простых инцидентов. Первые обусловлены имеющейся проблемой – ошибкой или неэффективным решением при проектировании и могут быть устранены посредством инициирования и реализации ИТ-изменения. Результат распознавания может быть как верным (простой инцидент распознается как простой, а «проблемный» как «проблемный»), а может

быть неверным (простой инцидент распознается как «проблемный», а «проблемный» – как простой). При неверном распознавании реализуется безрезультатный проектный цикл изменения ИТ-актива механизмом  $\mu(J_k)$  обработки проблем, либо разрешение реальной проблемы откладывается, что приводит к увеличению числа повторных «проблемных» инцидентов.

Обозначим через  $\rho(B^\Delta(CI^q))$  процедуру распознавания происшествий  $B^\Delta(CI^q)$ . В соответствии с ранее введенными обозначениями  $I = \bigcup_{k=1}^K I_k$  и  $J = \bigcup_{k=1}^K J_k$  представляют, соответственно, множества инцидентов и проблем, приходящихся на плановый интервал времени  $T$ .

$$\text{Пусть } I_k = I_k^{er}(\rho(B^\Delta(CI^q))) \cup I_k^{cor}(\rho(B^\Delta(CI^q))), \\ J_k = J_k^{er}(\rho(B^\Delta(CI^q))) \cup J_k^{cor}(\rho(B^\Delta(CI^q)))$$

где  $I_k^{er}(\rho(B^\Delta(CI^q)))$ ,  $J_k^{er}(\rho(B^\Delta(CI^q)))$  – множества некорректно распознанных механизмом  $\rho(B^\Delta(CI^q))$  происшествий. Пусть также  $I_k(J_k^{er})$  – количество повторных инцидентов, вызванных некорректным распознаванием проблем, а  $c(\rho(B^\Delta(CI^q)))$  – средние затраты на одно распознавание. Тогда задачу синтеза процедуры распознавания  $\rho(B^\Delta(CI^q))$  происшествий можно сформулировать следующим образом:

$$\sum_{k=1}^K |I_k(J_k^{er}(\rho(B^\Delta(CI^q))))| c(\mu(I_k)) + \\ + \sum_{k=1}^K |I_k^{er}(\rho(B^\Delta(CI^q)))| c(\mu(J_k)) \rightarrow \min_{\rho(B^\Delta(CI^q))} \quad (6) \\ \sum_{k=1}^K \{|I_k| + |J_k|\} c(\rho(B^\Delta(CI^q))) \leq C^*(\rho).$$

То есть требуется создать такую процедуру распознавания ИТ-происшествий, которая минимизирует совокупные затраты на устранение повторных инцидентов и обработку ошибочно признанных проблем при затратах на его функционирование, не превосходящих  $C^*(\rho)$ . При незрелых процессах ИТ-провайдера задачу (6) можно упростить и минимизировать, например, общее количество повторных инцидентов.

*Построение механизма распознавания  $\rho(B^\Delta(CI^q))$  на основе анализа количества повторных инциден-*

