Problemy, analiza przyczyn i postmortem: żeby ta sama awaria nie wróciła

Poziom: średnio

Dotyczy: zarządzanie problemami w Zielonej, GLPI, baza wiedzy, Graylog, Zabbix

Czas: ~35 min

Incydent kończy się, gdy usługa znów działa. Problem kończy się dopiero wtedy, gdy przyczyna zniknęła i awaria nie może wrócić. Strona opisuje, kiedy otwieramy problem, jak prowadzimy bazę znanych błędów, jak szukamy przyczyny trzema prostymi metodami i jak piszemy postmortem bez szukania winnych.

Skróty użyte na tej stronie (8)

Skrót	Co znaczy
GLPI	Gestionnaire Libre de Parc Informatique: otwarty system do zgłoszeń, inwentarza i bazy wiedzy IT; w Zielonej działa w kontenerze na docker01
ITIL	Information Technology Infrastructure Library: zbiór dobrych praktyk zarządzania usługami IT, dziś rozwijany przez PeopleCert; wersja 4 od 2019, wersja 5 od 2026
KEDB	Known Error Database: baza znanych błędów, czyli problemów z opisaną przyczyną albo obejściem, z której korzysta service desk
MSSQL	Microsoft SQL Server: silnik bazy danych Microsoftu, na nim w Zielonej stoi księgowość Optima
RMAN	Recovery Manager: narzędzie Oracle do kopii zapasowych
SRE	Site Reliability Engineering: sposób prowadzenia utrzymania wymyślony w Google, w którym inżynierowie oprogramowania pilnują niezawodności, z budżetem błędów i postmortemem bez szukania winnych
VPN	Virtual Private Network: szyfrowany tunel do sieci firmowej
WMS	Warehouse Management System: system zarządzania magazynem

Fiszka

□□ TYLKO ODCZYT

Otwarcie problemu: kiedy

```
Problem otwieramy, gdy:
- ten sam objaw wraca trzeci raz w ciągu 30 dni
- był incydent poważny (zawsze)
- incydent zamknięto obejściem, a przyczyna jest nieznana
- monitoring pokazuje trend, który doprowadzi do awarii (dysk, pamięć, certyfikat)
- dostawca zgłosił błąd w produkcie, którego używamy
```

Problem to osobny zapis w GLPI, powiązany ze wszystkimi incydentami, które do niego doprowadziły. Właścicielem problemu jest jedna osoba, nawet jeśli pracują nad nim dwie.

□□ **GLPI** = Gestionnaire Libre de Parc Informatique: otwarty system do zgłoszeń, inwentarza i bazy wiedzy IT; w Zielonej działa w kontenerze na docker01

□□ PLIK KONFIGURACYJNY

Zapisanie znanego błędu: wpis w bazie znanych błędów

```
kod: KE-NNNN
tytuł: "[Usługa] objaw widziany przez użytkownika"
objaw: "co widzi użytkownik i co widać w monitoringu"
przyczyna: "znana przyczyna albo: w analizie"
obejście: "kroki dla L1, krótko, do wykonania w 15 minut"
trwałe_rozwiązanie: "numer zmiany CHG albo: brak, decyzja z datą"
problem: PRB-NNNN
incydenty: [INC-NNNN, INC-NNNN]
właściciel: "imię administratora"
przegląd: "RRRR-MM-DD"
```

Wpis powstaje, gdy tylko mamy obejście, nawet jeśli przyczyny jeszcze nie znamy. Dyżurny L1 szuka w tych wpisach, zanim zacznie diagnozę od zera.

□□ TYLKO ODCZYT

Zebranie zdarzeń do osi czasu: journalctl --since --until

```
journalctl --since "RRRR-MM-DD GG:MM" --until "RRRR-MM-DD GG:MM" -p warning -o short-iso
```

Pokazuje wpisy dziennika systemowego od poziomu ostrzeżenia w oknie awarii, z pełną datą. Okno bierzemy szersze niż awaria: godzinę przed pierwszym objawem.

