

УДК 519.876.2

ББК 32.817

О СТРУКТУРИЗАЦИИ СИСТЕМЫ УПРАВЛЕНИЯ ДЕЯТЕЛЬНОСТЬЮ ИТ-ПРОВАЙДЕРА

Зимин В.В.¹, Кулаков С.М.², Зимин А.В.³

*(Сибирский государственный индустриальный университет,
Новокузнецк)*

Предлагаются и обосновываются базовые структуры и компоненты системы управления деятельностью поставщика ИТ-услуг. Декомпозиция системы управления осуществляется на основе представлений о жизненном цикле ИТ-сервиса, на структуризации портфеля сервисов ИТ-провайдера в соответствии со стадиями жизненного цикла и на применении процессного подхода для организации эффективного взаимодействия между синтезированными компонентами системы.

Ключевые слова: ИТ-деятельность, ИТ-сервис, жизненный цикл, принципы специализации и координации, портфель проектов, процесс, структура системы управления, механизм управления.

Структура системы управления портфелем сервисов, предоставляемых ИТ-провайдером клиентам, определяется свойствами сервисов и их пакетов, конкретизацией функций, процессов и механизмов управления и является, как правило, сложной и многоуровневой. Цель настоящей статьи состоит в изложении некоторых подходов, применение которых позволяет синтези-

¹ Зимин Валерий Викторович, кандидат технических наук, доцент (zimn.1945@mail.ru.)

² Кулаков Станислав Матвеевич, доктор технических наук, профессор (kulakov-ais@mail.ru.)

³ Зимин Алексей Валерьевич, аспирант (zab691@rambler.ru).

ровать структуру эффективной системы управления деятельностью ИТ-провайдера.

1. Функциональная структуризация жизненного цикла ИТ-сервиса.

Основой для построения структуры системы управления является результат функциональной декомпозиции деятельности ИТ-провайдера по производству и поставке сервисов. Структура жизненного цикла ИТ-сервиса s_i который является результатом декомпозиции деятельности, приведена на рисунке 1.

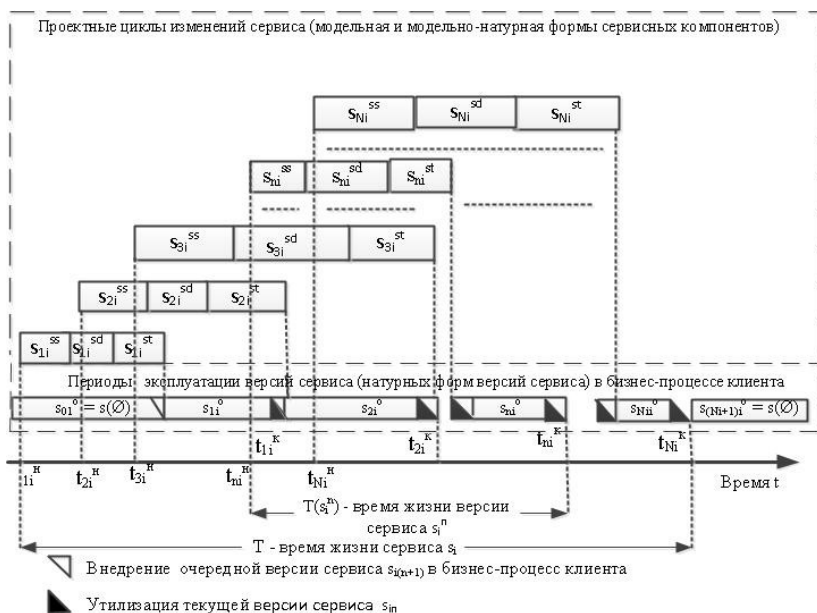


Рисунок 1. Структура жизненного цикла ИТ-сервиса

Здесь и далее используются понятия и обозначения:

$S^p = \{s_i | i \in I\}$ - портфель сервисов, поставляемых клиентам ИТ-провайдером.

$s(\emptyset)$ - «пустой» сервис (все стадии которого являются «пустыми»). Понятие «пустого» сервиса представляет удобство для логического анализа, в частности, позволяет рассматривать создание сервиса (первый проектный цикл - стратегию, проектирование и внедрение) как реализацию проектного цикла изменения относительно «пустого» сервиса $s(\emptyset)$.

s_{in}^0 - n-я версия сервиса s_i (результат n-го цикла изменения сервиса). Версия s_{in}^0 описывает n-ю модификацию сервиса, которую применяет клиент в своем бизнес-процессе.

$s_{ipr}^n = (s_{ni}^{ss}, s_{ni}^{sd}, s_{ni}^{st})$ - n-й проектный цикл изменения сервиса s_i , включающий стадии стратегии (s_{ni}^{ss}), проектирования (s_{ni}^{sd}) и внедрения (s_{ni}^{st}).

$s_i^n = (s_{ni}^{ss}, s_{ni}^{sd}, s_{ni}^{st}, s_{in}^0, s_{ni}^{su})$ - жизненный цикл n-й версии сервиса s_i , где s_{ni}^{su} - стадия утилизации.

