Models and mechanisms for managing the deployment of IT service releases in the operational environment

A V Zimin¹, I V Burkova² and V V Zimin¹

¹Siberian State Industrial University, 42 Kirova str., Novokuznetsk, 654007, Russia
²V.A. Trapeznikov Institute of Control Sciences of Russian Academy of Sciences, 65 Profsoyuznaya str., Moscow, 117997, Russia

E-mail: zimin.0169@yandex.ru

Abstract. The article discusses the tasks of forming and scheduling the deployment of IT service releases, the settings of which are based on the data about the structure of services stored in the configuration database (CMDB) of the service provider and that help to estimate the number of IT service failures caused by violations in the state of the operational environment due to release deployment. Three sequentially solved problems are considered: the problem of determining the number of releases, the task of determining the releases composition and the task of determining the schedule for deploying releases. The proposed mechanism for determining the number of releases minimizes the deployment costs and limits the failure rate of IT services by the capacity of a channel servicing the failures. The problem of forming the releases composition is formulated and solved as a problem of stones. The proposed mechanism for determining the release deployment schedule uses the discrete version of the local optimization method and is aimed at minimizing the average failure service time while limiting the “lost time” (time duration of incomplete load of the service channel, which increases the total processing time of all failures).

1. Introduction
The introduction of IT services is associated with the risk of erroneous changes in the “basic” state of the operational IT environment when embedding IT assets (configuration elements) necessary for the functioning of new or changed services [1,2,3]. The design service registers the necessary changes in the IT environment in the form of specially designed deployment requests, which describe the IT assets to be updated. With a significant number of renewable assets, the risk of disruptions in the operational environment becomes significant. To reduce the risk, the number of configuration elements embedded in the IT environment at one time is limited, that is, a set of requests for updating assets are broken into subsets (releases). Let us consider the tasks of creating and scheduling the deployment of IT service releases, the settings of which are based on the fact that the information from the configuration management database of the IT service provider (CMDB) allows the number of IT services, that will become unavailable to users due to the deployment of planned IT assets, to be estimated.

2. Estimation of the number of service failures due to incorrect deployment of IT assets
Let \( A = \{A_i | i=1,n\} \) denote the set of received requests for updating the IT environment, where \( A_i = \{a_{ij} | j=1,n\} \) and \( a_{ij} \) – the updated configuration elements. We assume that each \( a_{ij} \) belongs to one
of three types: type h (hard – equipment), type s (soft – software), type u (user – associated with user actions). Then:

\[
A_i = A_h \cup A_s \cup A_u,
\]

\[
A_h = \{a_{hij}^j | j=1,n_i^h\}, \quad A_s = \{a_{sij}^j | j=1,n_i^s\}, \quad A_u = \{a_{uij}^j | j=1,n_i^u\}
\]

\[
(1)
\]

\[
n_i = n_i^h + n_i^s + n_i^u.
\]

Let \(oS = \{s\}\) be the set of operated IT services, and \(A(s)\) – the set of configuration elements of an individual service \(s\). We will reveal the dependence of the number of failures of the operated services on the incorrect deployment of the assets of a single request, figure 1.

**Figure 1.** Connections of deployment request assets with the assets of IT services in operation.

Let \(S(a_h^i), S(a_s^i), S(a_u^i)\) denote the sets of the services, the functioning of which can be influenced by the deployment of assets \(a_{h}, a_{s}, a_{u}\). By definition

\[
S(a_h^i) = \{s(a_h^i) | A(s) \cap a_h^i \neq \emptyset, s \in S_o\} = \{s_i(a_h^i) | l=1, L(a_h^i)\},
\]

\[
S(a_s^i) = \{s(a_s^i) | A(s) \cap a_s^i \neq \emptyset, s \in S_o\} = \{s_i(a_s^i) | l=1, L(a_s^i)\},
\]

\[
S(a_u^i) = \{s(a_u^i) | A(s) \cap a_u^i \neq \emptyset, s \in S_o\} = \{s_i(a_u^i) | l=1, L(a_u^i)\}.
\]

(2)

