

# インシデントレスポンスを自動化で支援する Slack Bot で人機一体なセキュリティ対策を実現する

GMOペパボ 伊藤洋也

CloudNative Days Track B 2021/11/04 17:20-18:00



## 伊藤洋也 いとう ひろや (@hiboma)

所属: GMOペパボ セキュリティ対策室 (在職14年目) at 栃木県那須塩原

職位: プリンシパルエンジニア

得意: Linux 低レイヤーのトラブルシューティング

ブログ / 技術エントリ

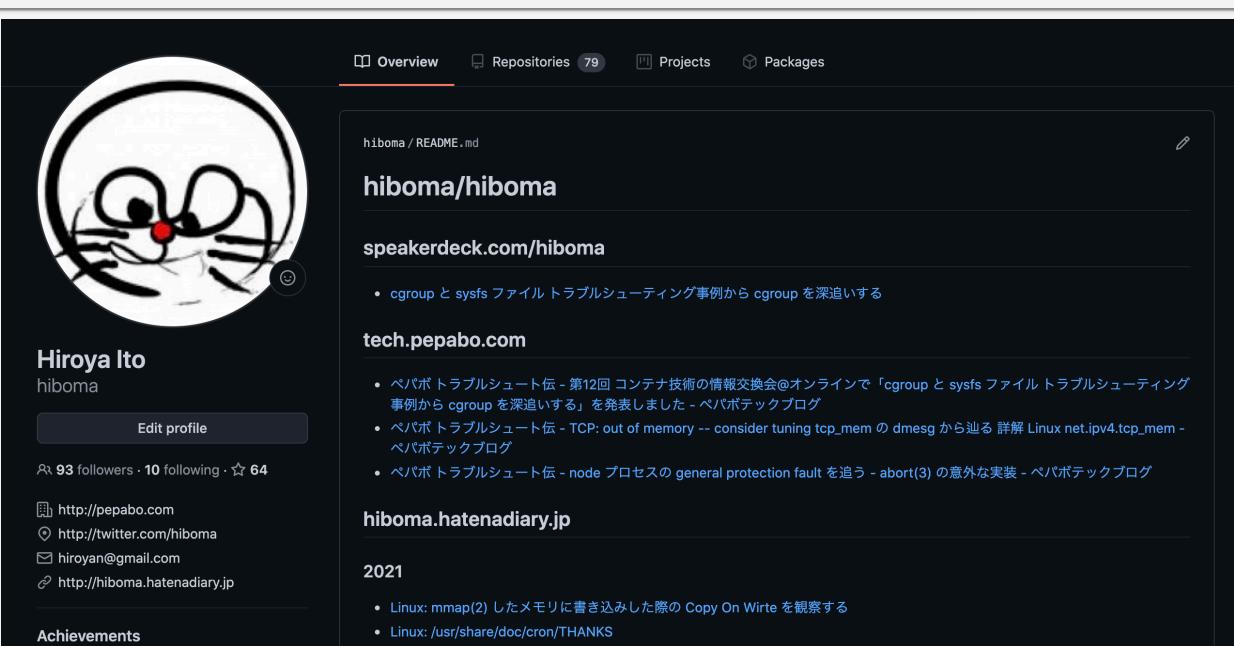


2021-10-08  
キャリアドリフトとITエンジニア

キャリアの話です

- ・バウンダリーレスキャリアとITエンジニア - hibomaの日記
- ・内的キャリアと自分で大切に思っている専門書 - hibomaの日記

<https://hiboma.hatenadiary.jp/>



hiboma / README.md  
**hiboma/hiboma**  
speakerdeck.com/hiboma  
tech.pepabo.com  
hiboma.hatenadiary.jp

A 93 followers · 10 following · ☆ 64

At 93 followers · 10 following · ☆ 64

http://pepabo.com  
http://twitter.com/hiboma  
hiroyan@gmail.com  
http://hiboma.hatenadiary.jp

Achievements

2021

- Linux: mmap(2)したメモリに書き込みした際の Copy On Write を観察する
- Linux: /usr/share/doc/cron/THANKS

<https://github.com/hiboma/hiboma>



## GMOペパボ セキュリティ対策室のミッション



GMOペパボのセキュリティ対策室とは情報セキュリティ基本方針を遵守し  
お客様、お取引先様、従業員から預る情報資産を適切に扱える  
文化形成、技術的仕組みをリードする組織

エンジニアリングのアプローチと併せて、組織内での文化の啓蒙にもコミットしていく組織です  
このスライドでも人機一体 = 人と機械が一体になる = Slack Bot の話と併せてマネジメントにも言及した話をします



Section 0

# 本編に先立って

## 本スライドでの「インシデント」の定義

本スライドで「インシデント」という用語を繰り返し用います。

GMOペパボでは「インシデント」の定義の通りとしています。本スライドもこれに準じた定義とします

情報資産のうち重要性1、2に属するものについて、アクセス権限がない人が閲覧することができた（機密性の問題）

重要性1. セキュリティ侵害が会社の財産、営業等へ重大な影響を及ぼす。

重要性2. セキュリティ侵害が会社の業務執行等へ重大な影響を及ぼす。

情報資産全てを対象にして、なくなってしまった（完全性の問題）

情報資産全てを対象に改ざんされてしまった（完全性の問題）

情報資産全てを対象に、一時的に利用できない状態が発生しお客様に影響が発生した（可用性の問題）

上記の定義に準じて、主に、可用性に影響する事象を「障害」と記載します

完全性・機密性に影響する事象を「セキュリティ・インシデント」として記載します

Section 1

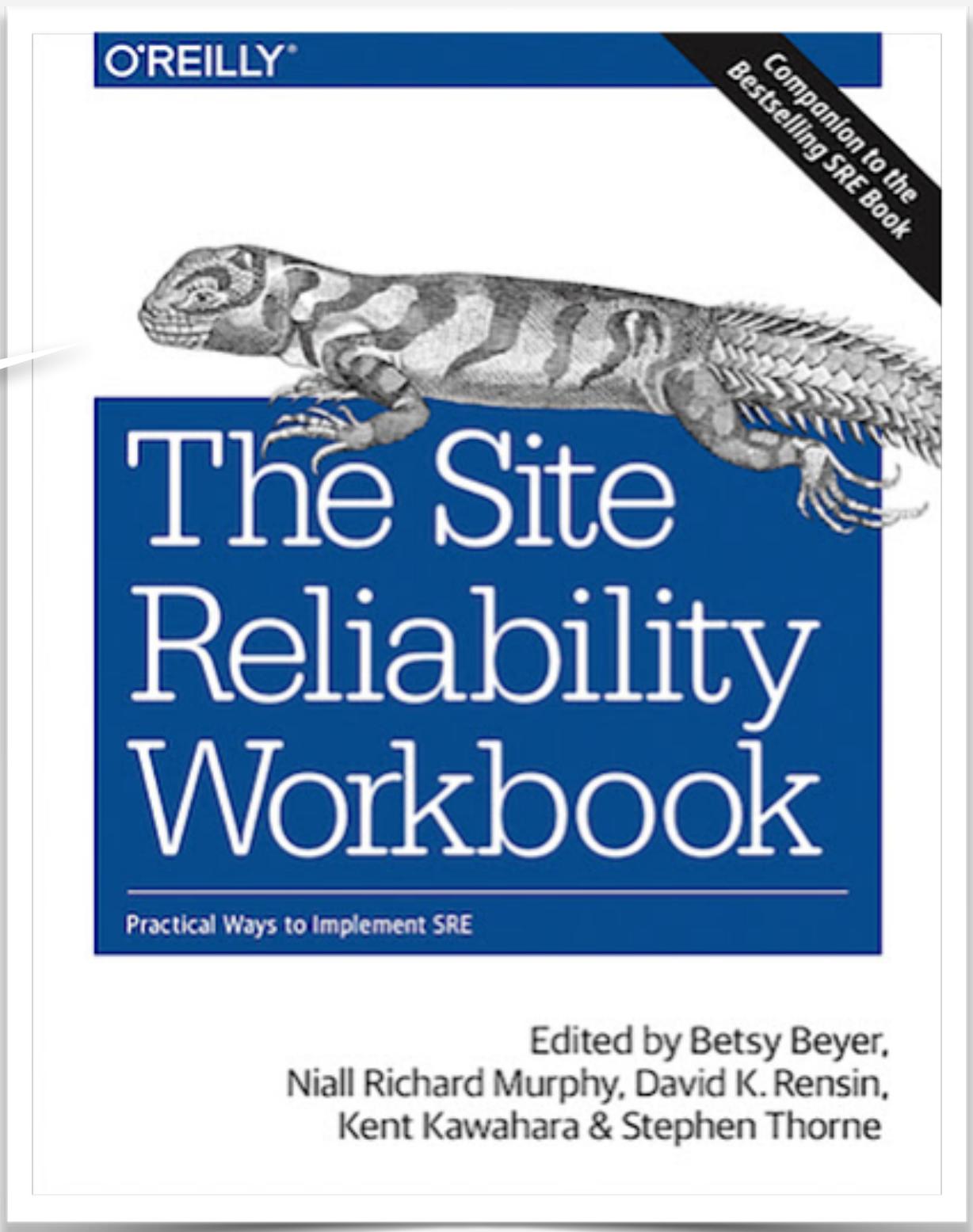
インシデントは避けられない



## インシデントは避けられない

CloudNative に標語される先進的な技術・アーキテクチャで作られたシステムが高信頼性を得できる一方で、依然として様々な問題が発生しインシデントを引き起こします

“誰もが自分のサービスが常にスムーズに動くことを望んでいますが、私たちは不完全な世界に生きているので、障害が発生することもあります。”



## 様々な問題

### アプリケーションの問題

- 設計・仕様の不備
- 実装のバグ
- リリースのミス

### 連携の問題

- 外部 API サービスの障害
- 依存するシステムの障害

### クラウド・インフラの問題

- IaaS, PaaS, \*aaSの障害
- DC・基幹ネットワーク障害

### ホストの問題

- カーネルのバグ・負荷
- ミドルウェアのバグ・負荷
- 設定のミス・不備

### 外部からの攻撃

- 不正なペイロード、DoS、マルウェア
- 脆弱性・ゼロデイ対応

### 人間の問題

- ヒューマンエラー(オペミス)
- システムの属人化

### ハードウェアの問題

- ディスクの故障
- 電源の故障
- ネットワーク機器の故障

### 社会状況の変化

- 法律・規制のアップデート
- 倫理・遵守意識の変化

## SRE とインシデント対応

SREにおいてもインシデント対応は切り離せないテーマになっている

### 『SRE サイトリアイアビリティエンジニアリング』

- 12章 効果的なトラブルシューティング
- 13章 緊急対応
- 14章 インシデント管理
- 15章 ポストモーテムの文化：失敗からの学び
- 16章 サービス障害の追跡