Przykład z `docker01`:

```
2026-10-09T15:47:13+02:00 docker01 dockerd[1203]: level=error msg="write
/docker/lib/containers/...: no space left on device"
```

journalctl = journal control: czytanie dziennika systemd

TYLKO ODCZYT

Zebranie zdarzeń z Windows do osi czasu: Get-WinEvent

```
Get-WinEvent -FilterHashtable @{LogName='System','Application'; Level=1,2,3;
StartTime=[datetime]'RRRR-MM-DD GG:MM'; EndTime=[datetime]'RRRR-MM-DD GG:MM'} -
ComputerName NazwaSerwera | Sort-Object TimeCreated | Select-Object TimeCreated,
ProviderName, Id, Message
```

> (gdzie poziomy 1, 2 i 3 to Critical, Error i Warning)

To samo dla serwerów Windows: błędy i ostrzeżenia z dzienników systemu i aplikacji w oknie awarii, posortowane po czasie. Wynik wklejamy do osi czasu w postmortemie.

TYLKO ODCZYT

Pięć pytań dlaczego: zapis w problemie

```
Objaw: Optima nie zapisywała faktur 28.08, 16:05 do 16:52
1. Dlaczego? sql01 odrzucał transakcje, bo log CDN_Zielona nie miał miejsca
2. Dlaczego? dysk L: z logiem zapełnił się w zamknięciu miesiąca
3. Dlaczego? kopia logu co godzinę nie nadążała za ruchem w zamknięciu miesiąca
4. Dlaczego? harmonogram kopii nie uwzględnia zamknięcia miesiąca
5. Dlaczego? nikt nie przeglądał harmonogramu od wdrożenia w 2023 roku
Przyczyna do usunięcia: brak przeglądu harmonogramu kopii w kalendarzu zmian
```

Zatrzymujemy się na przyczynie, którą możemy usunąć zmianą. Pięć to zwyczaj, a nie reguła: czasem wystarczą trzy pytania, czasem trzeba siedmiu.

🔒 TYLKO ODCZYT

Szablon postmortemu: strona w bazie wiedzy

```
Tytuł: Postmortem INC-NNNN: [Usługa] objaw, data
Podsumowanie: trzy zdania dla pana Zbigniewa
Wpływ: kto, jak długo, co nie zostało zrobione (faktury, wysyłki)
Oś czasu: godzina | co się stało | skąd wiemy
Przyczyna główna i czynniki składowe
Co zadziałało dobrze
Co nie zadziałało
Działania: co | właściciel | termin | numer CHG albo PRB
Wnioski dla bazy wiedzy: które artykuły poprawiamy
```

Postmortem piszemy w ciągu pięciu dni roboczych od incydentu poważnego. Strona trafia do książki „Postmortemy” w bazie wiedzy, z tagiem usługi.

Z życia wzięte

🔒 Z ŻYCIA WZIĘTE

Poniedziałek, 6:20, trzeci tydzień z rzędu. Marek dzwoni, że skanery w Swarzędzu tracą sesję w WMS i każdy magazynier musi się logować od nowa. Tomek trzeci raz zamyka incydent obejściem „ponowne logowanie”, ale tym razem przy zamknięciu GLPI pokazuje mu dwa poprzednie zgłoszenia z tym samym objawem, więc otwiera problem. Oś czasu z logów `ora01` pokazuje, że cotygodniowa pełna kopia RMAN, uruchamiana w niedzielę wieczorem, kończy się w poniedziałek o 6:05, dokładnie gdy czterdzieści skanerów loguje się naraz. Zmiana przesuwająca kopię na sobotę wchodzi w najbliższym oknie, a w bazie znanych błędów powstaje wpis z obejściem na czas oczekiwania.