In order to estimate the number of service failures, it is necessary to know the probabilities of incorrect deployment of various IT assets. The estimation of the values of these probabilities can be performed on the basis of the data from the configuration management database and the data of the “problems” management process (“problem” is a service failure due to an error in its assets). For each registered “problem”, the process assigns a category value (asset type) in which the service failure
took place due to an error. Thus, the configuration management database contains information about the total number of problems registered and resolved by the process, indicating which of them belong to each category (h, s, u). Let \( P(a^h) \), \( P(a^s) \), \( P(a^u) \) denote the relative shares of problems of the corresponding category taken as estimates for the probabilities of incorrect deployment of assets of various types. The average number of expected service failures due to incorrect deployment of asset of the sets \( A^h, A^s, A^u \), respectively, will be:

\[
m(A^h) = \frac{P(a^h)}{P(a^h)} \sum_j L(a^h_j),
\]

\[
m(A^s) = \frac{P(a^s)}{P(a^s)} \sum_j L(a^s_j),
\]

\[
m(A^u) = \frac{P(a^u)}{P(a^u)} \sum_j L(a^u_j).
\]  

(3)

The predicted numbers \( m(A) \) and \( m(A) \) of average number of service failures as a result of the deployment of the assets of a single request \( A_i \) and, accordingly, of the entire set of requests \( A = \{A_i | a = 1, n \} \) will be determined by the relations:

\[
m(A) = m(A^h) + m(A^s) + m(A^u).
\]  

(4)

\[
m(A) = \sum_i m(A_i).
\]  

(5)

According to the definition of a release (a set of IT assets of services embedded in the IT environment at a time), the set of solutions to the problem of generating releases is the set of all possible partitions of the set \( A^h = \{A_i | a = 1, n \} \) (the number of such partitions is 2n). Each partition \( \{A^k | k = 1, k(A) \} \) has the following properties: \( A^k = \bigcup_{i=1}^{k(A)} A_i \) - k-th release, \( k(A) \) - number of releases, \( \sum_{k=1}^{k(A)} n_k = n \), \( A^k \cap A^{k'} = \emptyset \) for \( k \neq k' \), \( \bigcup_{k=1}^{k(A)} A^k = A \) and \( k(A) \leq n \). By analogy with (5) let \( m(A^k) = \sum_i m(A_i) \) denote the number of service failures caused by release deployment \( A^k \).

### 3. The mechanism for determining the number of releases

Let us assume that all service failures caused by releases deployment are detected by users on the first day after deployment. Then the variable \( m(A^k) \) is the average number of failures (failure rate) due to the deployment of release assets \( A^k \) on the corresponding day. Let \( \Delta t \) denote the number of business days between two consecutive release deployments. The release deployment schedule, which determines the size, is formed in agreement with the design, operational services and users. Let \( \beta \) be the failure processing rate by the service channel (the average number of processed service failures per day). In order the channel could cope in \( \Delta t \) days with the flow of service failures, conditioned by the release deployment \( A^k \), the following conditions must be met:

\[
\frac{m(A^k)}{\Delta t} \leq \beta, \quad k = 1, k(A).
\]  

(6)

The release deployment process is costly and requires coordination of many participants. Therefore, the number of releases \( k(A) \) should be the minimum possible; however, it must ensure the fulfillment of inequalities (6). Obviously, in the inequality:

\[
\frac{m(A)}{\Delta t} \leq \beta
\]  

(7)
all requests $A_i, j=1,n,$ can be combined into a single release $A=\bigcup_{i=1}^{n} A_i$, and deployed at a time. If (7) is not satisfied, then the minimum number of releases for the set of deployment requests $\{A^k|k=1,k(A)\}$ is defined as the maximum variable $k(A)$ for which the inequality is satisfied

$$\frac{m(A)}{k(A)} \leq \beta.$$  

(8)

The variable $\frac{m(A)}{k(A)}$ characterizes the average number of failures per release per time interval $\Delta t$. Since $\beta=\text{const}$ and $\Delta t=\text{const}$, condition (8) requires the fulfillment of the inequalities

$$m(A^k) = \frac{m(A)}{k(A)} = \text{const}, \quad k=1,k(A).$$  

(9)

4. The formation mechanism of the releases composition (content)

Relations (9) make it possible to formulate and solve the problem of forming the composition of releases as a “stone problem”, defining the variables $m(A_i),i=1,n$ as the “stone weight”. Let us introduce the variable $x_k$:

$$x_k = \begin{cases} 1, & A_i \in A^k \\ 0, & A_i \notin A_k \end{cases}.$$  

(10)