### 『サイトリライアビリティワークブック』

- 8章 オンコール
- 9章 インシデント対応



### 『Building Secure & Reliable Systems』

- 16. Disaster Planning
- 17. Crisis Management
- 18. Recovery and Aftermath

⚠️ SRE 本の「インシデント」は、いわゆる「障害」に比重を置いた表現かもしれないが、本スライドではセキュリティインシデントも含む広い解釈とした



## Section 2

# 避けられないインシデントに 立ち向かう

## 組織のレジリエンスを語る = ポジティブな発信テーマに

2018~ 2021年に書かれた「障害」 「インシデント」 を題材としたアウトプットを集めました

- [Infra Study Meetup #3 LT - Incident Response @tjun 氏](#)
- [【いでよ障害対応太郎】我々はインシデントにどう向き合っているのか | by yoshiken | FiNC Tech Blog | Medium](#)
- [障害発生！全員集合？ – オンコールアンチパターンからの一步前進 - Cybozu Inside Out | サイボウズエンジニアのブログ](#)
- [重大事故の時にどうするか？ | miyasaka | note](#)
- [失敗して攻め続けるために - freeeのエンジニアが障害対応で実践していること - freee Developers Blog](#)
- [SREチームのセキュリティインシデントゲームデー - メドピア開発者ブログ](#)
- [Webサービスの障害対応のときの思考過程 - ぱいぱいにっき](#)
- [システム障害との向き合い方 @sinamon129 #tokyogirlsrb - Speaker Deck\)](#)
- [再発防止策を考える技術 / #phpconsen - Speaker Deck\)](#)
- [障害対応とポストモーテム - Quipper Product Team Blog](#)
- [スタディストにおける障害対応の実践と今後の展望. #srefukuoka で登壇してきました！スタディスト開発ブログ | Medium](#)
- [【保存版】東京リージョンの AWS 障害発生時にクラスメソッドのテクニカルサポートチームがやっていること | DevelopersIO](#)
- [Webアプリケーションの障害対応について改めて意識すべき点ややれると良いことをまとめる - stefafafan の fa は3つです](#)
- [システム障害対応演習を実施した話 | NAVITIME\\_Tech | note](#)

SRE の広がりと併せて、 組織での障害やトラブル対応 = レジリエンスについてのアウトプットが増えたように思います。

twitter や ソーシャルブックマークでも建設的・好意的なコメントがつき、 ポジティブなテーマとして読まれている空気を感じます

## 組織のレジリエンスを語る > 一般化・普遍化する

先のエントリ群では、技術の各論より、プロセス・コミュニケーション・組織文化にフォーカスした内容が多い

- プロセスの構造化
  - マニュアル作り、フローの策定、役割分担
- 組織文化づくり
  - 失敗を非難しない文化 (Blameless)、ふりかえり / ポストモーテムの文化
- コミュニケーションツールの活用
  - Slack の bot や Workflow 活用

異なる事業を営む IT 企業が、よく似たプラクティス・アンチパターンを導いているのが印象的です。

組織を超えて一般化・普遍化して語りうる題材なのでしょう。逆に、ペパボの話が他の組織に響くところがあるかと思います

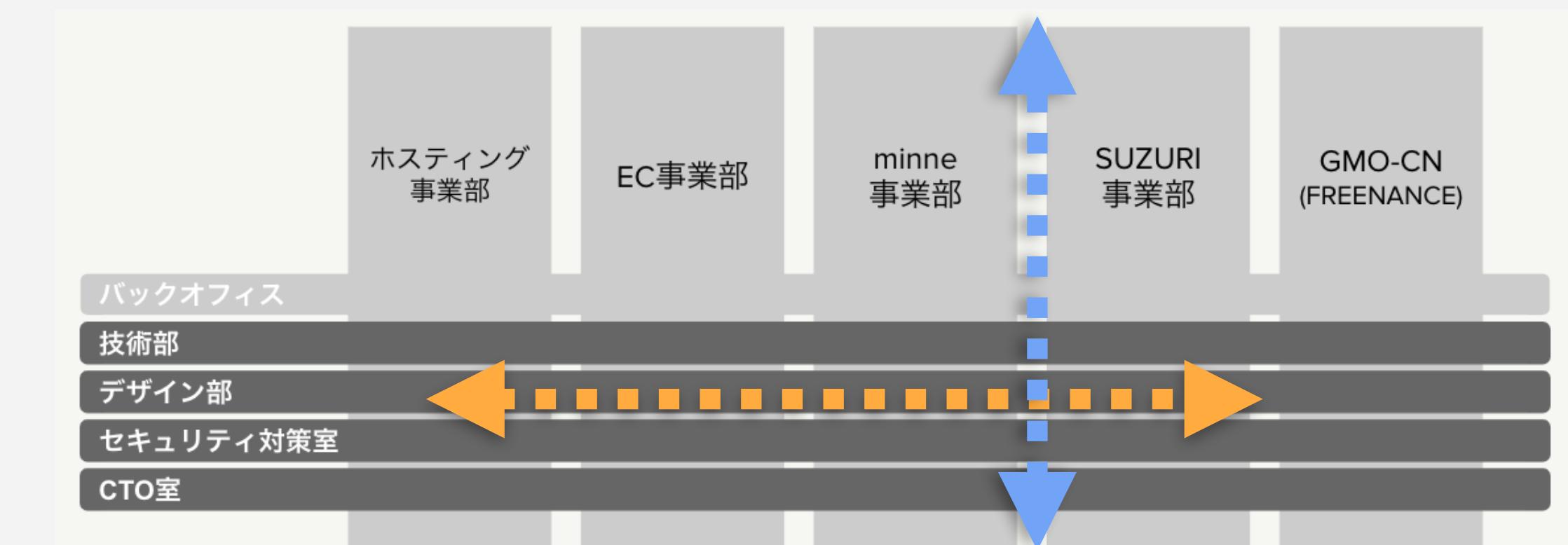
## インシデント対応 = 組織の縦・横と職種を超えたコラボレーション

GMOペパボでは、事業部制を敷いており、事業部ごとにサービスが従属します

基盤技術（例：プライベートクラウド）やサービス関連携で障害が発生した場合は、  
部署・サービス間を横断してコミュニケーションをとる必要があります

技術職以外の職種もインシデント対応に加わるケースもあります

- カスタマーサポート
- マネージャー
- 法務部
- 取締役



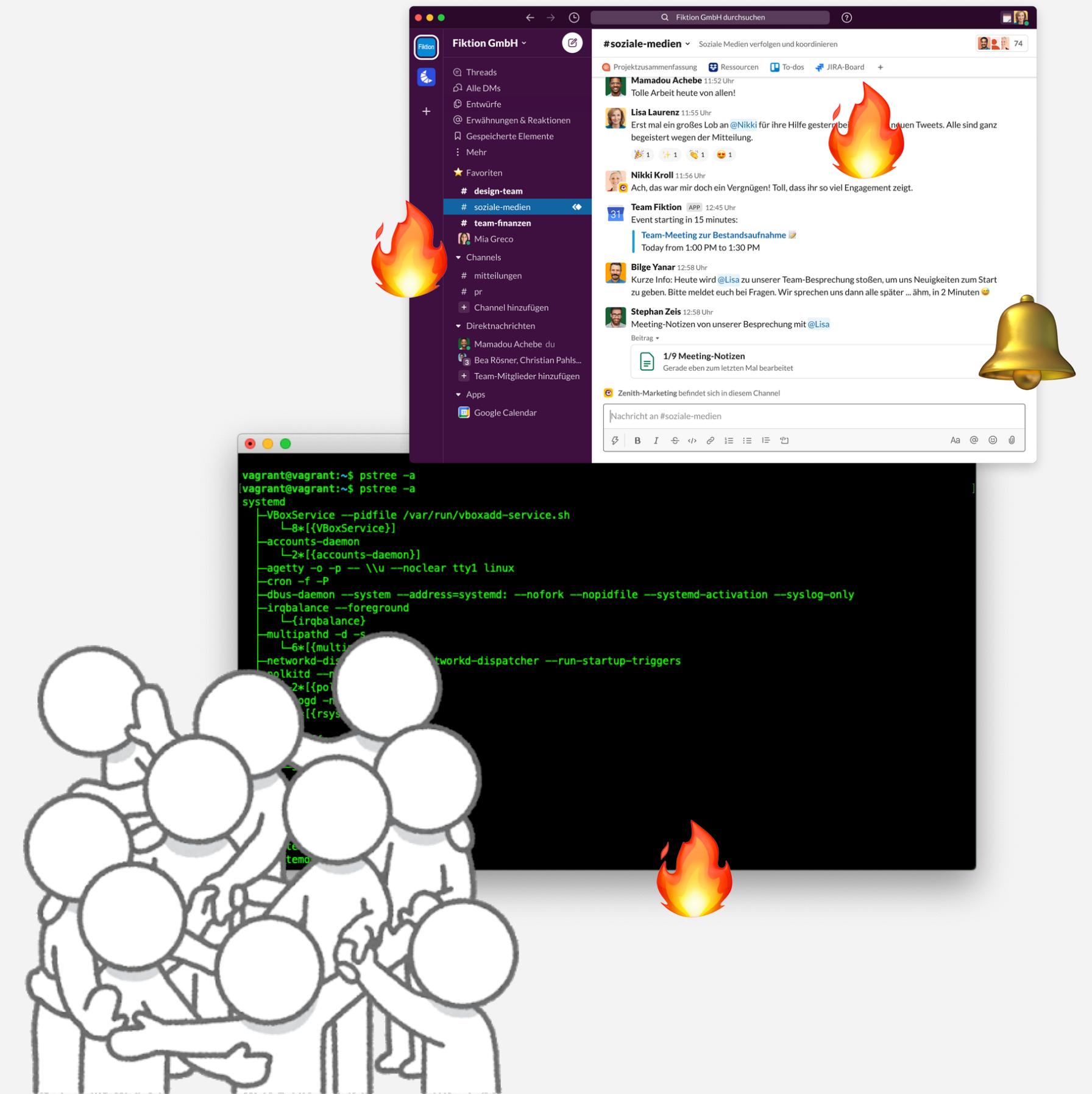
インシデント対応では組織の構造と職種間を超えてのコラボレーションが要求されます (\*)