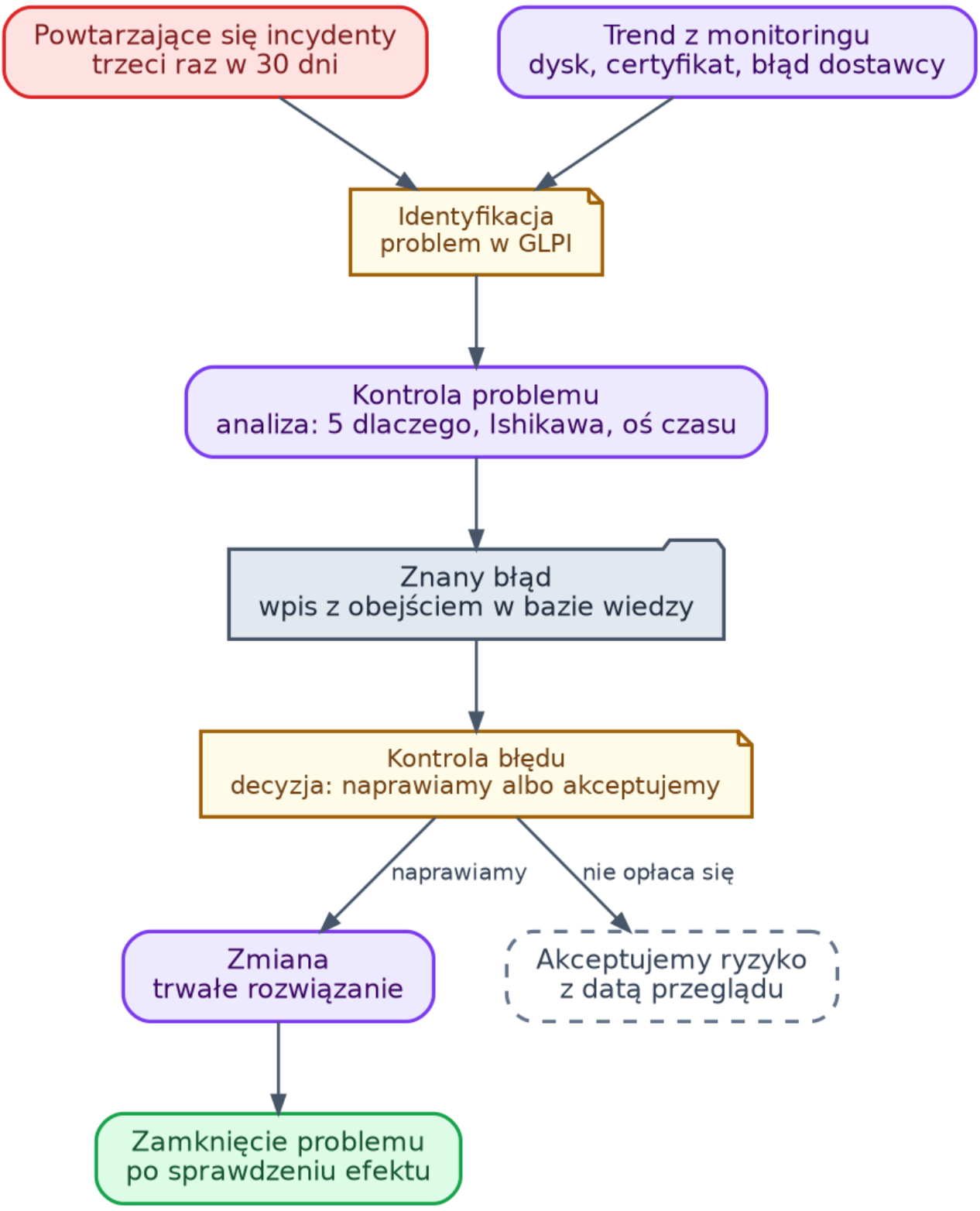
Jak to działa

🔒 Rozwiń: jak to działa pod spodem

Problem i jego trzy etapy

> **Pojęcie: zarządzanie problemami.** Praktyka, której celem jest zmniejszenie liczby i skutków incydentów przez znajdowanie ich przyczyn, wypracowanie obejść dla tych, których nie da się od razu usunąć, i usuwanie przyczyn zmianami [1]. ITIL 4 dzieli ją na trzy etapy, które w małym zespole przechodzą płynnie jeden w drugi.