$s_i = (s_i^n | n=1, \dots, N_i)$ - жизненный цикл сервиса s_i (упорядоченная последовательность жизненных циклов его версий), где N_i - количество версий сервиса s_i .

$T(s_i^n) = (t_{ni}^H, t_{ni}^K)$ - период жизненного цикла n-ой версии s_{ni} сервиса s_i .

$T = T(s_i)$ - период жизненного цикла сервиса s_i .

Содержание каждой стадии жизненного цикла версии s_i^n сервиса s_i (стратегии s_{ni}^{ss} , проектирования s_{ni}^{sd} , ввода в эксплуатацию s_{ni}^{st} , эксплуатации s_{in}^0 и утилизации s_{ni}^{su}) представляет собой совокупность специализированных функций-работ¹, которые являются типовыми для всех версий s_i^n , $n = 1, N_i$, всех сервисов $s_i \in S^P$. Это обстоятельство составляет одну из основ для синтеза структуры системы управления деятельностью ИТ-провайдера.

¹ Функция {function} - 2) обязанность, круг деятельности, назначение, роль [1].

2. Синтез структуры системы управления портфелем сервисов на основе жизненного цикла сервиса

Состав портфеля сервисов $S^P = \{s_i \mid i \in I\}$ ИТ-провайдера изменяется со временем, (рисунок 1). В каждый момент времени одни сервисы эксплуатируются, другие – разрабатываются с целью последующего ввода в эксплуатацию, третьи - утилизируются, четвертые – обновляются. Чтобы описать структуру портфеля сервисов S^P и показать его динамическую природу, введем следующие понятия.

Под состоянием версии $s_{ni}(t)$ сервиса s_i в момент времени $t \in T(s_{ni})$ будем понимать стадию жизненного цикла версии s_{ni} , выполняемую в момент времени t . Будем считать, что, $s_{ni}(t) = \{\emptyset\}$ для $t \in T(s_{ni})$. Как видно из рисунка 1, версии s_{1i} сервиса s_i в момент времени t_{1i}^H соответствует состояние $s_{1i}(t_{1i}^H) = \{s_{1i}^{ss}\}$. Аналогично, версии s_{Ni} в момент времени t_{Ni}^H соответствует состояние $s_{Ni}(t_{Ni}^H) = \{s_{Ni}^{sd}\}$. Заметим, что оператор $s_{ni}(t)$ представляет собой однозначное отображение – одному моменту времени соответствует одна реализуемая стадия жизненного цикла версии сервиса.

Определим состояние $s_i(t)$ сервиса s_i в момент времени t как совокупность состояний всех его версий в момент t , то есть:

$$(1) \quad s_i(t) = \bigcup_{n=1}^{Ni} s_{ni}(t)$$

В частности, $s_i(t_{ni}^H) = \{s_{1i}^o, s_{2i}^{st}, s_{3i}^{sd}, s_{ni}^{ss}\}$, а $s_i(t_{3i}^H) = \{s(\emptyset), s_{1i}^{st}, s_{2i}^{ss}, s_{3i}^{ss}\} = \{s_{1i}^{st}, s_{2i}^{ss}, s_{3i}^{ss}\}$. Будем также считать также, что для $t \in T(s_i)$ оператор $s_i(t) = \{s(\emptyset)\}$. Так как состояние $s_{ni}(t)$ каждой версии сервиса в каждый момент времени t может принимать только одно из пяти возможных значений $\{s_{ni}^{ss}, s_{ni}^{sd}, s_{ni}^{st}, s_{ni}^{so}, s_{ni}^{su}\}$, то $s_i(t)$ можно представить в виде объединения пяти подмножеств:

$$(2) \quad s_i(t) = \bigcup_{n=1}^{Ni} s_{ni}(t) = (\bigcup_{n \in N_{iss}} s_{ni}(t)) \cup (\bigcup_{n \in N_{isd}} s_{ni}(t)) \cup (\bigcup_{n \in N_{ist}} s_{ni}(t)) \cup (\bigcup_{n \in N_{iso}} s_{ni}(t)) \cup (\bigcup_{n \in N_{isu}} s_{ni}(t)),$$

где совокупность подмножеств $\{N_{iss}, N_{isd}, N_{ist}, N_{iso}, N_{isu}\}$ множества $\{1, \dots, N_i\}$ есть его разбиение (подмножества не пересекаются и их объединение образует исходное множество). Заметим, что для отдельных сервисов отдельные подмножества могут быть пустыми.

Текущее состояние портфеля сервисов $S^P(t) = \{s_i(t) \mid i \in I(t)\}$ естественно определить как совокупность состояний входящих в него сервисов, то есть:

$$(3) \quad S^P(t) = \bigcup_{i \in I} S_i(t) = \bigcup_{i \in I} \bigcup_{n=1}^{N_i} S_{ni}(t).$$

С учетом (2) получаем:

$$(4) \quad S^P(t) = \left(\bigcup_{i \in I} \bigcup_{n \in N_{iss}} S_{ni}(t) \right) \cup \left(\bigcup_{i \in I} \bigcup_{n \in N_{isd}} S_{ni}(t) \right) \cup \left(\bigcup_{i \in I} \bigcup_{n \in N_{ist}} S_{ni}(t) \right) \cup \left(\bigcup_{i \in I} \bigcup_{n \in N_{iso}} S_{ni}(t) \right) \cup \left(\bigcup_{i \in I} \bigcup_{n \in N_{isu}} S_{ni}(t) \right).$$