Then the task of forming the releases composition takes the form of a “stone problem”:

$$\max_{k=1,k(A)} m(A^k) = \sum_{i=1}^{n} m(A_i)x_k \rightarrow \min \quad \sum_{i=1}^{n} x_k = 1, \quad k=1,k(A).$$  

(11)

(12)

Applying one of the well-known algorithms solving the “stone problem” for a set of requests, we obtain a solution $\{A^k|k=1,k(A)\}$ where $A^k=\bigcup_{i=1}^{n} A_i$ is the k-th release, and $\{m(A^k)|k=1,k(A)\}$ is the set of “weights” of the corresponding releases.

5. Mechanism of scheduling the releases deployment

The variable $m(A^k)$ describes the number of service failures caused by the deployment of release $A^k$. Dividing this variable by the intensity of the service channel $\beta$ (average number of failures serviced per unit time) we obtain the time interval $\tau_k=\tau(A^k)$ required to restore the availability of IT services that failed as a result of deployment of the k-th release:

$$\tau(A^k) = \frac{m(A^k)}{\beta}.$$  

(13)

If there $\tau_k$ is less than time interval $\Delta t$ between two consecutive release deployments ($\tau_k<\Delta t$), then the service channel will have a “simple” duration ($\Delta t-\tau_k$) in the interval $\Delta t$ (“lost time” for servicing). Because

$$\sum_{k=1}^{k(A)} \tau_k = \frac{m(A^k)}{\beta} = \frac{m(A)}{\beta} \approx k(A)\Delta t,$$  

(14)
then the total time for servicing failures caused by all releases can be increased by this “idle time”. If \( k \tau > \Delta t \), the channel will “delay” for the time \( k(\tau - \Delta t) \), thereby increasing the average service time of one failure associated with the next release by this amount. Let us introduce the following notations:

\[
\begin{align*}
+ p_k \tau &= (\tau(A) - \Delta t) > 0, \quad (15) \\
- p_k \tau &= (\tau(A) - \Delta t) \leq 0. \quad (16)
\end{align*}
\]

Let us divide the set of releases \( \{A^k | k=1, k(A)\} \) into two subsets \( + k A = \{A | k=1, k(A)\} \) and \( - k A = \{A | k=1, k(A)\} \), where \( k^-(A) \) and \( k^+(A) \) are the number of releases for which relations (15) and (16) are true, respectively. The following equality holds for the partition:

\[
+ k(A) + k(A) = k(A). \quad (17)
\]

Let \( p_{+(A | p=1, k(A))} \) and \( p_{-(A | p=1, k(A))} \) be the release deployment sequences for which the corresponding sequences \( (\tau(A^p) | p=1, k(A)) \) and \( (\tau(A)| p=1, k(A)) \) are not increasing. Let also \( t_k = \{t_k \}, k=1,2,3,\ldots \), where \( (t_k, t_{k+1}) = \Delta t \) is the sequence of times at which the deployment of releases is performed. Let \( \tau_p(A^p| p=1, k(A)) \) denote the total time of the delays in the start of failures servicing due to the deployment of individual releases, caused by incomplete failures servicing associated with the deployment of previous releases. Then,

\[
\tau_p(A^p| p=1, k(A)) = (\tau_+ - \Delta t) + (\tau_+ + \tau_+ + \Delta t) + \ldots + \sum_{k=1}^{k(A)} \tau_k (k(A) - k + 1) \tau_k \frac{1 + k(A)}{2} k(A) \Delta t 
\]

Let \( \tau_y(A^p| p=1, k(A)) \) denote the total idle time of the service channel when deploying releases in accordance with the sequence \( (A^p| p=1, k(A)) \). Since each term in (18) is not negative, then for a non-decreasing sequence \( (\tau_p | p=1, k(A)) \) the variable \( \tau_y(A^p| p=1, k(A)) \) is equal to zero.