⚠ 私はGMOペパボ以外の会社に所属したことがないため、他社様ではどのように対応を行うのか、知見・経験が全くありません。フィードバックをもらえると幸いです！

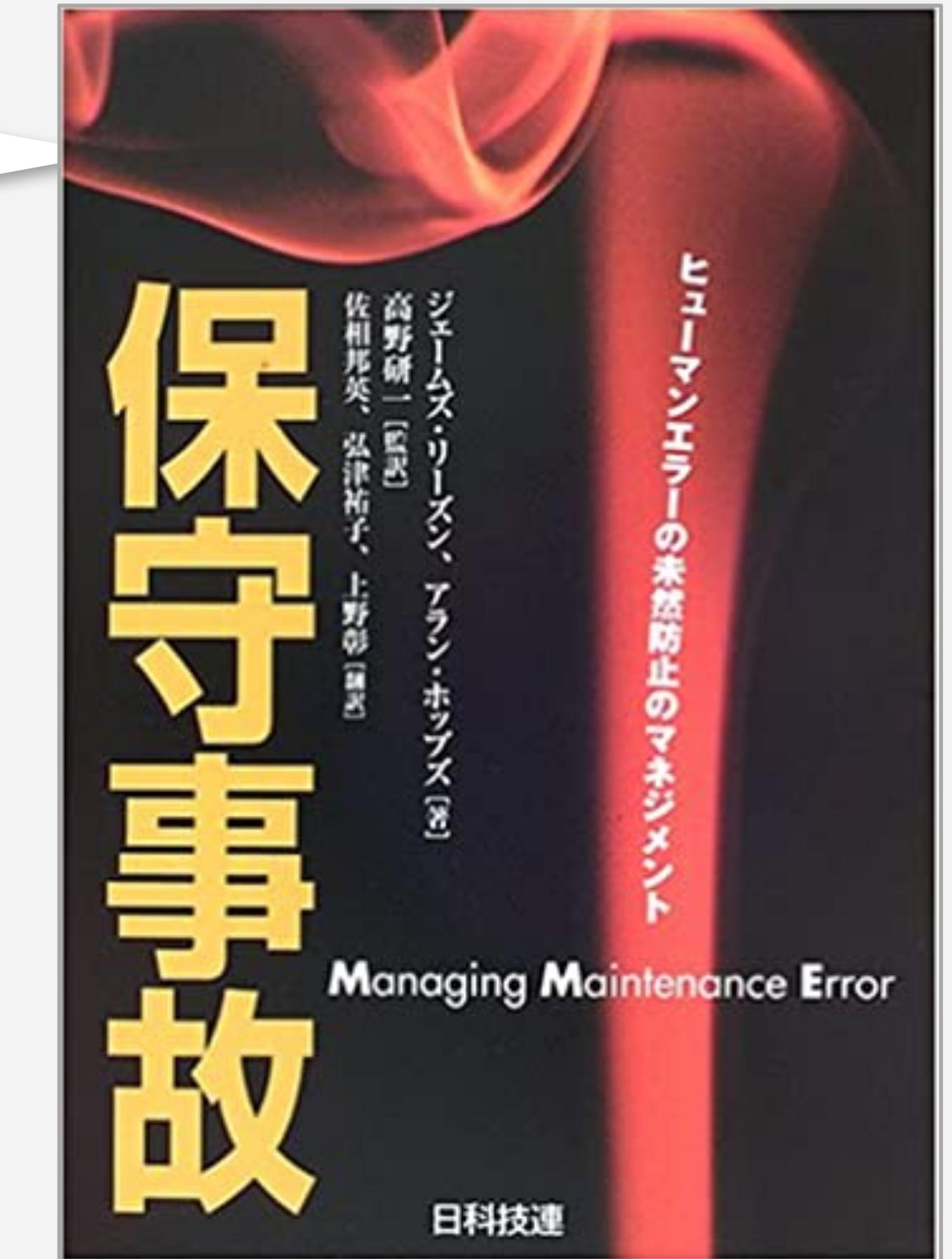
## 緊急時のオペレーションは大変

緊急時のオペレーションは目の前の問題で大変です

- ・大量の監視アラートが流れて同僚とのチャットがうまくいかなかった
- ・原因調査に没頭しまい、肝心の復旧作業が遅れた
- ・ただぼーっと同僚の作業（チャット）を眺めていた
- ・復旧作業に夢中で関係者に連絡が後回しになった
- ・記録をまとめようとしたが、ログがいろんなチャンネルに散逸している
- ・マニュアルやフローがあるが無視・忘れがちにされる
- ・障害を同僚が知らなくてヘルプしてもらえないかった（あるいは逆の立場）
- ・… etc



“機械と人間の重要な違いの一つは、機械は過負荷になると突然停止するのに対して、人間はゆっくりと機能を落とすことである。この傾向は、人間が刻々と増大する情報処理要求に直面した際に顕著となる”



## 人間も大変だ。インシデントマネジメントが必要

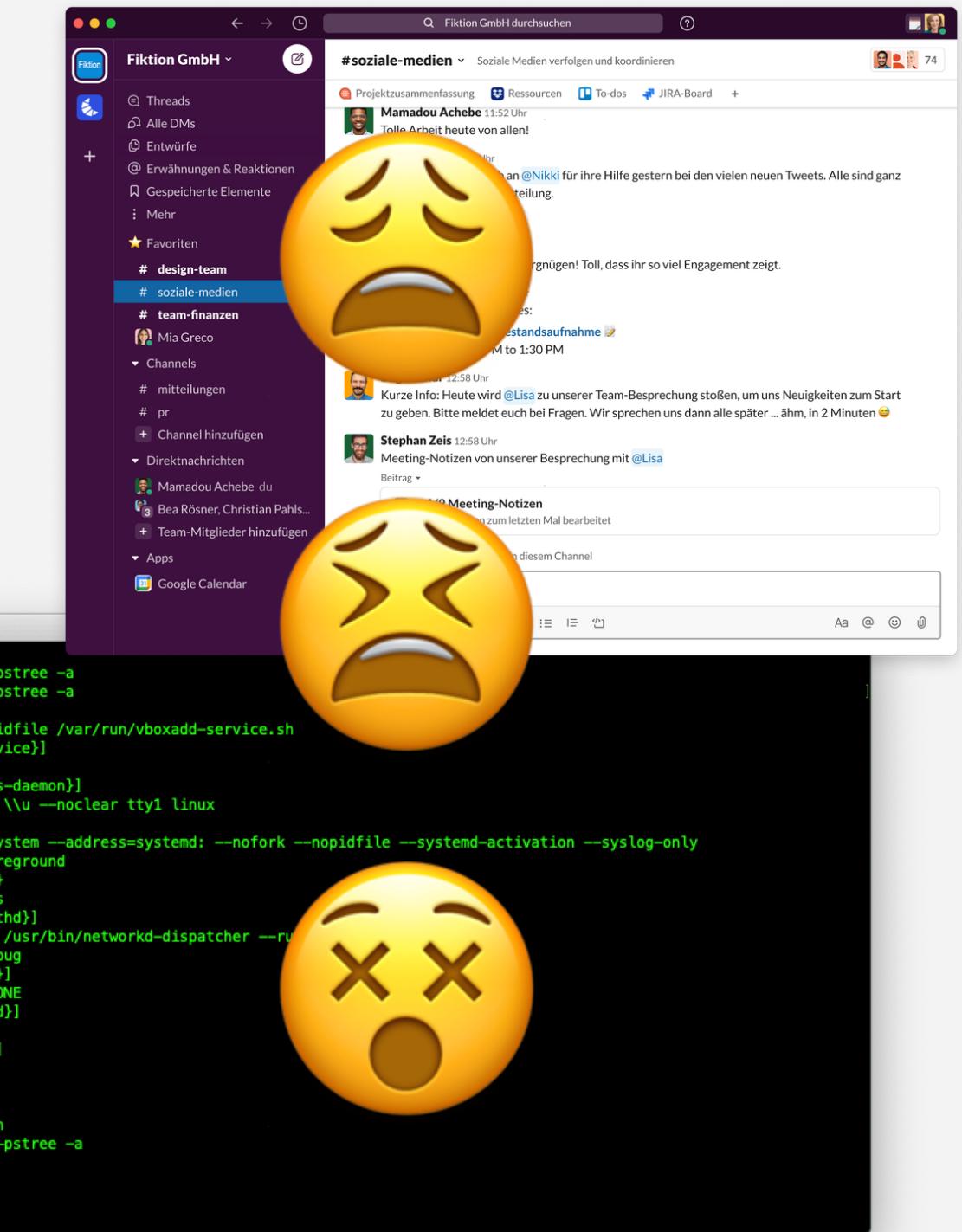
はやく復旧してビジネスへの影響を抑えたい、でも、人間にも負荷がかかっている

- 状況・方法の不確実性が高い状況下で作業するプレッシャー
- 作業の疲労や緊張からくる身体・精神へのストレス
- 多様な職種間コミュニケーションをまとめむずかしさ

<人間>に襲いかかり対応を妨げます。復旧作業のミスで二次被害も不安です。

システムを運用する組織においてはインシデント対応に立ち向かう

人間を支援する仕組みや方法論 = インシデントマネジメントが必要です





### Section 3

インシデントに  
エンジニアリングで立ち向かう

## DevSecOps Cycleとインシデント対応

ペパボでは DevSecOps Cycle として開発サイクルをモデル化しています

Incident Response もシフトレフトの対象になります

- ・ インシデント発生時のマニュアル・フロー整備
- ・ インシデント対応レベルのアセスメントの実施
- ・ インシデント対応訓練の定期実施
- ・ 脅威モデリング
- ・ ドキュメント整備(システム構成図、データモデル)
- ・ インシデントマネジメントの支援を slack bot で自動化



## Slack bot でインシデントマネジメントを支援する

**sssbot** はペパボのインシデントマネジメントを支援する slack bot です (\*)

### インシデント対応プロセスの標準化

- ・部署・サービス・パートナーに依らず統一したプロセスを提供します

### インシデント対応マニュアルの遵守

- ・ヒューマンエラーの防止、特定の人に依拠しない再現可能な対応を実施します

### インシデント対応のコミュニケーション支援

- ・フロー・マニュアルに準じた指示出し、社員のやりとりをヘルプします

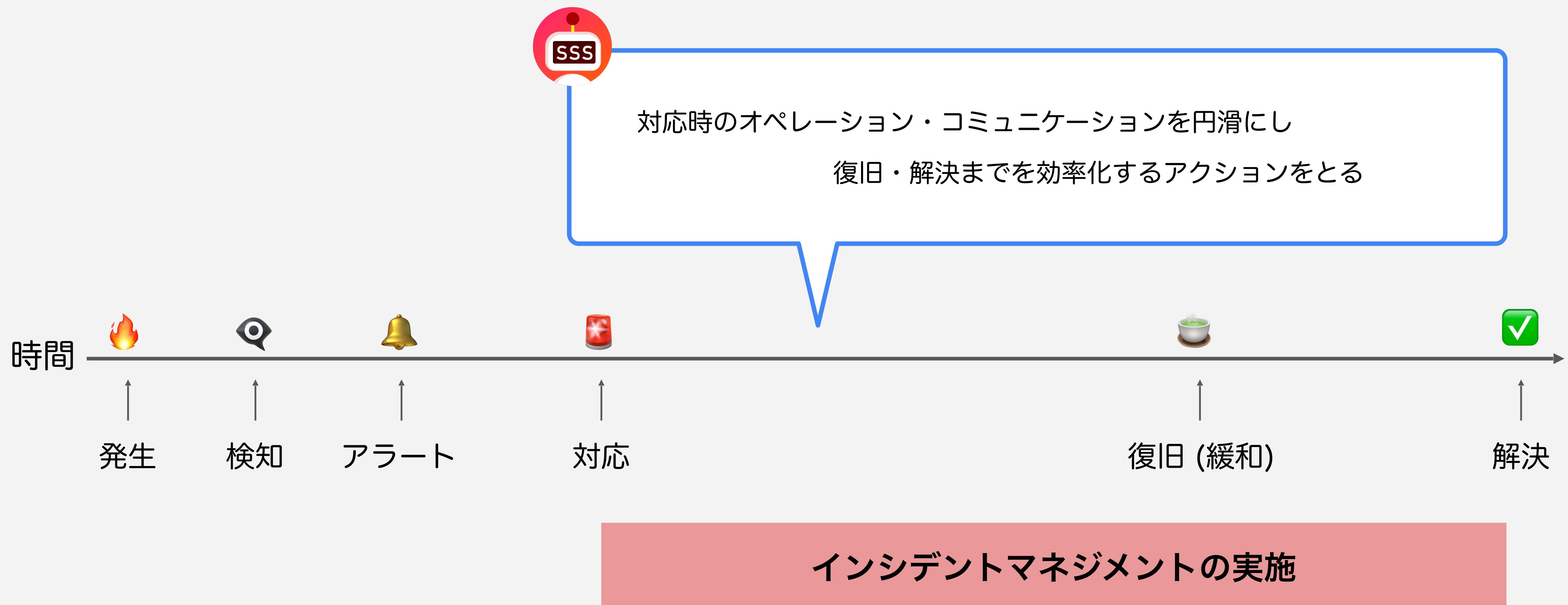


アイコンは同僚の @chocolatina 作成

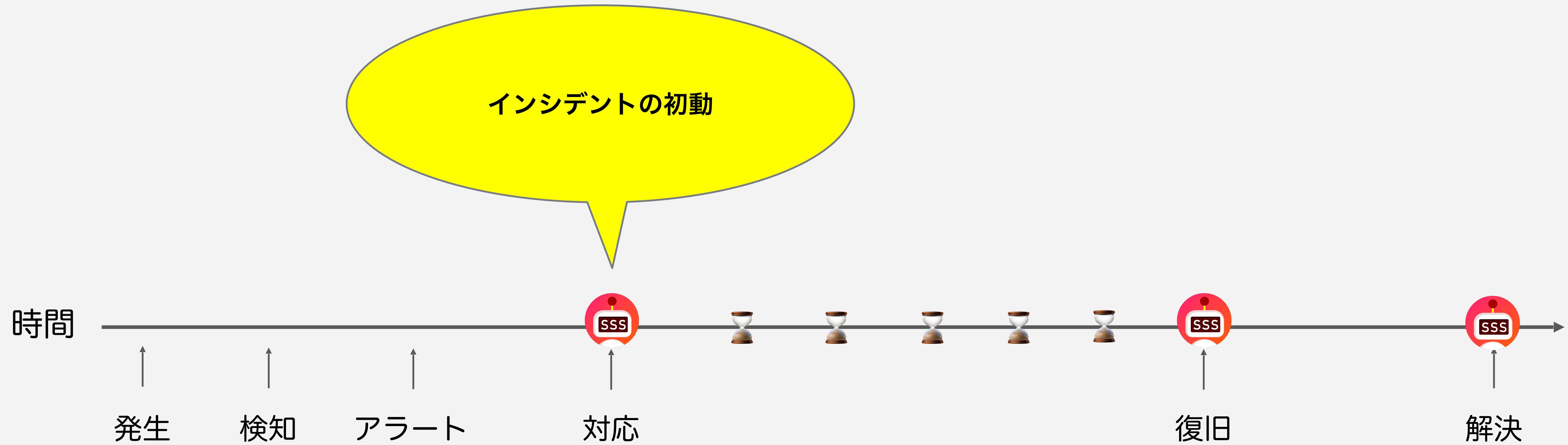
名前は セキュリティをトピックとする slack チャンネル名が #sss に由来します