Следует заметить, что множество $I = \{i\}$ также зависит от времени, то есть $I = I(t)$, так как регулярно одни версии сервисов создаются, а другие - утилизируются. В соответствии с (4) портфель сервисов представлен в виде объединения пяти подмножеств. В первое подмножество входят версии сервисов, состояние (статус) которых в момент времени t соответствует стадии стратегии (s^{ss}), во второе – состояние которых соответствует стадии проектирования (s^{sd}), в третьей - стадии внедрения (s^{st}), в четвертой - стадии эксплуатации (s^{so}), в пятой - стадии утилизации (s^{su}). Традиционно [4] структуру портфеля сервисов $S^P(t)$ представляют в виде двух подмножеств:

$$(5) \quad S^P(t) = S^C \cup S^D,$$

где S^C - каталог эксплуатируемых сервисов, статус которых равен s^{so} (C – catalog) и инвестиционный портфель S^D (разрабатываемые сервисы, D – development), рисунок 2.

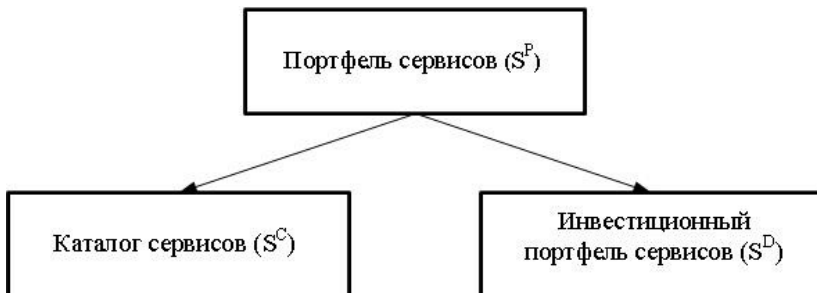


Рисунок 2. Структура портфеля сервисов S^P

Первые три подмножества из (4) относятся, очевидно, к S^D , четвертое – к S^C . Квалификация пятого подмножества ($i \in I, n \in N_{isu} \cdot S_{ni}(t)$) требует комментария. Утилизация сервисных активов выполняется (рисунок 1) в двух случаях:

1. При выводе текущей версии s_{ni} сервиса из эксплуатации и замене её следующей версией $s_{(n+1)i}$. В этом случае сервисные активы текущей версии утилизируются, а очередной (новой) – вводятся в эксплуатацию.

2. При выводе сервиса s_i из эксплуатации, то есть при выводе из эксплуатации его последней версии s_{Nii} . В этом случае утилизируются сервисные активы последней версии.

Первый случай соответствует составу работ, выполняемых на стадии внедрения s^{st} , второй – собственно утилизации сервиса s_i . С учетом сделанных замечаний целесообразно представлять текущее состояние портфеля сервисов $S^P(t)$ в виде объединения трех подмножеств:

$$(6) \quad S^P(t) = S^C(t) \cup S^D(t) \cup S^U(t), \text{ где}$$

$$(7) \quad S^C(t) = \{ i \in I, n \in N_{iso} \cdot S_{ni}(t) \},$$

$$(8) \quad S^D(t) = \{ i \in I, n \in N_{iss} \cdot S_{ni}(t) \} \cup \{ i \in I, n \in N_{isd} \cdot S_{ni}(t) \} \cup \{ i \in I, n \in N_{ist} \cdot S_{ni}(t) \} \cup \{ i \in I, n \in (N_{isu} \setminus N_i) \cdot S_{ni}(t) \}.$$

$$(9) \quad S^U(t) = \{ i \in I \cdot S_{Ni}(t) \}.$$

В учетом сделанных замечаний преобразуем соотношение (4):

$$(10) S^P(t) = (\bigcup_{i \in I, n \in N_{iss}} S_{ni}(t)) \cup (\bigcup_{i \in I, n \in N_{isd}} S_{ni}(t)) \cup ((\bigcup_{i \in I, n \in N_{ist}} S_{ni}(t)) \cup (\bigcup_{i \in I, n \in (N_{isu} \setminus N_i)} S_{ni}(t))) \cup (\bigcup_{i \in I, n \in N_{iso}} S_{ni}(t)) \cup (\bigcup_{i \in I} S_{nii}(t)).$$

Введем обозначения:

$$(11) S^{Pss} = \bigcup_{i \in I, n \in N_{iss}} S_{ni}(t),$$

$$(12) S^{Psd} = \bigcup_{i \in I, n \in N_{isd}} S_{ni}(t),$$

$$(13) S^{Pst} = (\bigcup_{i \in I, n \in N_{ist}} S_{ni}(t)) \cup (\bigcup_{i \in I, n \in (N_{isu} \setminus N_i)} S_{ni}(t)),$$

$$(14) S^{Pso} = \bigcup_{i \in I, n \in N_{iso}} S_{ni}(t),$$

$$(15) S^{Psu} = \bigcup_{i \in I} S_{nii}(t).$$

Тогда:

$$(16) S^P(t) = S^{Pss}(t) \cup S^{Psd}(t) \cup S^{Pst}(t) \cup S^{Pso}(t) \cup S^{Psu}(t).$$