*тов.* Для решаемой задачи ограничимся представлением об ИТ-сервисе как о совокупности взаимодействующих конфигурационных элементов (сервисных компонентов):  $S_v = (s_{v1}, s_{v2}, \dots, s_{v\omega}, \dots, s_{v\Omega(v)})$ . Пусть  $B^\Delta(CI^q) = \{b^\Delta(ci_n^q) | n = 1, N^q\}$  произвольное происшествие с конфигурацией  $CI^q$ . На основании информации о происшествии определим множество тех элементов конфигурации  $CI^q$ , в функционировании которых зафиксировано отклонение от нормы:  $\{ci_\psi^q\} = \{ci_\psi^q | b_m^\Delta(ci_\psi^q) \neq 0, \psi \in \Psi(B^\Delta(CI^q)) \subset 1, N^q\}$ . Здесь  $\Psi(B^\Delta(CI^q))$  – множество номеров элементов конфигурации  $CI^q$  с отклонением от нормы. Это множество не пусто, так как  $B^\Delta(CI^q)$  – ИТ-происшествие. Для каждого элемента  $ci_\psi^q$  определим подмножество  $S(ci_\psi^q) = \{s_v(ci_\psi^q) | s_v \in S^c, ci_\psi^q \subset s_{v\omega}, \omega \in \Omega(v)\}$  ИТ-сервисов, входящих в каталог  $S^c$  и содержащих в сервисных активах конфигурационные элементы, зарегистрированные в ИТ-происшествии  $B^\Delta(CI^q)$  как функционирующие с отклонениями от штатного режима. Основу предлагаемого механизма распознавания  $\rho(B^\Delta(CI^q)) = \{\rho(ci_\psi^q) | \psi \in \Psi(B^\Delta(CI^q))\}$  вида происшествия  $B^\Delta(CI^q)$  составляет анализ данных, содержащихся в конфигурационной базе, об инцидентах и проблемах с сервисами, входящими в подмножества  $S(ci_\psi^q)$ ,  $\psi \in \Psi(B^\Delta(CI^q))$ .

Пусть

$$S_\psi(ci_\psi^q) = \{s_1(ci_\psi^q), s_2(ci_\psi^q), \dots, s_n(ci_\psi^q), \dots, s_{n(\psi)}(ci_\psi^q)\}, \\ \psi \in \Psi(B^\Delta(CI^q)).$$

Определим для каждого сервиса  $s_n(ci_\psi^q)$ ,  $n \in 1, n(\psi)$ ,  $\psi \in \Psi(B^\Delta(CI^q))$ , функционал  $\Phi(s_n(ci_\psi^q))$ , значение которого равно количеству инцидентов, зарегистрированных в CMDB и причиной которых стало функционирование конфигурационного элемента  $ci_\psi^q$ . Тогда общее количество  $\Phi(ci_\psi^q)$  инцидентов, вызванных функционированием элемента  $ci_\psi^q$  определяется функционалом  $\Phi(ci_\psi^q) = \sum_{n=1}^{n(\psi)} \Phi(s_n(ci_\psi^q))$ ,  $\psi \in \Psi(B^\Delta(CI^q))$ . Если

$c(\mu(I(ci_\psi^q)))$  – затраты на устранение одного инцидента с элементом  $ci_\psi^q$ , то общие затраты  $C(ci_\psi^q)$  на устранение всех инцидентов с  $ci_\psi^q$  будут

$$C(ci_{\psi}^q) = \sum_{n=1}^{n(\psi)} \varphi(s(ci_{\psi}^q))c(\mu(I(ci_{\psi}^q))). \quad (7)$$

Используем функционалы  $C(ci_{\psi}^q)$ ,  $\psi \in \Psi(B^{\Delta}(CI^q))$ , в качестве критериев для принятия решения о виде происшествия с конфигурационными элементами  $ci_{\psi}^q$ :

$$\rho(ci_{\psi}^q) = \begin{cases} \text{проблема,} & \text{если } C(ci_{\psi}^q) \geq C^*(ci_{\psi}^q) \\ \text{инцидент,} & \text{если } C(ci_{\psi}^q) < C^*(ci_{\psi}^q). \end{cases} \quad (8)$$

Здесь  $C^*(ci_{\psi}^q)$  – пороговый уровень затрат на устранение инцидентов с конфигурационным элементом  $ci_{\psi}^q$ , оправдывающий проектные затраты на реализацию необходимого изменения  $ci_{\psi}^q$ . При незрелых ИТ-процессах в качестве критерия для принятия решения можно использовать функционал  $\Phi(ci_{\psi}^q)$ .

**4. Функциональная структура системы управления ИТ-происшествиями.** На рисунке 2 приведена укрупненная структура системы оперативного управления потоком ИТ-происшествий, в основу которой положены решения выше рассмотренных задач.