Let us formulate the following optimization problem:

\[
\begin{align*}
\tau_y(A^p| p=1, k(A)) &\rightarrow \min, \quad (19) \\
\tau_y(A^p| p=1, k(A)) &= 0. \quad (20)
\end{align*}
\]

Substantially, the task is to find such sequence of releases deployment \( (A^p_k| p=1, k(A)) \) that minimizes the total time of delays \( \tau_y \) of the start in failures servicing caused by the deployment of individual releases, and ensures completion of service of all failures in the standard period of time \( (0, \Delta t k(A)) \), i.e. ensures equality \( \tau_y = 0 \).

To solve problems (19) – (20), we apply a discrete version of the local optimization method (sequential improvement of some chosen basic solution). As a basic solution let us choose a sequence \( (A^p_k| p=1, k(A)) \) of release deployment for which the corresponding sequence \( (\tau(A^p_k)| p=1, k(A)) \) is non-increasing. For the purpose of simplicity a diagram of the method application for the case \( k(A) = 5 \) is provided. Let us assume that \( |A^+| = k^-(A) = 2 \) and \( |A^-| = k^+(A) = 3 \).

1. We form the product of sets \( A^+ \) and \( A^- \):
\[ A^+ \times A^- = \{(A^1_+, A^1_-), (A^2_+, A^2_-), (A^3_+, A^3_-), (A^4_+, A^4_-), (A^5_+, A^5_-), (A^6_+, A^6_-)\} \] (21)

2. The set \( A^+ \times A^- \) is divided into two subsets \( A^{++} \) and \( A^- \), defining them as follows:
\[
A^{++} = \{(A^p_+, A^p_-)\mid \tau(A^p_+, A^p_-) - 2\Delta t > 0\}, \quad (22)
\]
\[
A^- = \{(A^p_+, A^p_-)\mid \tau(A^p_+, A^p_-) - 2\Delta t \leq 0\}, \quad (23)
\]
\[
A^p_+, A^p_- \in \{A^k_+ \mid k = 1, k(A)\}. \quad (24)
\]

3. The elements of the sets \( A^{++} \) and \( A^- \) are arranged not in the order of increase variables \( \tau(A^p_+, A^p_-) \), respectively.

4. Let us choose from the set \( A^{++} \) such element that
\[
(A^p_+, A^p_-)_{\text{min}} = \arg \min_{(A^p_+, A^p_-) \in A^{++}} (\tau(A^p_+, A^p_-)). \quad (25)
\]

5. Let us include an ordered pair of releases \((A^p_+, A^p_-)\) as the initial component of the desired sequence of releases deployment.

6. Let us exclude from the set \( A^- \) elements containing releases \( A^{++} \) and \( A^{--} \).

7. Let us choose from the set \( A^- \) such element \((A^p_+, A^p_-)\) that
\[
(A^p_+, A^p_-)_{\text{min}} = \arg \min_{(A^p_+, A^p_-) \in A^-} (\tau(A^p_+, A^p_-)). \quad (26)
\]

8. Let us include an ordered pair of releases \((A^p_+, A^p_-)\) as the second component of the desired sequence of releases deployment.

9. The desired deployment sequence is complemented with the remaining fifth release.

6. Example

Let us consider an example of solving the problem with the initial data shown in table 1.

| A  | A_1 | A_2 | A_3 | A_4 | A_5 | A_6 | A_7 | A_8 | A_9 |
|----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| n_i | 15  | 12  | 10  | 14  | 8   | 17  | 25  | 20  | 15  |
| n_i^b | 4   | 3   | 5   | 6   | 1   | 11  | 17  | 2   | 5   |
| n_i^s | 9   | 6   | 4   | 3   | 5   | 5   | 1   | 15  | 4   |
| n_i^u | 2   | 3   | 1   | 5   | 2   | 1   | 7   | 3   | 6   |
| p(a^b) | 0.015 | 0.037 | 0.023 | 0.009 | 0.043 | 0.026 | 0.046 | 0.049 | 0.023 |
| p(a^s) | 0.023 | 0.019 | 0.049 | 0.006 | 0.051 | 0.009 | 0.04  | 0.011 | 0.044 |
In accordance with the initial data \( m(A) = 43.5 \). With the intensity of service \( \beta = 1.3 \) and \( \Delta t = 7 \) the minimum number of releases will be

\[
k(A) = \frac{43.5}{7 \times 1.3} = 5. 
\]  

(27)