В (16) текущий портфель сервисов $S^P(t)$ представлен в виде объединения пяти подмножеств соответственно стадиям жизненного цикла версий сервисов. В частности, подмножество $S^{Pss}(t)$ описывает совокупность версий сервисов портфеля, статус которых в момент времени t соответствует стадии стратегии (s^{ss}), подмножество $S^{Psd}(t)$ - совокупность версий сервисов портфеля, статус которых в момент времени t соответствует стадии проектирования (s^{sd}), подмножество $S^{Pst}(t)$ - совокупность версий сервисов портфеля, статус которых в момент времени t соответствует стадии внедрения (s^{st}), подмножество $S^{Pso}(t)$ - совокупность версий сервисов портфеля, статус которых в момент времени t соответствует стадии эксплуатации (s^{so}), подмножество $S^{Psu}(t)$ - совокупность версий сервисов портфеля, статус которых в момент времени t соответствует стадии утилизации (s^{su}).

Опираясь на принцип специализации разумно принять, что каждое из сформированных подмножеств портфеля сервисов $S^P(t)$ является объектом управления отдельной подсистемы управления. Обозначим эти подсистемы управления через $СУ(S^{Pss}(t))$, $СУ(S^{Psd}(t))$, $СУ(S^{Pst}(t))$, $СУ(S^{Pso}(t))$, $СУ(S^{Psu}(t))$.

На рисунке 3 представлена укрупненная структура системы управления портфелем сервисов $S^P(t)$. Тонкими стрелками показаны прямые горизонтальные связи между подсистемами, которые отражают их взаимодействие при производстве, эксплуата-

ции и утилизации ИТ-сервисов. В частности подсистема $CY(S^{Pss}(t))$ формирует для других подсистем подмножество $S^{Psd}(t)$, $S^{Pst}(t)$, $S^{Pso}(t)$, $S^{Psu}(t)$, соответствующие бюджеты, совокупность $SLR(s_i)$ требований к полезности и применимости всех сервисов, а также совокупность $SAC(s_i)$ критериев приемки сервисов в эксплуатацию. Все стадии жизненных циклов версий сервисов имеют сильные технологические связи, поэтому каждая предшествующая (в соответствии с жизненным циклом сервиса) подсистема управления имеет прямые связи с каждой последующей подсистемой.

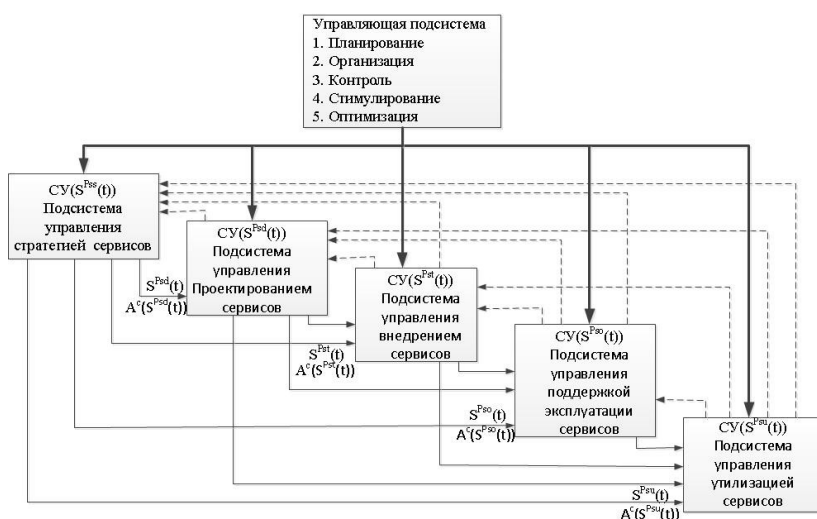


Рисунок 3. Укрупненная структура системы управления портфелем сервисов $S^P(t)$

Обратные связи между подсистемами показаны пунктирными стрелками. Они антисимметричны прямым связям – каждая последующая по технологии производства сервисов подсистема управления имеет обратную связь с каждой предыдущей, чтобы иметь возможность инициировать необходимые оптимизационные изменения для подсистем, формирующих управляющие и входные воздействия. Управляющая подсистема, поми-

мо выполнения стандартного набора функций управления (цикл Файоля [2]) - планирования, организации, контроля и стимулирования, реализует динамическую оптимизацию механизмов функционирования, взаимодействий и управления для подсистем $СУ(S^{Pss}(t))$, $СУ(S^{Psd}(t))$, $СУ(S^{Pst}(t))$, $СУ(S^{Pso}(t))$, $СУ(S^{Psu}(t))$.

В заключение этого раздела заметим, что предлагаемая структура системы управления динамическим портфелем сервисов $S^P(t)$ имеет четко выраженную проектную ориентированность, обусловленную следующими аспектами:

- проектной ориентированности подсистем $СУ(S^{Pss}(t))$, $СУ(S^{Psd}(t))$, $СУ(S^{Pst}(t))$, которые представляют собой фазы проекта [3];

- наличием системы динамической оптимизации, задача которой состоит в идентификации неэффективных механизмов в изменяющихся внутренних и внешних условиях, формировании портфеля проектов сервисных улучшений, инициации и контроле выполнения этих проектов.