Etap	Co robimy	Co powstaje
Identyfikacja problemu	zauważamy powtarzające się incydenty, trendy w monitoringu, zgłoszenia dostawców	zapis problemu w GLPI z powiązanymi incydentami
Kontrola problemu	analizujemy przyczynę, szukamy obejścia, oceniamy, czy usunięcie się opłaca	wpis znanego błędu z obejściem
Kontrola błędu	pilnujemy znanych błędów, wracamy do nich, gdy zmieni się sytuacja, zlecamy trwałe rozwiązanie	wniosek o zmianę, a po niej zamknięcie problemu



Rysunek 1. Droga problemu w Zielonej: powtarzające się incydenty i trendy z monitoringu otwierają problem, analiza daje wpis w bazie znanych błędów, trwałe rozwiązanie przechodzi przez zmianę, a problem zamykamy dopiero po sprawdzeniu efektu (kliknięcie otwiera pełny

rozmiar)

Problem reaktywny i proaktywny. Reaktywny zaczyna się od incydentów, które już się zdarzyły (skanery w poniedziałek). Proaktywny zaczyna się od sygnału, zanim ktoś ucierpiał: Zabbix pokazuje, że wolne miejsce na `/docker` spada o 2 procent tygodniowo, certyfikat wygasa za 30 dni, dostawca ogłasza błąd w wersji, której używamy. W dwuosobowym zespole proaktywne problemy przeglądamy raz w tygodniu, w przeglądzie tygodniowym.

Nie każdy problem usuwamy. Czasem trwałe rozwiązanie kosztuje więcej niż obejście stosowane raz na kwartał. Wtedy problem zostaje znanym błędem z obejściem, a decyzję „nie naprawiamy” zapisujemy z datą przeglądu. To uczciwsze niż problem otwarty na zawsze.

Baza znanych błędów

> **Pojęcie: baza znanych błędów.** Miejsce, w którym trzymamy znane błędy z obejściami, tak żeby service desk mógł je szybko znaleźć. W ITIL v3 nazywano ją KEDB (Known Error Database). W Zielonej to nie osobny program, tylko dwa połączone miejsca: problem w GLPI (z historią i powiązaniami) i krótki artykuł w bazie wiedzy BookStack z tagiem `typ=znany-blad` (z obejściem dla L1).

Zasady prowadzenia:

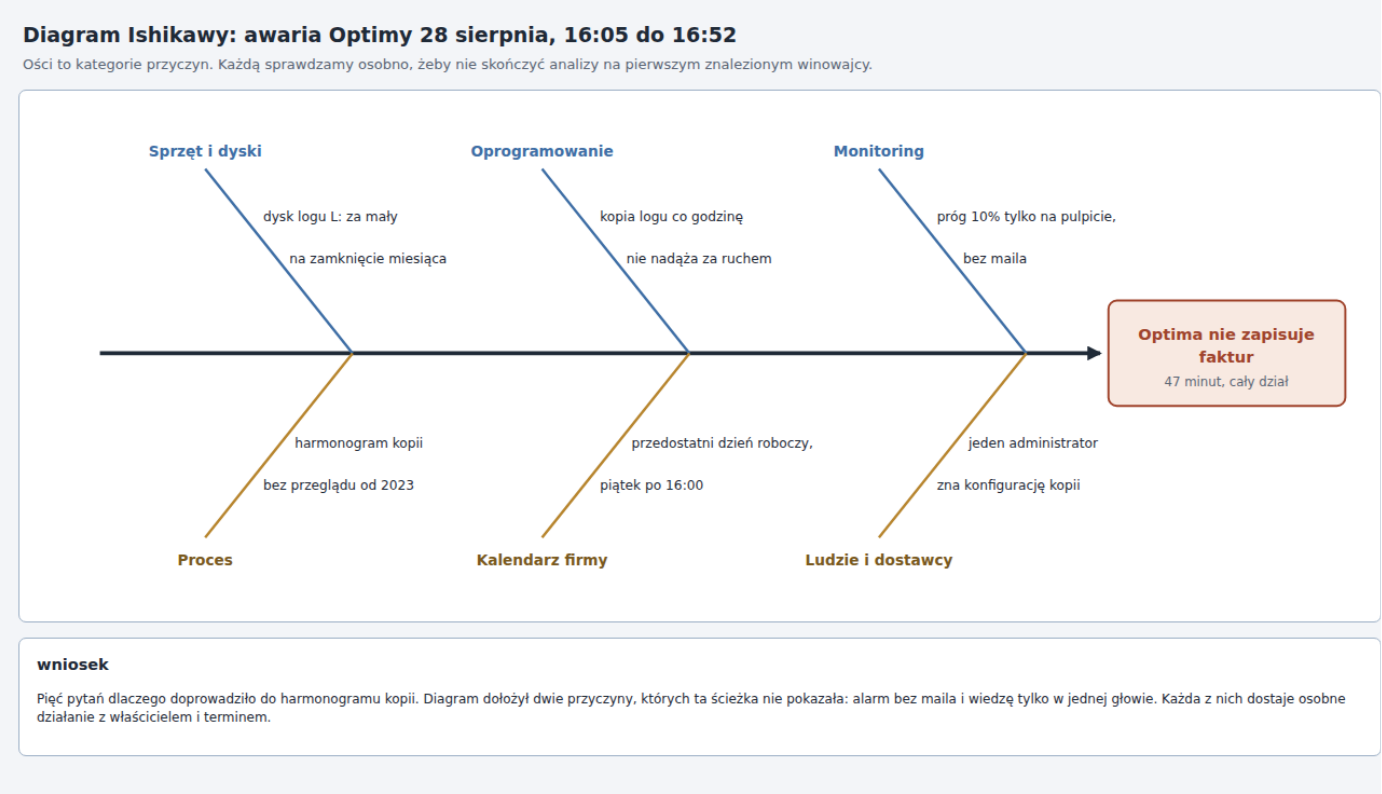
1. Wpis powstaje, gdy jest obejście, a nie dopiero po znalezieniu przyczyny.
2. Tytuł opisuje objaw słowami użytkownika, bo po nich L1 szuka.
3. Obejście da się wykonać w 15 minut bez uprawnień administratora serwerów albo jest jasno oznaczone, że wymaga L2.
4. Po trwałym rozwiązaniu wpisu nie kasujemy, tylko oznaczamy jako rozwiązany z numerem zmiany: przyda się, gdy błąd wróci po aktualizacji.

Analiza przyczyn

> **Pojęcie: analiza przyczyny źródłowej (RCA).** Uporządkowane szukanie przyczyny, której usunięcie zapobiegnie powtórzeniu awarii, zamiast poprzestania na pierwszym objawie. Używamy trzech prostych metod, często razem.

Pięć pytań dlaczego. Pytamy „dlaczego?” o objaw, potem o odpowiedź, i tak dalej, aż dojdziemy do przyczyny, którą możemy usunąć zmianą w systemie albo w sposobie pracy. Metoda wywodzi się z Toyoty. Jej słabość: prowadzi jedną ścieżką, a awarie często mają kilka przyczyn naraz, więc przy poważnych incydentach łączymy ją z diagramem Ishikawy.

Diagram Ishikawy. Nazywany też diagramem rybiej ości, od nazwiska japońskiego specjalisty od jakości Kaoru Ishikawy. Objaw stoi w głowie ryby, a ości to kategorie przyczyn; dla każdej kategorii zapisujemy, co mogło się przyczynić. Zaletą jest to, że zmusza do sprawdzenia kategorii, o których nie pomyśleliśmy, na przykład procesu albo dostawcy, gdy wszyscy patrzą na serwer.



Rysunek 2. Diagram Ishikawy dla awarii Optimy z 28 sierpnia: sześć kategorii przyczyn, od sprzętu i oprogramowania po proces, ludzi, dostawców i kalendarz firmy (kliknięcie otwiera pełny rozmiar)

Oś czasu. Lista zdarzeń z godzinami, zebrana z logów, monitoringu, zgłoszeń i wątku w Teams. Oś czasu odpowiada na pytania, których nie da się rozstrzygnąć z pamięci: co było pierwsze, ile minęło od alarmu do reakcji, która czynność naprawiła usługę. Godziny bierzemy ze źródeł, a nie z pamięci, i dlatego wszystkie serwery muszą mieć zsynchronizowany czas z `dc01`.