Applying one of the well-known algorithms for solving the “stone problem” for 9 requests and 5 releases (the “greedy” algorithm is used in this example) and ordering the received releases in descending order of their “weights”, we obtain the following basic solution \([4,5]\):

\[
\begin{align*}
A^1 &= \{A_1, A_3\}, m(A^1) = 4.7 + 7.7 = 9.4, \\
A^2 &= \{A_2\}, m(A^2) = 8.8, \\
A^3 &= \{A_5, A_6\}, m(A^3) = 4.8 + 3.7 = 8.5, \\
A^4 &= \{A_8, A_3\}, m(A^4) = 5.4 + 3.1 = 8.5, \\
A^5 &= \{A_2, A_9\}, m(A^5) = 5.1 + 3.2 = 8.3.
\end{align*}
\]  

(28)

We form the product of the sets \( A^- \) and \( A^- \), divide it into subsets \( A^{++}, A^- \), and arrange the elements of each of them not by the increase of the variables \( \tau(A^{++}, A^-) \), table 2.

| \( p(a^*) \) | 0.031 | 0.009 | 0.019 | 0.044 | 0.031 | 0.05 | 0.04 | 0.050 | 0.015 |
|----------|--------|--------|--------|--------|--------|------|------|--------|--------|
| \( n^{su}(A_i) \) | 0.329 | 0.252 | 0.33 | 0.292 | 0.36 | 0.381 | 1.102 | 0.413 | 0.381 |
| \( L(a^i) \) | 93 | 70 | 76 | 79 | 45 | 67 | 128 | 55 | 51 |
| \( L(a^i) \) | 112 | 127 | 51 | 155 | 39 | 33 | 55 | 138 | 36 |
| \( L(a^i) \) | 24 | 10 | 25 | 32 | 27 | 34 | 18 | 24 | 31 |
| \( m(A^+_1) \) | 1.395 | 2.59 | 1.748 | 0.711 | 1.935 | 1.742 | 5.888 | 2.695 | 1.173 |
| \( m(A^-_1) \) | 2.576 | 2.413 | 2.499 | 0.93 | 1.989 | 0.297 | 2.2 | 1.518 | 1.584 |
| \( m(A^+_1) \) | 0.744 | 0.09 | 0.475 | 1.408 | 0.837 | 1.7 | 0.72 | 1.2 | 0.465 |
| \( m(A^-_1) \) | 4.715 | 5.093 | 4.722 | 3.049 | 4.761 | 3.739 | 8.808 | 5.413 | 3.222 |

Performing steps 4 - 9, we get two equivalent solutions:
for which
\[ \tau_z = 1.6, \text{ but } \tau_y = 0, \text{ while for a basic solution } (A^1, A^2, A^3, A^4, A^5) \]
\[ \tau_z = 2.5 \text{ and } \tau_y = 0. \]

7. Conclusion
For the considered task of forming releases, the assumption that the incorrect deployment of release assets is manifested through service failures immediately, in the shortest period from the moment of deployment, is essential. In practice, this assumption is usually not fulfilled, since the use of services is cyclical (daily, weekly, monthly, quarterly, annually). Of interest is the problem of generating releases, which takes into account both the structural properties of IT services and the dynamics of their use by users.

References
[1] 2007 OGC-ITIL V3-6 Service Lifecycle, Introduction ITIL TSO p 173
[2] Zimin V V, Ivushkin A A, Kulakov S M and Ivushkin K A 2013 Fundamentals of Life Cycle Management Services for Informatics and Automation Systems (ITIL Best Practices): Textbook for Universities (Kemerovo: Kuzbassvuzizdat) p 500
[3] Zimin V V, Burkova I V and Zimin A V 2017 Models and Mechanisms for Lifecycle Management of IT Services. Monograph (Saarbrucken: LAB LAMBERT Academic Publishing) p 328
[4] Burkova I V 2012 Network Programming Method in Project Management Tasks abstract of doc. thesis in 05.13.10 (Moscow: ICS RAS) p 40
[5] Burkov V N, Dzyubko S I and Yagupov A A 1995 Automatics and Telemechanics 7 124 –130
[6] Novikov DA 2007 Theory of Management of Organizational Systems (Moscow: Fizmatgiz) p 584