3. Процессная внутри- и межстадийная структуризация ИТ-деятельности

Процессный подход является инструментом координации¹. Применение процессов² позволяет повысить эффективность взаимодействия, преимущественно иерархически организованных, функционально-специализированных структурных компонентов.

Задача координации функционирования специализированных подсистем $СУ(S^{Pss}(t))$, $СУ(S^{Psd}(t))$, $СУ(S^{Pst}(t))$, $СУ(S^{Pso}(t))$, $СУ(S^{Psu}(t))$ состоит в синтезе рентабельной совокупности $A^c(t) = \{a_j^c(t) | j \in J(t)\}$ интегрированных процессов $a_j^c(t)$, обеспечиваю-

¹ Координация — (от лат. со совместно и ordinatio упорядочение) взаимосвязь, согласование, приведение в соответствие. БЭС (2000).

² Процесс (activity) – устойчивая, целенаправленная совокупность взаимосвязанных функций-операций, преобразующая по определенной технологии входы в выходы, представляющие ценность для потребителя [3].

щих поставку клиентам ИТ-сервисов приемлемого качества, стоимости и рисков применения.

Формально процесс a_j^c описывается упорядоченной последовательностью функций-операций:

$$(17) \quad a_j^c = (a_{mj}^c \mid m \in M_j)$$

Типовая структура системы управления процессом для известной в менеджменте модели процесса [4] приведена на рисунке 4.

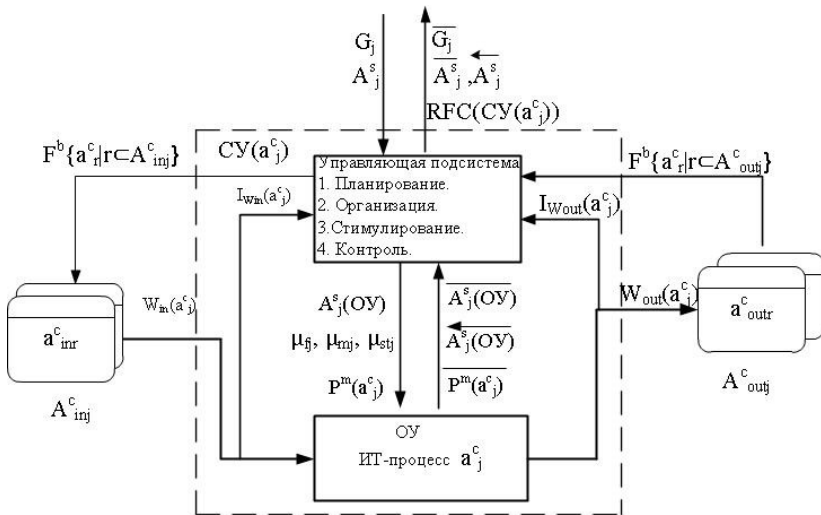


Рисунок 4. Структура системы управления отдельным ИТ-процессом

Здесь:

- $A_{inj}^c = \{a_r^c \mid r \in (J_j), W_{in}(a_r^c) \neq \emptyset\}$ – подмножество процессов, формирующих входы процесса a_j^c ;

- $A_{outj}^c = \{a_r^c \mid r \in (J_j), W_{out}(a_r^c) \neq \emptyset\}$ - подмножество процессов, для которых процесс a_j^c , формирует выходы;

- $W_{in}(a_j^c)$ и $W_{out}(a_j^c)$ - входы и выходы процесса a_j^c соответственно;

- $I_{Win}(a_j^c)$ и $I_{Wout}(a_j^c)$ – информация о входах и выходах соответственно;

- $G_j(a_j^c) = \{g_{jp}(a_j^c) | p \in P_j\}$ – цели процесса a_j^c ;

- $A_j^s = \{a_{jl}^s | l \in L_j\}$, $A_j^s = \{a_{jl}^s | l \in L_j\}$ ИТ-активы, запрашиваемые и выделяемые системе управления $CY(a_j^c)$;

- $A_j^s(OY) = \{a_{jl}^s(OY) | l \in L_j\}$, $A_j^s(OY) = \{a_{jl}^s(OY) | l \in L_j\}$ – ИТ-активы, запрашиваемые и выделяемые для выполнения процесса a_j^c ;

- $\mu_{ij}(a_j^c)$, $\mu_{mj}(a_j^c)$, $\mu_{stj}(a_j^c)$, – механизмы функционирования, управления и стимулирования соответственно [2];

- $P^m(a_j^c) = \{p_d^m(a_j^c) | d \in D_j\}$ – показатели эффективности процесса a_j^c ;

- $F^b(a_r^c | r \subset A_{inj}^c)$, $F^b(a_r^c | r \subset A_{outj}^c)$ – обратные связи;

- $RFC(CY(a_j^c)) = (rfc_h(CY(a_j^c)) | h \in H_j)$ – запросы на изменение (проекты), инициируемые $CY(a_j^c)$;

Черта сверху означает информацию, сообщаемую вышестоящему уровню управления.