Przykład osi czasu dla awarii Optimy:

Godzina	Co się stało	Skąd wiemy
15:10	wolne miejsce na dysku logu `sql01` spada poniżej 10 procent	Zabbix, wyzwalacz Warning (bez maila)
15:58	log transakcyjny `CDN_Zielona` nie może urosnąć	dziennik zdarzeń `sql01`, zdarzenie 9002

Godzina	Co się stało	Skąd wiemy
16:05	pani Halina dzwoni: zapis faktur wisi	GLPI, INC-0412
16:09	ogłoszony incydent poważny, pierwszy komunikat do księgowości	wątek w Teams, mail
16:31	diagnoza: pełny dysk logu	notatka w INC-0412
16:44	kopia logu wykonana ręcznie, miejsce zwolnione	historia kopii w MSSQL
16:52	pani Halina potwierdza zapis faktur	GLPI
16:58	komunikat końcowy	mail

Z tej osi od razu widać dwa działania: wyzwalacz przy 10 procentach musi wysyłać maila (dziś tylko pulpit), a harmonogram kopii logu trzeba zagęścić na zamknięcie miesiąca.

> □□ Zdarzenie 9002 w MSSQL oznacza pełny log transakcyjny. Szczegóły i kolejność działań są w książce o bazach danych, w części o MSSQL.

Postmortem bez szukania winnych

Pojęcie: postmortem. Pisemny przegląd po awarii: co się stało, jaki był wpływ, jaka była przyczyna i co zrobimy, żeby się nie powtórzyło [2]. W ITIL 4 podobną rolę pełni przegląd po incydencie poważnym.

Pojęcie: postmortem bez szukania winnych. Przegląd, w którym zakładamy, że każdy działał w dobrej wierze, z wiedzą, którą wtedy miał, i szukamy, co w systemie i w sposobie pracy pozwoliło na błąd, a nie kto go popełnił [2]. Książka Google o SRE opisuje to jako kulturę, w której ludzie zgłaszają błędy, zamiast je ukrywać, bo za zgłoszenie nikt nie jest karany.

To nie jest uprzejmość, tylko praktyczny rachunek. Jeśli Tomek boi się, że za zmianę na `fw01` w piątek po 15:00 zostanie skrytykowany, następnym razem nie przyzna się, że to on ją zrobił, a diagnoza potrwa trzy razy dłużej. Pytamy więc „dlaczego zasady pozwoliły na zmianę w piątek po 15:00?”, a nie „dlaczego Tomek to zrobił?”.

Kiedy piszemy postmortem:

Sytuacja	Postmortem
Incydent poważny	zawsze, w ciągu 5 dni roboczych
Utrata danych albo podejrzenie wycieku	zawsze, razem z dokumentacją dla zarządu
Incydent P1 bez ogłoszenia incydentu poważnego	krótki, w notatce problemu
Zmiana, która spowodowała awarię	zawsze, nawet krótki
Sytuacja „o włos” (prawie awaria)	zalecany, bo uczy najtaniej

Postmortem czytamy razem na przeglądzie tygodniowym, a działania z niego trafiają do GLPI jako problemy albo zmiany z właścicielem i terminem. Postmortem bez działań z terminami jest tylko opowiadaniem.