\*1) 2018年後半から CLI ツールとして開発を始めて、徐々に Slack Bot (Application) として作り込んでいった

# ペパボのインシデント対応プロセスのモデル図



# インシデント対応のタイムラインと sssbot の支援



# インシデントにエンジニアリングで立ち向かう

GMOペパボ

@sssbot にメンションを飛ばすと、インシデント対応チャネル作成のボタンが現れます

hiroya 12:57 PM  
ん やばいアラートきてるな。チャネル作ろう。

@sssbot

sssbot at peacetime APP 12:57 PM  
hi @hiroya、なんでしょうか

💡 障害ですか?

sss セキュリティインシデントですか?

ヘルプの詳細は <https://example.com/sssbot-no-help> に記載しています

チャンネルを作る!

チャンネルを作る!

## 💡 bot の呼び出しは簡潔に

bot 呼び出しのコマンドをみんなに覚えてもらうのは、大変です。メンションだけで呼び出しましょう。選択したインシデントの類型によって、マニュアル・フローにのっとり sssbot が支援をしてくれます。

# インシデントにエンジニアリングで立ち向かう

GMOペパボ

フォーム入力して対応チャネルを作成します



## 💡 チャンネル名の規約

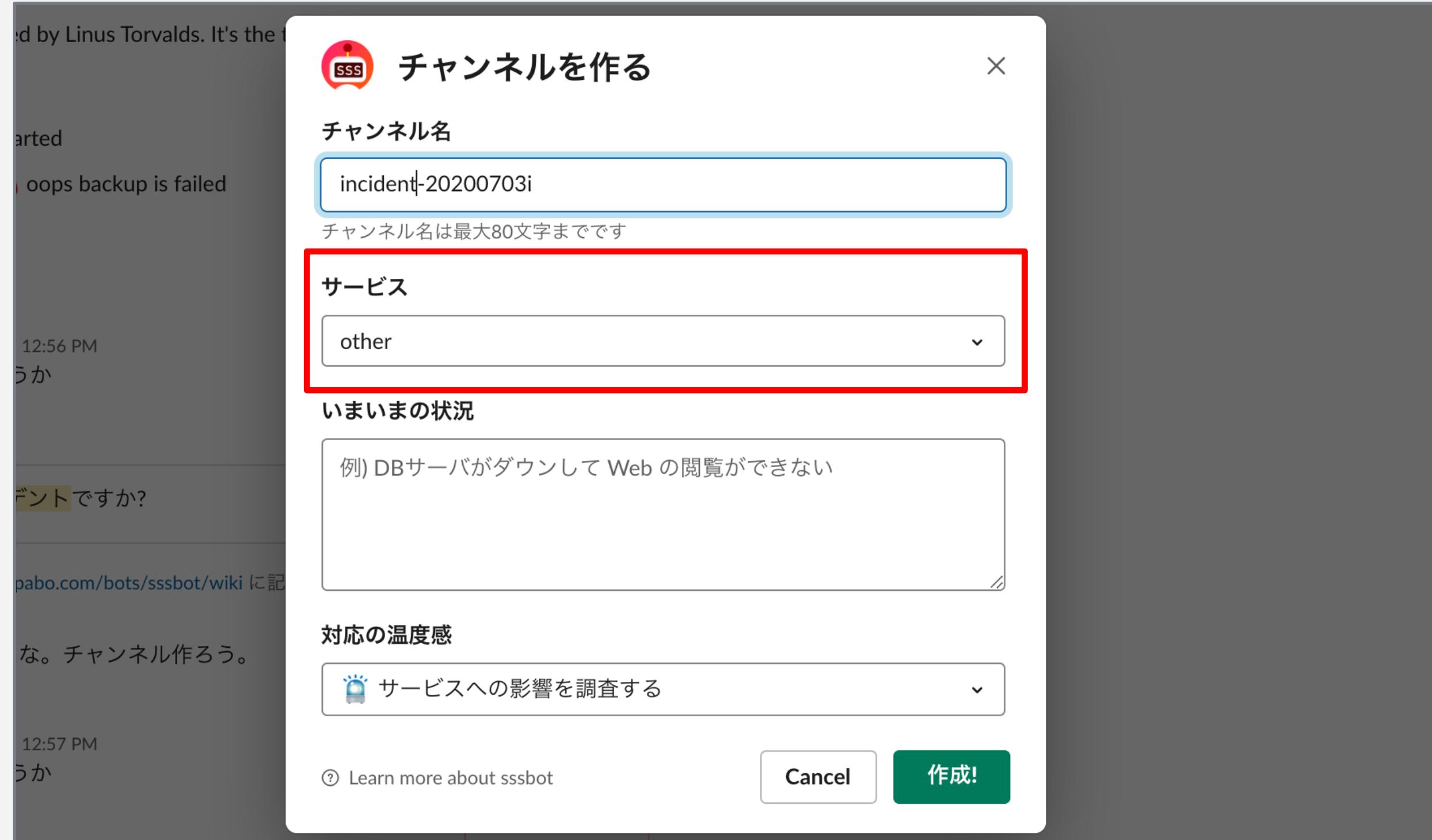
チャンネル名に規約を設けて識別できるようにします #サービス名-YYMMDDi (例: #example-20210101i)

ペパボでは複数のサービスを提供しているため、それぞれでチャネルを作れる規約とします

# インシデントにエンジニアリングで立ち向かう

GMOペパボ

サービスを選択し、初動対応チームを決定します。



初動対応チームはインシデントの初動にアサインされるグループです。部門、サービスごとに編成されます。

slack の ユーザグループ (例 @hoge-incident-response ) で定義します。カスタマーサポートやマネージャーもチームに入ります。

ユーザグループは <https://github.com/j-o-lantern0422/arisaid> を使って YAML にしてバージョン管理している

# インシデントにエンジニアリングで立ち向かう

GMOペパボ

チャンネル作成時点で把握している状況を入力します



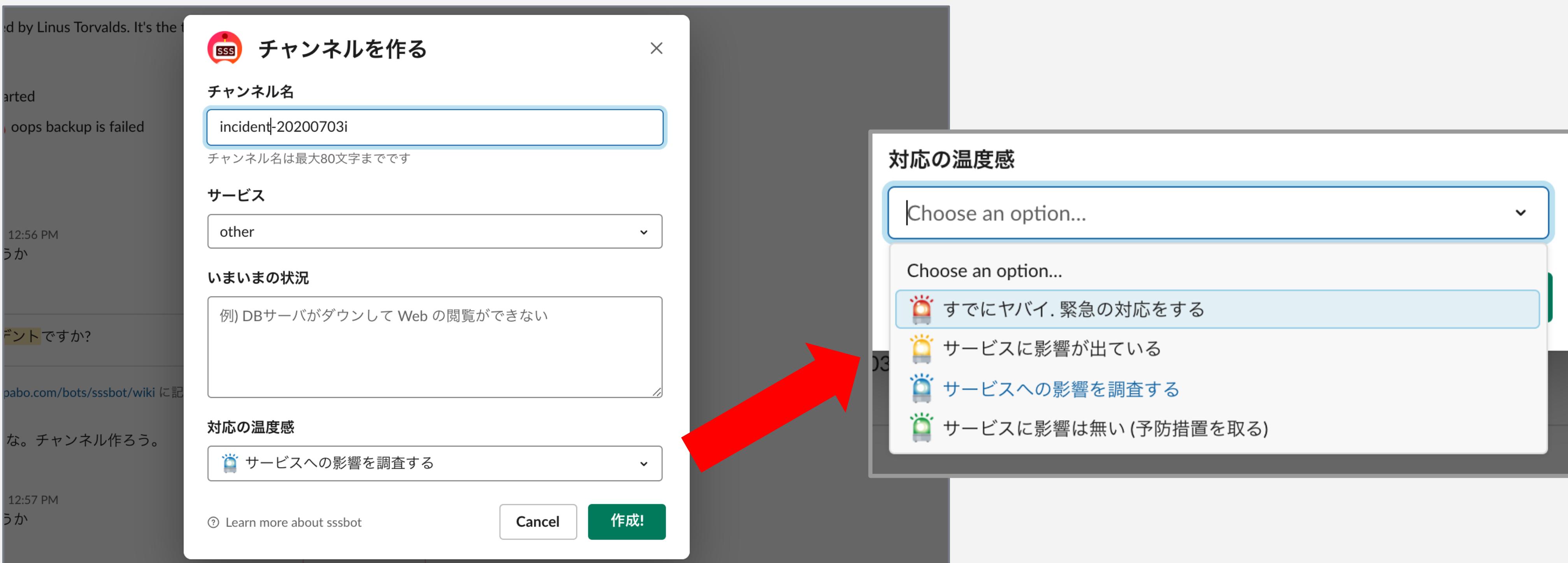
初動対応チームに状況を伝える

チャンネルを作成する人 = 状況を把握できている人と想定して、サマリをまとめてもらいます。

# インシデントにエンジニアリングで立ち向かう

GMOペパボ

温度感でインシデントの重大度(緊急度)を評価します。



## 💡 温度感で「みんな」に「やばさ」を伝える

チャンネル作成者がインシデントの重大さ・緊急度をつけます。直感に任せた評価でよく、正確さは重視しません。

マネージャーやカスタマーサポートなど、技術がよく分からぬ同僚への伝達手段でもあります

# インシデントにエンジニアリングで立ち向かう

GMOペパボ

フォーム入力後、チャンネルをセットアップします



## 一箇所に記録と対応を集約

インシデントチャネルに対応と記録を集約しましょう。進行が複数チャネルに拡散するのを避け、一元化します。

アラートが大量に流れるチャネルや、スレッドの進行はアンチパターンと(\*)しています

⚠️ アラートで対応ログやスレッドが流れてしまう、スレッド内にスレッドが作れない、スレッドの検索が煩雑、トピック等のメタデータを使えない ... 等々の理由です

# インシデント対応のタイムラインと sssbot の支援



インシデントマネジメントの実施

# インシデントにエンジニアリングで立ち向かう

GMOペパボ

チャンネル作成後、インシデントの発生を宣言します

sssbot-staging アプリ 13:38  
@hiboma がフォームを入力中です  
@hiboma インシデントレスポンスの準備をしています。お待ちください  
⚠ #sss-20200703i が作成されました