Управляющими воздействиями для процесса a_j^c , вырабатываемыми вышестоящим уровнем управления, являются цели процесса $G_j(a_j^c) = \{g_{jp}(a_j^c) | p \in P_j\}$ и ИТ-активы $A_j^s = \{a_{jl}^s | l \in L_j\}$, выделяемые для его функционирования. Система управления $CY(a_j^c)$ процессом не только отчитывается перед вышестоящим уровнем о достижении целей и издержках, но и инициирует запросы на реализацию проектов $RFC(a_j^c) = (rfc_h(a_j^c) | h \in H_j)$ по изменению как собственных механизмов функционирования и управления, так и по изменению процессов $A_{inj}^c = \{a_r^c | r \in (J_j), W_{in}(a_r^c) \neq \emptyset\}$, поставляющих для него входы. Важными особенностями системы управления процессом являются постоянный контроль показателей качества функционирования процесса $P^m(a_j^c) = \{p_d^m(a_j^c) | d \in D_j\}$ - показателей качества реализации процесса, его результата, показателей удовлетворенности заказчика процессом и продуктом процесса.

На рисунке 5 приведена структура системы управления, которая является типовой для подсистем $CY(S^{Pss}(t))$, $CY(S^{Psd}(t))$, $CY(S^{Pst}(t))$, $CY(S^{Pso}(t))$, $CY(S^{Psu}(t))$. Обозначим типовую подсистему через $CY(S^{Pph}(t))$. Объектом управления в типовой системе

является совокупность $A^c(S^{Pph}(t) = \{a_j^c(t) | j \in J^{ph}\}$ взаимодействующих ИТ-процессов $a_j^c(t)$, каждым из которых периодически решается конкретная задача рассматриваемой стадии жизненного цикла сервисов.

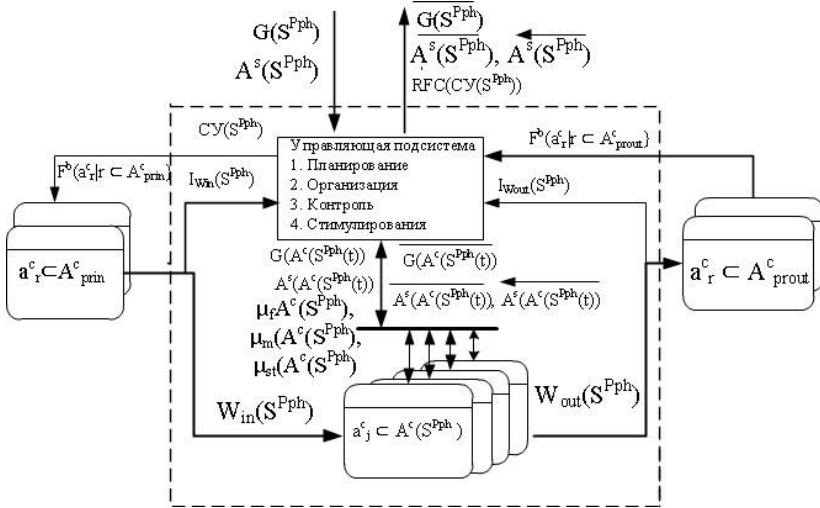


Рисунок 5. Принципиальная структура подсистемы управления $CY(S^{Pph}(t))$ стадией жизненных циклов портфеля сервисов

Обозначения рисунка 5:

- $G(S^{Pph}(t)) = \{g_p(S^{Pph}(t)) | p \in P(S^{Pph}(t))\}$ – совокупность целей, устанавливаемых системой управления $CY(S^P(t))$ портфелем сервисов для подсистемы $CY(S^{Pph}(t))$;

- $A^s(S^{Pph}(t)) = \{a_l^s(S^{Pph}(t)) | l \in L(S^{Pph}(t))\}, A^s(S^{Pph}(t)) = \{a_l^s(S^{Pph}(t)) | l \in L(S^{Pph}(t))\}$ – активы (ресурсы и возможности), запрашиваемые и выделяемые системой $CY(S^P(t))$ для функционирования подсистемы $CY(S^{Pph}(t))$;

- $A^c_{prin}(t) = \{a_r^c | a_r^c \in (A^c(t) \setminus A^c(S^{Pph}(t))), W_{in}(a_r^c) \neq \emptyset\}$ – подмножество процессов других подсистем, формирующих входы для подсистемы $CY(S^{Pph}(t))$;