Częste błędy

Objaw	Przyczyna	Co robimy
Ten sam incydent zamknięty po raz piąty	nikt nie otworzył problemu	reguła „trzeci raz w 30 dni”, a w GLPI widok incydentów pogrupowanych po kategorii i elemencie konfiguracji
Analiza kończy się na „błąd ludzki”	zatrzymaliśmy się za wcześnie	pytamy dalej: co pozwoliło na błąd, czego zabrakło w procesie, narzędziu, szkoleniu
Postmortem szuka winnego	kultura karania	piszemy o systemie i decyzjach, bez nazwisk w przyczynie; nazwiska tylko przy działaniach (właściciel)
Działania z postmortemu nigdy nie zrobione	brak właściciela i terminu	każde działanie jako zapis w GLPI, przegląd tygodniowy sprawdza terminy
Oś czasu się nie zgadza między serwerami	różny czas na serwerach	synchronizacja czasu z `dc01` na wszystkich serwerach, alarm Zabbixa przy rozjeździe
Znany błąd bez obejścia	wpis zrobiony, ale niedokończony	obejście jest polem obowiązkowym; jeśli go nie ma, wpisujemy „brak obejścia, eskalować do L2”
Problem otwarty od roku	nikt nie zdecydował, że nie naprawiamy	decyzja „akceptujemy ryzyko” z datą przeglądu i zamknięcie problemu jako znanego błędu

□□ Ćwiczenia

1. VPN Kasi zrywa się co kilka dni w różnych hotelach. Incydent zamykaliśmy już czterokrotnie restartem klienta VPN. Co robimy przy piątym?

Odpowiedź

Otwieramy problem i wiążemy z nim pięć incydentów, zapisujemy znany błąd z obejściem (restart klienta), a w analizie zbieramy oś czasu z logów `fw01` i z laptopa Kasi z godzinami zerwań. Dopiero z niej widać, czy przyczyną jest sieć hotelowa, wersja klienta, czy ustawienia tunelu.

1. W pięciu pytaniach dlaczego doszliśmy do odpowiedzi „bo Tomek zapomniał”. Co dalej?

Odpowiedź

Pytamy dalej: dlaczego dało się zapomnieć? Brak listy kontrolnej, brak przypomnienia w GLPI, brak drugiej pary oczu przy czerwonej karcie. Przyczyną do usunięcia jest brak zabezpieczenia w procesie, a działaniem na przykład lista kontrolna w szablonie zmiany.

1. Zabbix pokazuje, że certyfikat `git.zielona.example` wygasa za 25 dni. Czy to problem?

Odpowiedź

Tak, problem proaktywny: nic jeszcze nie działa źle, ale bez działania za 25 dni będzie incydent. Zapisujemy go, przypisujemy właściciela i zmianę standardową z odnowieniem certyfikatu, a przy okazji sprawdzamy, dlaczego odnowienie nie jest automatyczne.

1. Pan Zbigniew pyta po awarii Optimy, kto zawinił. Co odpowiadamy i co mu dajemy?

Odpowiedź

Odpowiadamy, że szukamy przyczyny w systemie, a nie winnego, bo to daje szybsze i uczciwsze zgłaszanie błędów w przyszłości. Dajemy mu podsumowanie z postmortemu w trzech zdaniach, wpływ (ile faktur, ile minut), przyczynę i listę działań z właścicielami i terminami.

Powiązane: Incydent i incydent poważny: od zgłoszenia do zamknięcia · Zmiany, organ zatwierdzający i okna zmian: jak zmieniamy, żeby nic nie zepsuć · Konfiguracja i wiedza: NetBox jako CMDB i BookStack jako baza wiedzy · Host, element danych, wyzwalacz, akcja: jak to się składa

Źródła: [1] Parafraza własna na podstawie: AXELOS, „ITIL Foundation, ITIL 4 Edition” (2019), praktyka problem management (etapy identyfikacji, kontroli problemu i kontroli błędu) i hasła problem, known error, workaround. · [2] Beyer, Jones, Petoff, Murphy (red.), „Site Reliability Engineering: How Google Runs Production Systems”, O'Reilly 2016, rozdział „Postmortem Culture: Learning from Failure” (sre.google/sre-book), parafraza własna. · Ishikawa i pięć pytań dlaczego: opisy ogólnodostępne w literaturze o zarządzaniu jakością.

Revision #7
Created 2026-10-11 14:26:57 UTC by Tomasz Konieczny
Updated 2026-10-26 14:25:37 UTC by Tomasz Konieczny