👉 sssbot-stagingさんがピン留めしました  
インシデント対応チャンネルを作りました。#sss-20200703i にjoinをお願いします

チャンネル #sss-20200703i	サービス sss
サマリ これはテストです	温度感 💡 BLUE サービスへの影響を調査する

⚠ #dev に通知しました  
⚠ #leaders に通知しました

## 💡 インシデントを宣言する

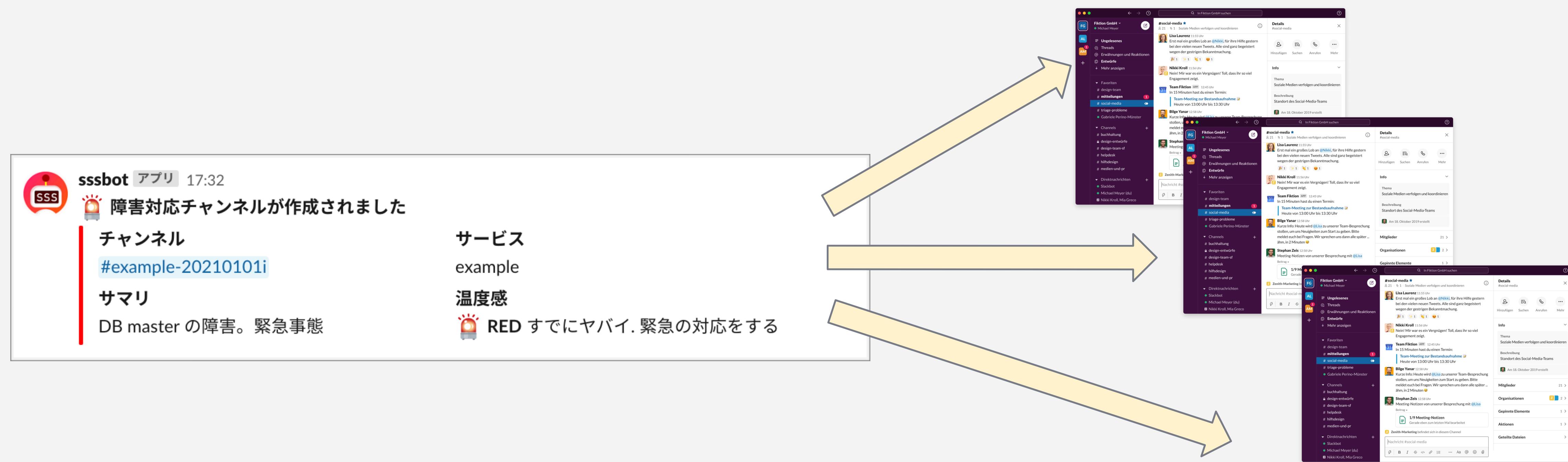
「大声」をあげて みんなにインシデントの発生を認識してもらいます。

Slack のチャンネル作成は暗黙的で気が付きにくいのを解決する UI/UX と考えています。

# インシデントにエンジニアリングで立ち向かう

GMOペパボ

インシデントの宣言は複数のチャネルにブロードキャストされます



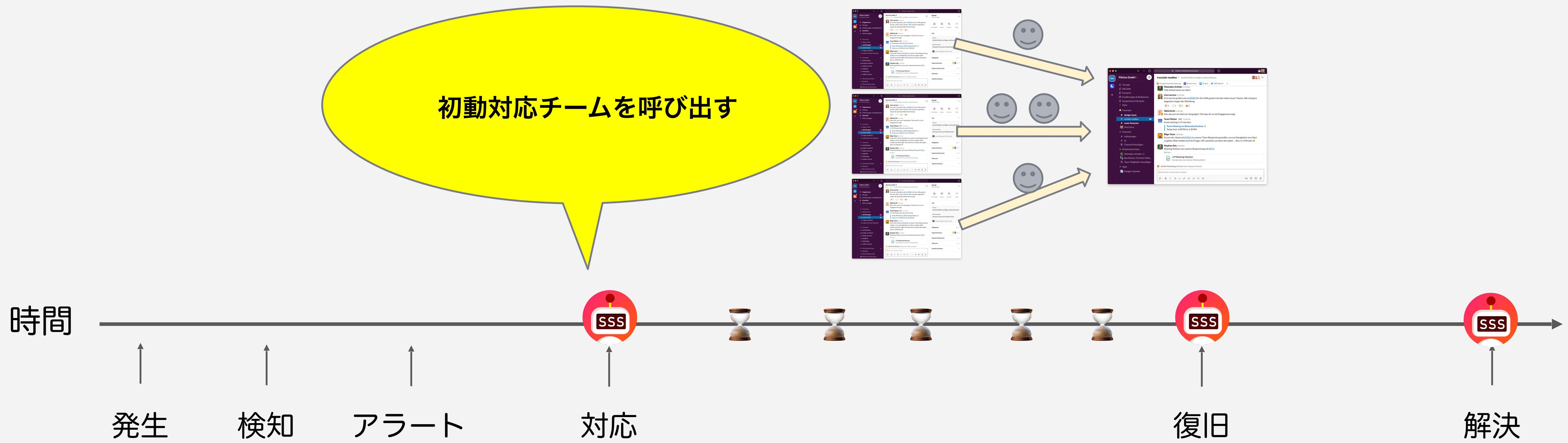
## 組織構造 (チャネル) を横断して宣言する

ペパボの slack チャンネルは 事業部・サービス・職種 単位で分かれています。

開発者チャネル、マネージャーチャネル、カスタマーサポートチャネルに通知を出します。

bot のマルチポストは少々うるさいですが、頻度の少ない通知ですし、気がつかないよりずっとよいことでしょう

# インシデント対応のタイムライン: sssbot 呼び出しのイベントモデル図



インシデントマネジメントの実施

# インシデントにエンジニアリングで立ち向かう

GMOペパボ

インシデント対応チャネルに初動対応チームを invite します



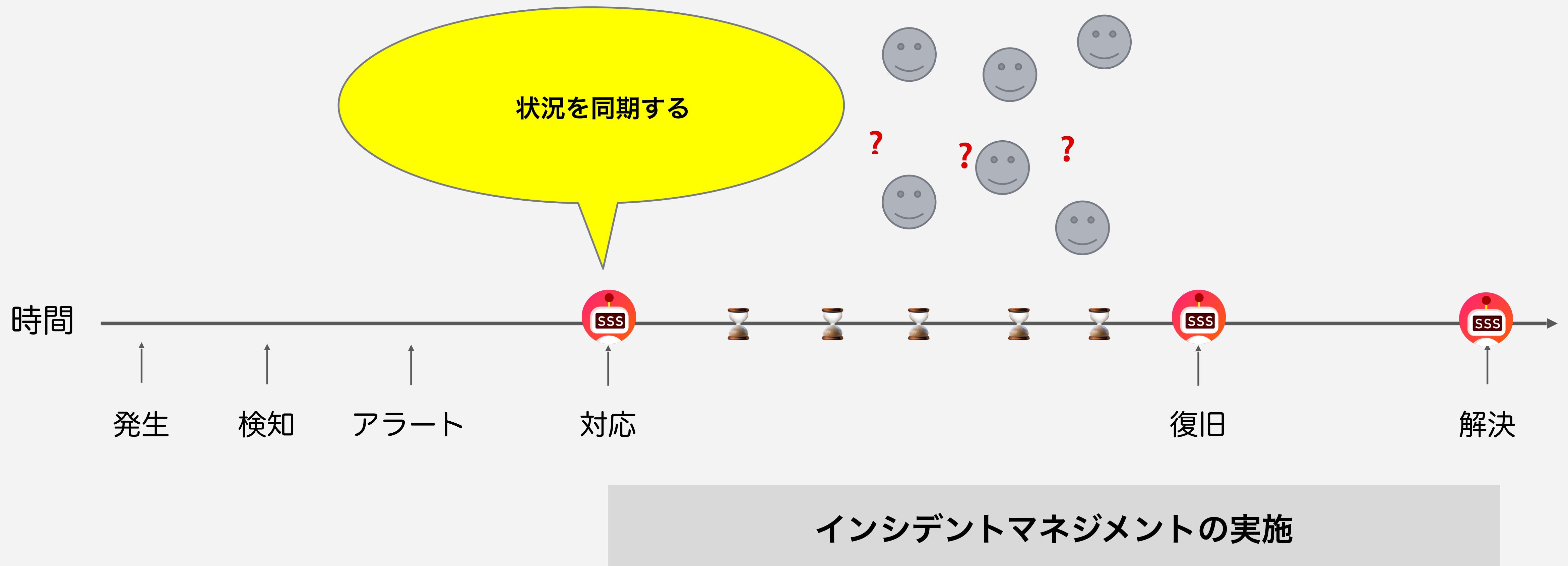
## 必ず呼ぶ人も bot に任せる

インシデントの初動にはいるチームを sssbot が自動で呼び出します。

CTO, VPoE, セキュリティ対策室も呼び出して重大なインシデントの見逃しを防ぎます。

たくさんの「人を呼び出す」のは面倒で 心理的障壁の高い作業です。bot に任せてしまいましょう。

# インシデント対応のタイムライン: sssbot 呼び出しのイベントモデル図



# インシデントにエンジニアリングで立ち向かう

GMOペパボ

チャンネル作成者にメンションを飛ばし「いまいまの状況」を書くように促します。

 **sssbot** アプリ 13:40  
@hiboma チャンネルを作ってくれてありがとうございます。いまいまだどんな状況でしょうか?

- 影響範囲は? ( どんな機能や画面で影響でてる?)
- ユーザー対応の要否は? ( お知らせ、お問い合わせ、ソーシャル対応は必要?)
- 原因・復旧の目処はつきそうか?