- $A^c_{\text{prout}}(t) = \{a^c_r \mid a^c_r \subset (A_c(t) \setminus A^c(S^{\text{Pph}}(t)), W_{\text{out}}(a^c_r) \neq \emptyset\}$ - подмножество процессов других подсистем, для которых подсистема $CY(S^{\text{Pph}}(t))$ формирует входы;
- $W_{\text{in}}(S^{\text{Pph}}(t)), W_{\text{ure}}(S^{\text{Pph}}(t))$ – входы и выходы подсистемы $CY(S^{\text{Pph}}(t))$;
- $I_{\text{Win}}(S^{\text{Pph}}(t))$ и $I_{\text{Wout}}(S^{\text{Pph}}(t))$ – информация о входах и выходах соответственно;
- $G(A^c(S^{\text{Pph}}(t)))$ – цели, устанавливаемые для процессов подсистемы $CY(S^{\text{Pph}}(t))$;
- $\overleftarrow{A^s(A^c(S^{\text{Pph}}(t)))}, A^s(A^c(S^{\text{Pph}}(t)))$ – ИТ-активы, запрашиваемые и выделяемые для выполнения процессов подсистемы $CY(S^{\text{Pph}}(t))$;
- $F^b(a^c_r \mid r \subset A^c_{\text{prin}}(t)), F^b(a^c_r \mid r \subset A^c_{\text{prout}}(t))$ – обратные связи;
- $\mu_f(A^c(S^{\text{Pph}}(t))), \mu_m(A^c(S^{\text{Pph}}(t))), \mu_{st}(A^c(S^{\text{Pph}}(t)))$ – механизмы соответственно функционирования, управления и стимулирования процессов подсистемы $CY(S^{\text{Pph}}(t))$ и их взаимодействия;
- $RFC(CY(S^{\text{Pph}}(t))) = (rfc_h(CY(S^{\text{Pph}}(t)))) \mid h \in H_j$ – запросы на изменения (проекты), инициируемые подсистемой $CY(S^{\text{Pph}}(t))$.

Черта сверху означает информацию, сообщаемую вышестоящему уровню управления.

Конкретизация типовой структуры для подсистем $CY(S^{\text{Pss}}(t)), CY(S^{\text{Psd}}(t)), CY(S^{\text{Pst}}(t)), CY(S^{\text{Pso}}(t)), CY(S^{\text{Psu}}(t))$ может быть выполнена после решения задач синтеза совокупностей процессов $A^c(S^{\text{Pss}}(t)) = \{a^c_j(t) \mid j \in J^{\text{ss}}\}$, $A^c(S^{\text{Psd}}(t)) = \{a^c_j(t) \mid j \in J^{\text{sd}}\}$, $A^c(S^{\text{Pst}}(t)) = \{a^c_j(t) \mid j \in J^{\text{st}}\}$, $A^c(S^{\text{Pso}}(t)) = \{a^c_j(t) \mid j \in J^{\text{so}}\}$, $A^c(S^{\text{Psu}}(t)) = \{a^c_j(t) \mid j \in J^{\text{su}}\}$, обеспечивающих рентабельную поставку сервисов клиентам.

Для представления о возможном составе процессов, применяемых на различных стадиях жизненного цикла сервиса приведем краткое описание процессов, предлагаемых в библиотеке ITIL (версия 3) для подсистем $CY(S^{\text{Pss}}(t)), CY(S^{\text{Psd}}(t)), CY(S^{\text{Pst}}(t)), CY(S^{\text{Pso}}(t)), CY(S^{\text{Psu}}(t))$, которые синтезированы на основе обобщения «лучших практик».

Подсистема управления стратегиями сервисов $CY(S^{\text{Pss}}(t))$ обеспечивает разработку стратегии ИТ-провайдера стратегии для версий сервисов $s_{ni}(t) \in S^{\text{Pss}} = \{i \in I, n \in N_{iss}, s_{ni}(t) \text{ и включает совокупность процессов } A^c(S^{\text{Pss}}(t)) = \{a^c_j(t) \mid j \in J^{\text{ss}}\}$

1. Процесс a^c_{STR} формирования стратегии ИТ-провайдера и приведения её в соответствие с изменяющимися бизнес-стратегиями его клиентов.

2. Процесс $a^c(S^P(t))$ формирования рентабельного портфеля сервисов $S^P(t)$, соответствующего стратегиям клиентов и ИТ-провайдера.

3. Процесс $a^c_{slr}(S^P(t))$ формирования требований $slr_h(s_{ni}) \in SLR(s_{ni})$ к показателям полезности и применимости сервисов и критериев $sic_k(s_{ni}) \in SIC(s_{ni})$ приемки сервисов в эксплуатацию.

4. Процесс $a^c_{ph}(S^P(t))$ формирования и актуализации совокупностей $S^{Pss}(t)$, $S^{Psd}(t)$, $S^{Pst}(t)$, $S^{Pso}(t)$, $S^{Psu}(t)$ версий сервисов и ИТ-активов $A^s(A^c(S^{Pss}(t)))$, $A^s(A^c(S^{Psd}(t)))$, $A^s(A^c(S^{Pst}(t)))$, $A^s(A^c(S^{Pso}(t)))$, $A^s(A^c(S^{Psu}(t)))$, выделяемых для функционирования подсистем $CY(S^{Pss}(t))$, $CY(S^{Psd}(t))$, $CY(S^{Pst}(t))$, $CY(S^{Pso}(t))$, $CY(S^{Psu}(t))$.