Здесь SLAs – соглашения о качестве обслуживания, OLAs – соглашения о качестве операционного взаимодействия, UCs – подписанные контракты, St – система стимулирования за достижение показателей SLAs и OLAs [3]. В систему входят подсистемы: «Service desk», распознавания ИТ-происшествий, обработки событий, инцидентов и проблем. Каждая из подсистем обладает механизмом взаимодействия с внешними системами управления следующими объектами: технической инфраструктурой и приложениями (2-ой уровень поддержки), проектированием и внедрением (3-ий уровень поддержки), внешними поставщиками ИТ-активов (4-ый уровень поддержки). «Service desk» представляет собой 1-ую линию поддержки. Основная цель системы оперативного управления

состоит в устранении ИТ-происшествий с минимальным снижением удовлетворенности клиентов при заданном уровне затрат на ее создание и функционирование.

**Заключение.** В основе классификации и постановок задач синтеза механизмов распознавания и обработки ИТ-происшествий лежат представления о проектно-процессной версионной модели ИТ-сервиса [4], которая предусматривает оптимизацию ИТ-активов, функционирующих со сбоями или неэффективно.

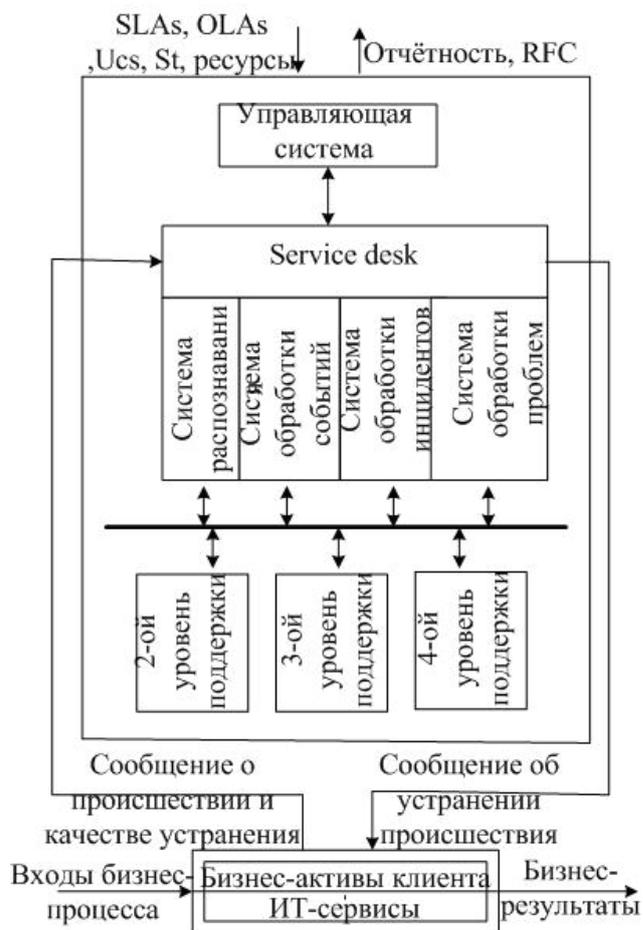


Рис. 2. Структура системы оперативного управления ИТ-происшествиями

### Список литературы

1. OGC-ITIL V3-1- Service Strategy, TSO 2007. -373р.
2. Зимин В.В., Кулаков С.М., Зимин А.В. /Активы ИТ-провайдера как объекты ситуационно-динамического управления // Известия вузов. Черная металлургия, №2 2011. С.47-53.
3. OGC-ITIL V3- 4 – Service Operation TSO 2007. -397р.
4. Зимин В.В., Кулаков С.М., Зимин А.В. /О структуризации системы управления деятельностью ИТ-провайдера Системы управления и информационные технологии, №2.1(48), 2012. – С. 198-202