## 💡 まずは状況を伝えて、みんなで足並みを揃える

突如、チャンネルに呼び出される初動対応チームは、そもそも何が起きたのかすら把握していない人もいます。

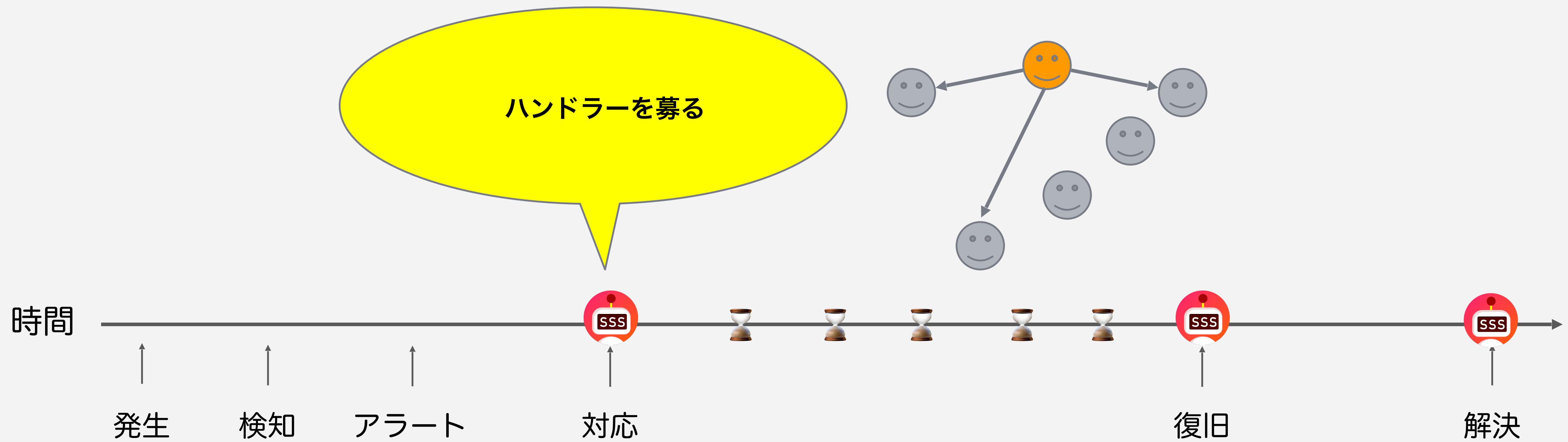
チャンネルを作成した人を、初動の時点で状況を理解している人と想定して、bot から状況説明を促します。

## 💡 避けたいアンチパターン

「チャンネル作成者が、状況を説明せずに復旧作業にはいり、周囲を置いてきぼりにする」

「技術職はサマリを読んで状況を理解したが、カスタマーサポートやマネージャーらは用語が分からず理解できていない」

# インシデント対応のタイムライン: sssbot 呼び出しのイベントモデル図



インシデントマネジメントの実施

# インシデントにエンジニアリングで立ち向かう

GMOペパボ

チャンネル作成後から数分すると sssbot がインシデントハンドラーの決定を促します

sssbot-staging アプリ 14:42  
インシデントハンドラーはどなたですか?  
ハンドラーはわたしです

@hiboma さんが ハンドラーはわたしです を押しました  
@hiboma さんがハンドラーですね  
Slack でインシデントを検知した際のメッセージは分かりますか?  
\* 監視システムがアラートを通知した際のメッセージ  
\* パートナーの誰かがインシデントに気が付いた時のメッセージ  
教えてください  
# コマンドの例  
@sssbot インシデントの検知 <https://example.com/slack/p1634721688039800>

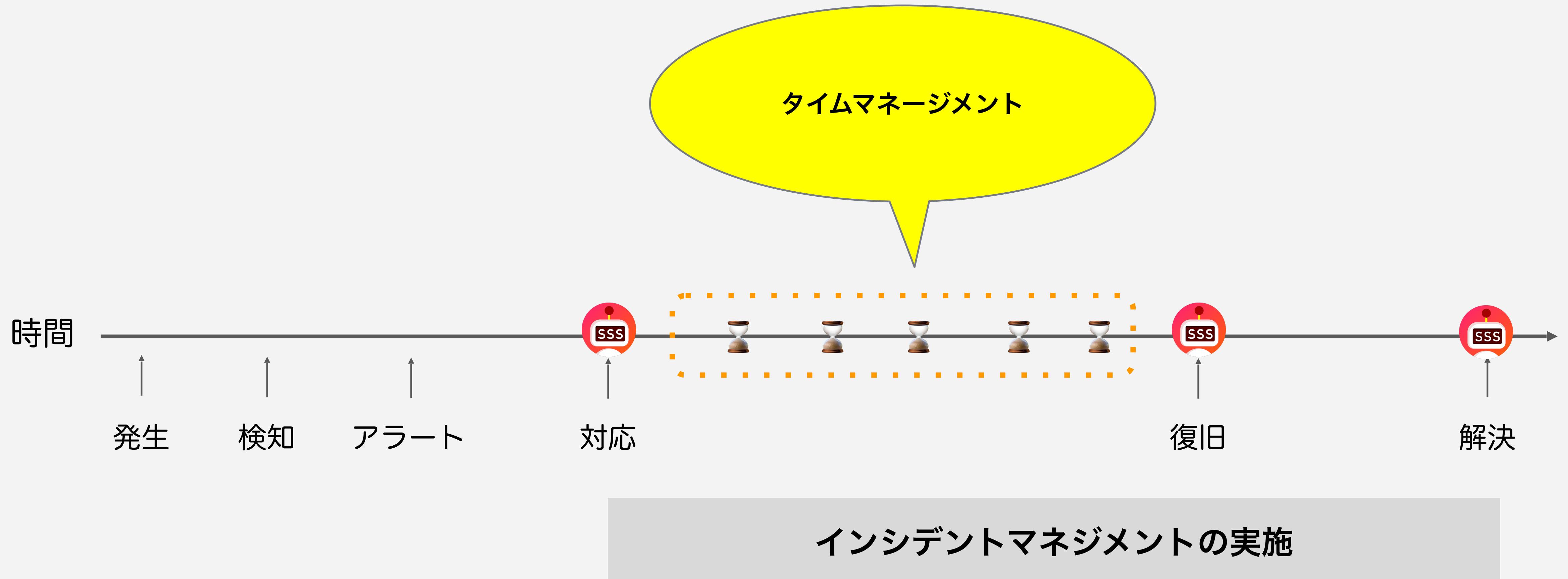
コミュニケーション方法を選択・確立しましょう  
\* このまま Slack のチャットで進行しますか?  
\* ビデオ会議で進行しますか? ( は [ここから作れます](#))  
\* 後から見た方が理解できるようにログを残しましょう

## 💡 ハンドラーをたてて対応する

ペパボでは インシデントハンドラー(=インシデントコマンダー) をたてて進行を委ねます。

ハンドラーにコミュニケーションを引導するリーダーシップを発揮してもらいます

# インシデント対応のタイムライン: sssbot 呼び出しのイベントモデル図



# インシデントにエンジニアリングで立ち向かう

GMOペパボ

15分ごとにタイムキーパーが起動して経過時間を知らせます

The screenshot shows a mobile application interface for 'Timekeeper-kun'. At the top, there's a profile icon with 'sss' and a timer icon, followed by the text 'タイムキーパーくん APP 11:00 AM'. Below this, a yellow box displays the message 'インシデントの検知から 56分 経過しています'. To the right is a dropdown menu labeled 'コマンド'. The main content area contains the following information:

- ハンドラー: [@hiboma](#)
- 事象レベル: 1
- 検知: 2020-07-01 10:03:53 +0900
- 初動: 2020-07-01 10:15:39 +0900
- 復旧: 2020-07-01 10:56:42 +0900

## 💡 時限のイベントで介入する

インシデントが対応深刻化・長期化すると、疲労やプレッシャーも重なり時間を失念しがちです。

定期的な状況判断・意思決定を促すために、タイムキーパーが介入し、時間を伝えます。

## 💡 15分なのは何故か？

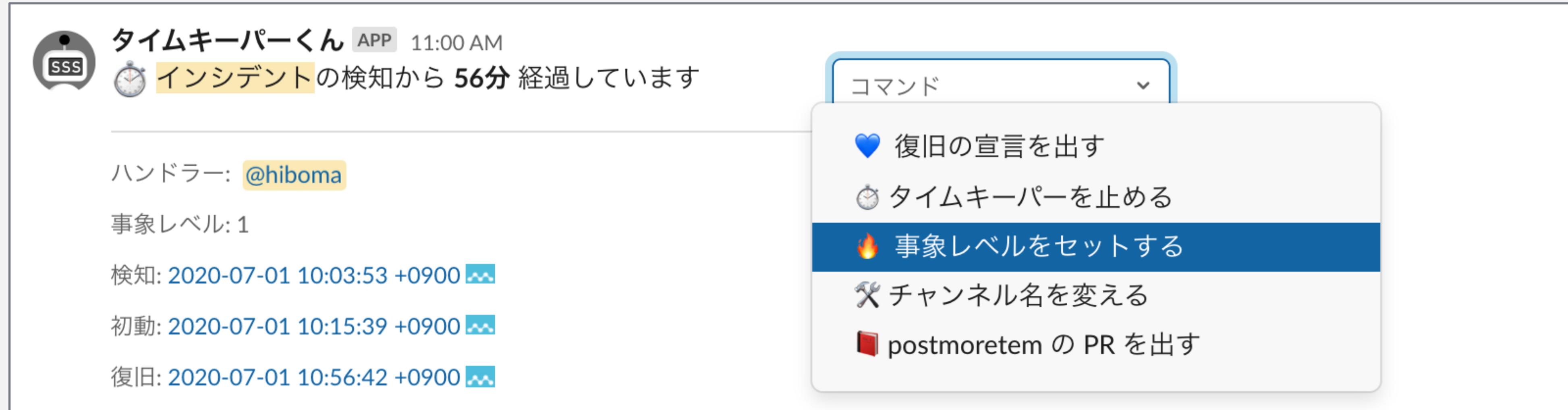
「障害情報の掲載、外部組織への定期連絡を行う際に 30分を区切りにして

アクションをとることが多く15分周期で調整のアクションをとるとちょうどいい」と、社内フィードバックを受けた設定です。

# インシデントにエンジニアリングで立ち向かう

GMOペパボ

タイムキーパーはコマンド実行の UI を備えます



💡 コマンドレスで呼び出せる

タイムキーパーにコマンド実行のダッシュボード UI をもたせています。

# インシデントにエンジニアリングで立ち向かう

GMOペパボ

事象レベル(\*) を決定すると、sssbot はレベルに応じたマニュアルを指示します。

The screenshot shows a message from the bot 'sssbot-staging' (@hiboma) at 15:29. The message content is as follows:

sssbot-staging アプリ 15:29  
@hiboma さんが 🔥 事象レベルをセットする を押しました

事象レベルを選択してください

事象レベルは インシデント対応マニュアル - 事象レベル基準表 に準じます

#sss-ixf7h-20200623i の事象レベルが 3 🔥🔥🔥 に宣言されました

@hiboma 事象レベルの宣言に合わせて確認をお願いします

- 障害情報は出しましたか?
- SNS (Twitter/Facebook) のお知らせは出しましたか?

もし、まだでしたら担当者をアサインしましょう! インシデント対応マニュアル 事象レベル3対応手順 で TODO 確認もしましょう

A yellow callout box points to the text 'ハンドラーに指示出しします' (Handler will be instructed) in the bottom right corner.