Подсистема управления проектированием сервисов $CY(S^{Psd}(t))$ осуществляет проектирование версий сервисов $s_{ni}(t) \in S^{Psd} = \bigcup_{i \in I} \bigcup_{n \in Nisd} s_{ni}(t)$ и включает совокупность процессов $A^c(S^{Psd}(t)) = \{a^c_j(t) | j \in J^{sd}\}$:

1. Проектирование и автоматизация процесса $a^c_{ph}(S^P(t))$ для подсистемы управления стратегией сервисов.

2. Проектирование и автоматизация процесса $a^c_{slm}(S^P(t))$ управления качеством обслуживания клиентов.

3. Проектирование с целью удовлетворения требований $slr_h(s_{ni}) \in SLR(s_{ni})$ к версиям сервисов:

- проектирование доступности $a_v(s_{ni})$ версии сервиса s_{ni} ;
- проектирование мощности $c_p(s_{ni})$ версии сервиса s_{ni} ;
- проектирование непрерывности $c_n(s_{ni})$ версии сервиса s_{ni} ;
- проектирование безопасности $s_c(s_{ni})$ версии сервиса s_{ni} ;

4. Проектирование и автоматизация управления поставщиками $v^n \in V^n(S^P(t))$ сервисов и их компонентов.

Подсистема управления внедрением сервисов $CY(S^{Pst}(t))$ включает совокупность процессов $A^c(S^{Pst}(t)) = \{a^c_j(t) | j \in J^{st}\}$, обеспечивающих ввод версий сервисов $s_{ni}(t) \in S^{Pst} = (\bigcup_{i \in I} \bigcup_{n \in Nist} s_{ni}(t)) \cup (\bigcup_{i \in I} \bigcup_{n \in Nisu \setminus Ni} s_{ni}(t))$ в эксплуатацию. В состав $A^c(S^{Pst}(t))$ входит четыре процесса.

1. Процесс a_{ch}^c обработки изменений $r_{fc_h} \in (RFC(CY(S^{Pss}(t))) \cup RFC(CY(S^{Psd}(t))) \cup RFC(CY(S^{Pst}(t))) \cup RFC(CY(S^{Pso}(t))) \cup RFC(CY(S^{Psu}(t))))$, инициированных процессами всех подсистем управления системы $CY(S^P(t))$.

2. Процесс a_{cf}^c формирования, учета и верификации конфигураций активов ИТ-провайдера.

3. Процесс a_{tr}^c формирования и тестирования релизов на основе сервисных компонентов, подготовленных подсистемой проектирования $CY(S^{Psd}(t))$ и развертывания версий сервисов $s_{ni}(t)$ в бизнес-среде.

4. Процесс a_{knw}^c накопления и использования ИТ-знаний.

Подсистема управления поддержкой эксплуатации сервисов $CY(S^{Pso}(t))$ осуществляет поддержку клиентов при применении ими версий сервисов $s_{ni}(t) \in S^{Pso} = \{i \in I, n \in N_{iso}, s_{ni}(t) \text{ и включает совокупность процессов } A^c(S^{Pso}(t) = \{a_j^c(t) | j \in J^{so}\}:$

1. Процесс a_{ivn}^c обработки событий.

2. Процесс a_{in}^c обработки инцидентов.

3. Процесс a_{pb}^c обработки «проблем».

4. Процесс a_{sr}^c обработки запросов на обслуживание.

5. Процесс a_{slm}^c обеспечения качества обслуживания в соответствии со SLA¹.

Литература

1. НОВИКОВ Д.А. *Теория управления организационными системами*. М.: МПСИ, 2005. - 584 с.
2. *Механизмы управления: учебное пособие* / под редакцией Д.А. НОВИКОВА. -М.: ЛЕНАНД, 2011. – 192 с.
3. РЕПИН В.В., ЕЛИФЕРОВ В.Г. *Процессный подход к управлению* / В.В. РЕПИН, В.Г. ЕЛИФЕРОВ. - РИА «Стандарты и качество», 2005, - 404с.
4. OGC-ITIL V3- 6 – Service Lifecycle – Introduction ITIL TSO 2007. -173p.

¹ SLA – соглашение о качестве обслуживания [4].

ABOUT STRUCTURIZATION OF THE CONTROL SYSTEM BY ACTIVITY OF THE IT PROVIDER

Zimin Valeriy Victorovich, assistant professor (zimin.1945@mail.ru.)

Kulakov Stanislav Matveevich, Doctor of Science, professor (Kulakov-ais@mail.ru.)

Zimin Aleksey, post-graduate student (zab691@rambler.ru).

Предлагаются и обосновываются базовые структуры и компоненты системы управления деятельностью поставщика ИТ-услуг. Декомпозиция системы управления осуществляется на основе авторского представления о жизненном цикле ИТ-сервиса, на структуризации портфеля сервисов ИТ-провайдера в соответствии со стадиями жизненного цикла и на применении процессного подхода для организации эффективного взаимодействия между синтезированными компонентами системы.

Base structures and components of a control system are offered and proved by activity of the supplier of IT Services. Control system decomposition is carried out on the basis of author's representation about IT Service life cycle, on structurization of a portfolio of services of the IT Provider according to stages of life cycle and on application of the process approach for the organisation of effective interaction between the synthesised components of system.

Keywords: IT Activity, IT Service, life cycle, specialisation and coordination principles, function, process, control system structure, the management mechanism.