## 💡 定型の指示出しは bot に言わせる

マニュアルに沿った指示出しは、人間が行うと煩わしいものです。ボットに任せてしまいましょう。  
忘れがちなタスクもボットにリマインドしてもらいましょう。

注) GMOペパボのインシデント対応マニュアルにてインシデントの規模に応じた事象レベルを付けることが定められている

# インシデントにエンジニアリングで立ち向かう

GMOペパボ

事象レベルの決定も複数チャネルにブロードキャストします

sssbot-staging アプリ 15:29  
@hiboma さんが 🔥 事象レベルをセットする を押しました

事象レベルを選択してください

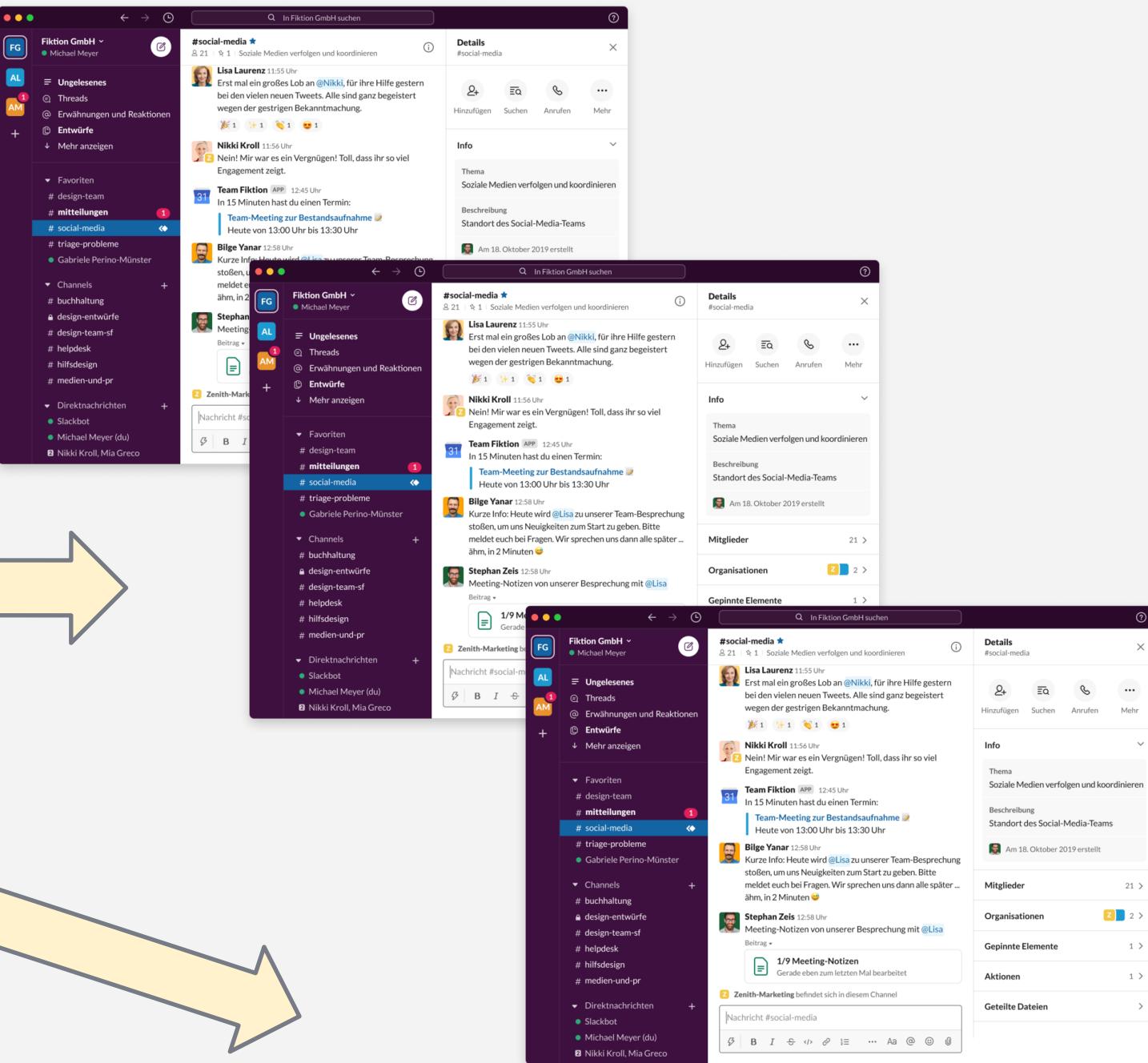
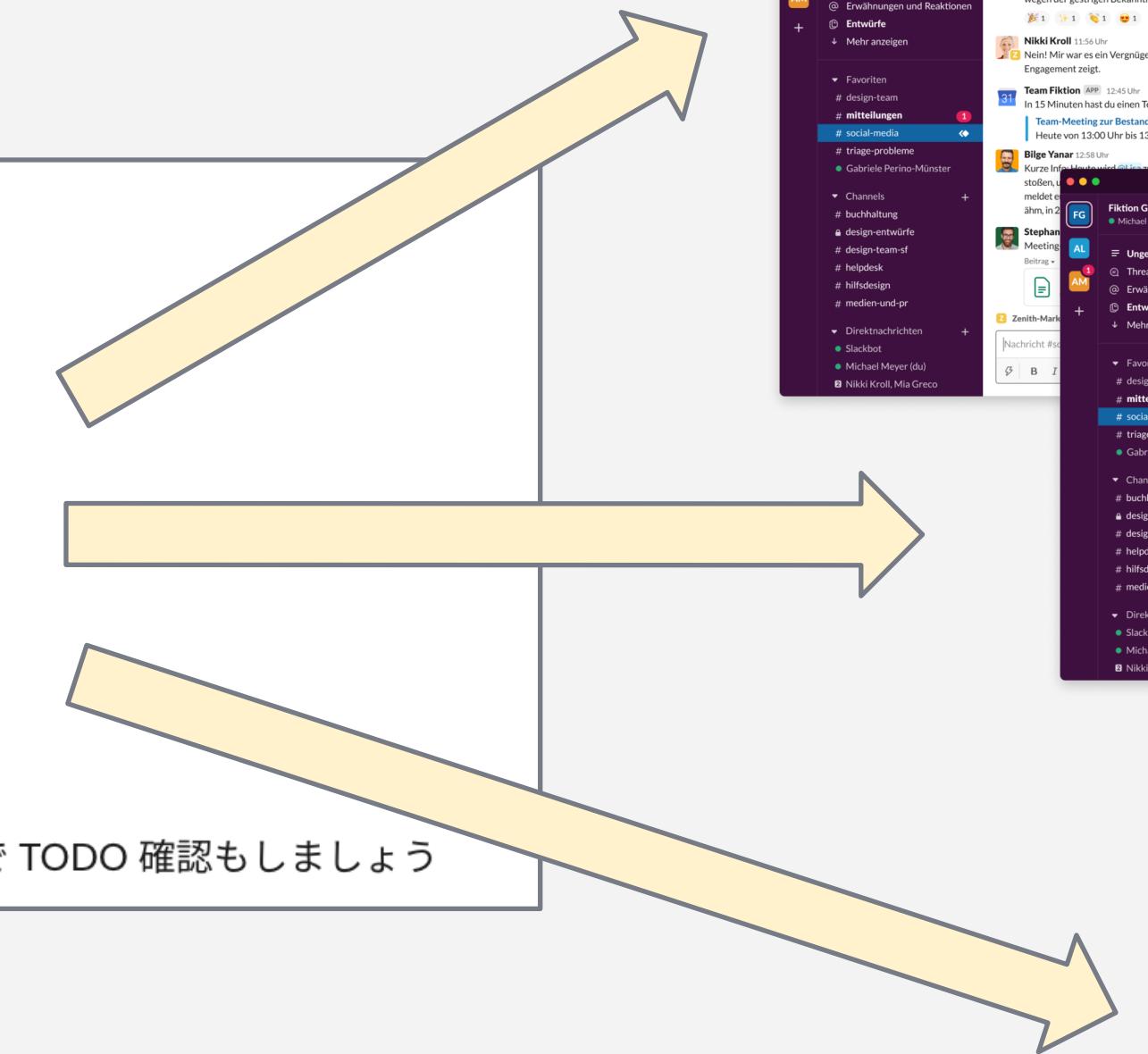
事象レベルは インシデント対応マニュアル - 事象レベル基準表 に準じます

#sss-ixf7h-20200623i の事象レベルが 3 🔥🔥🔥 に宣言されました

@hiboma 事象レベルの宣言に合わせて確認をお願いします

- 障害情報は出しましたか?
- SNS (Twitter/Facebook) のお知らせは出しましたか?

もし、まだでしたら担当者をアサインしましょう! インシデント対応マニュアル 事象レベル3対応手順 で TODO 確認もしましょう

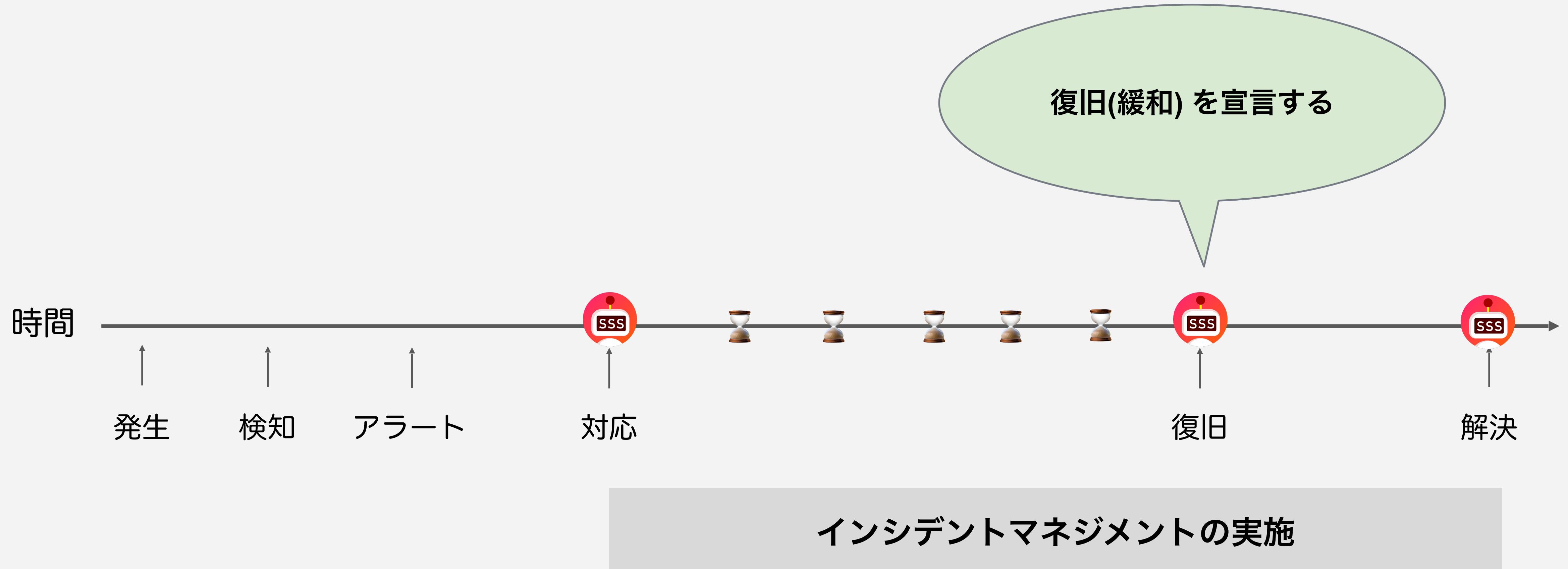


## 報告を bot に任せる

開発者チャネル、カスタマサポートチャネルなどにマルチポストする作業を bot に任せます。

マネージャー・リーダー・取締役がいる private チャンネルにも通知が飛びます。

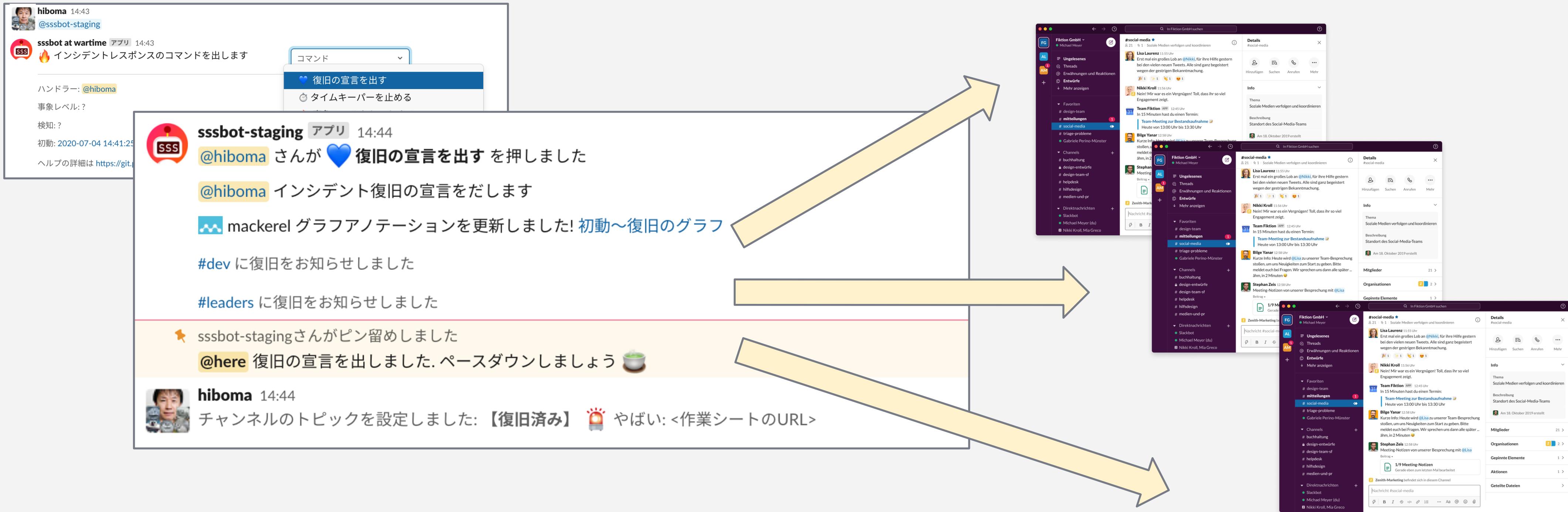
# インシデント対応のタイムライン



# インシデントにエンジニアリングで立ち向かう

GMOペパボ

インシデントから復旧(緩和)したら宣言をブロードキャストします



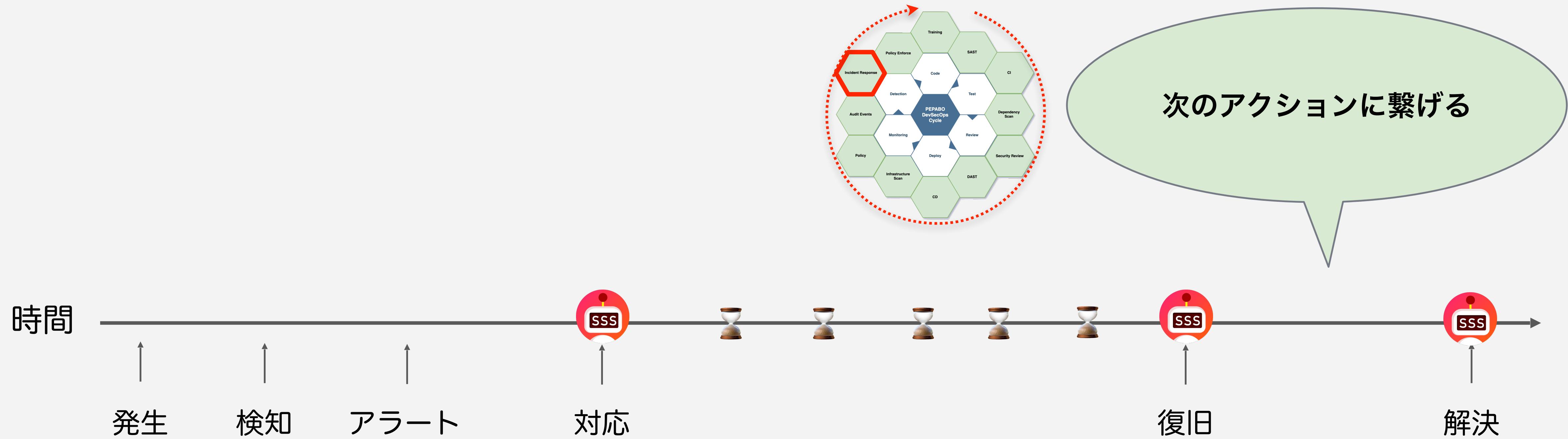
💡 復旧の宣言を出し、対応の優先度を下げる

インシデント対応は 非常に高い優先度で割り込むタスク とみなせます。

対応がひと段落したら復旧を宣言して、対応の優先度を下げましょう。

また、宣言をブロードキャストすることで「チャンネル内の人間だけが復旧を認識していた」というアンチパターンを避けます。

# インシデント対応のタイムライン

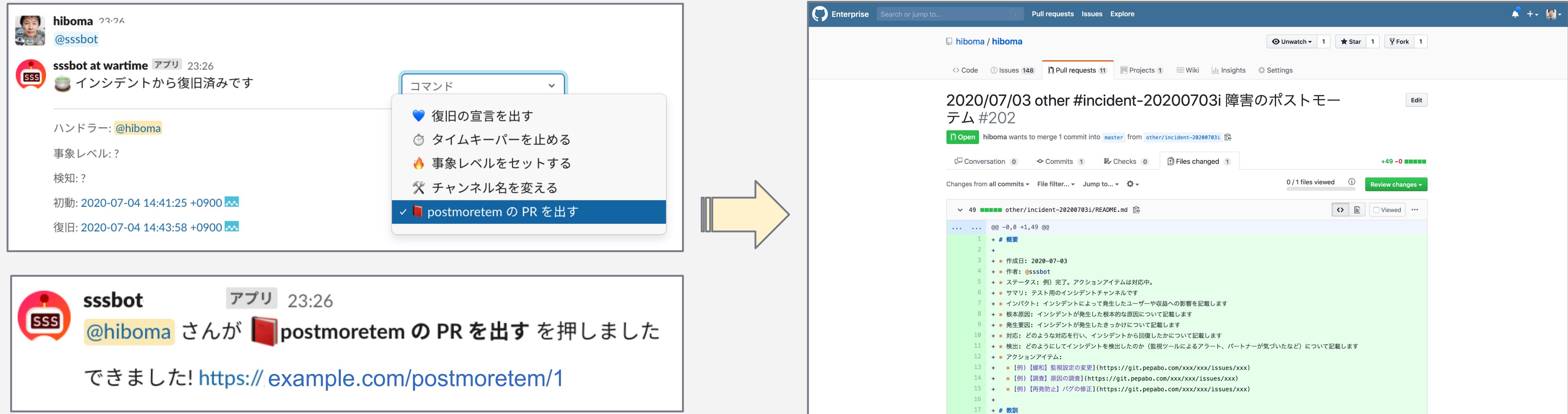


インシデントマネジメントの実施

# インシデントにエンジニアリングで立ち向かう

GMOペパボ

sssbot はポストモーテムの作成も支援します。 Pull Request も自動作成できます。



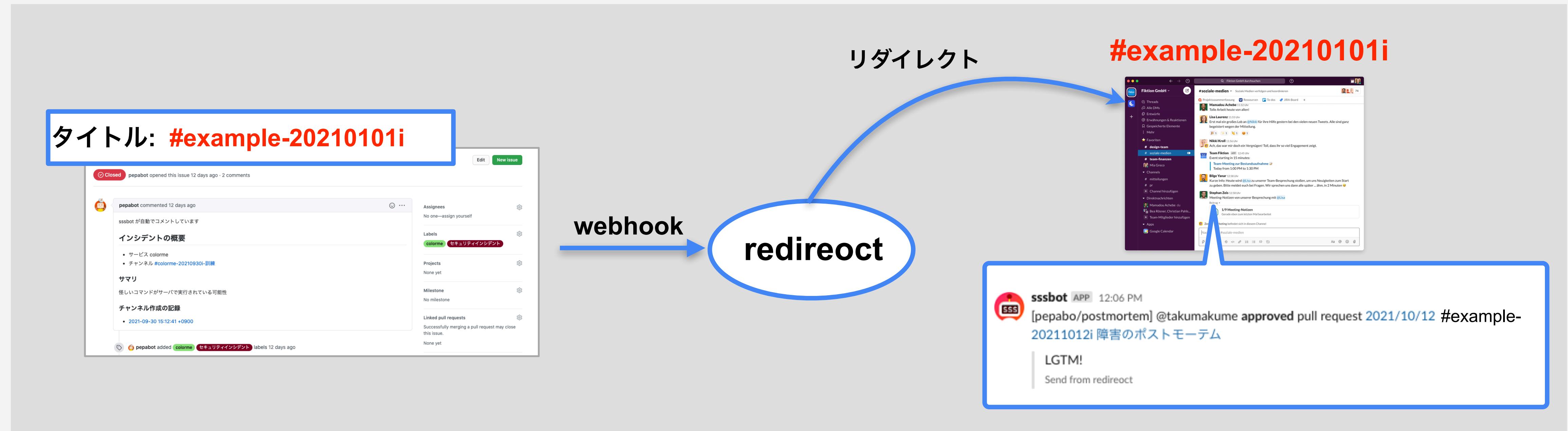
## 💡 ドキュメントのテンプレートを PR する

sssbot が記録したサービス名、サマリ、イベント発生の時刻を埋め得た markdown のテンプレートを PR します  
ポストモーテム (\*1) の pull request を作成し、ふりかえり・根本原因分析・再発防止の検討・実施へつなげます。

ポストモーテムのフォーマットは『SRE サイトリライアビリティエンジニアリング』をベースにしています

postmoretem の作成はそこそこの負担になります。軽微な障害では postmortem の作成を省くのもよいと思います (SLI/SLO 等も加味して評価する)

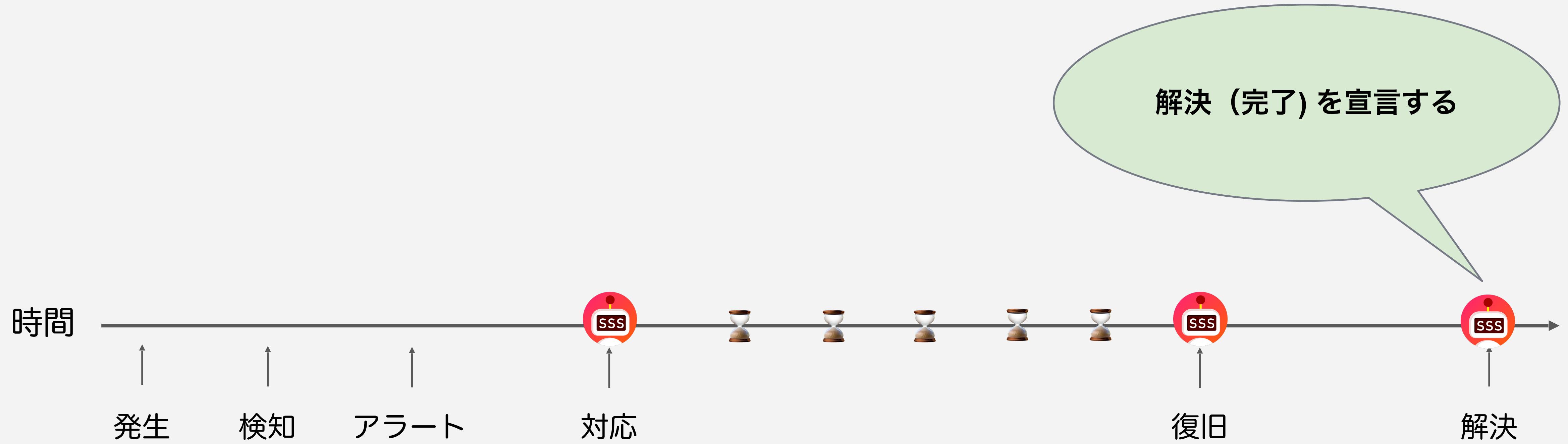
チャンネルを動的に作るので postmortem リポジトリからインシデントチャンネルに webhook 通知するには工夫が必要です  
issue/PR タイトルにチャンネル名を書くと任意のチャンネルに通知できるアプリを作って解決しています



postmortem の Pull Request についてのレビュー・コメントもインシデントチャンネルに通知して  
事後対応のコミュニケーションもチャンネルに集約します

- git push では branch 名からチャンネル名を判定して リダイレクトする ( original/example-20210101i )
- GitHub の通知をリダイレクトするので redirect + octopus = redireoct です。Sinatra で作りました。

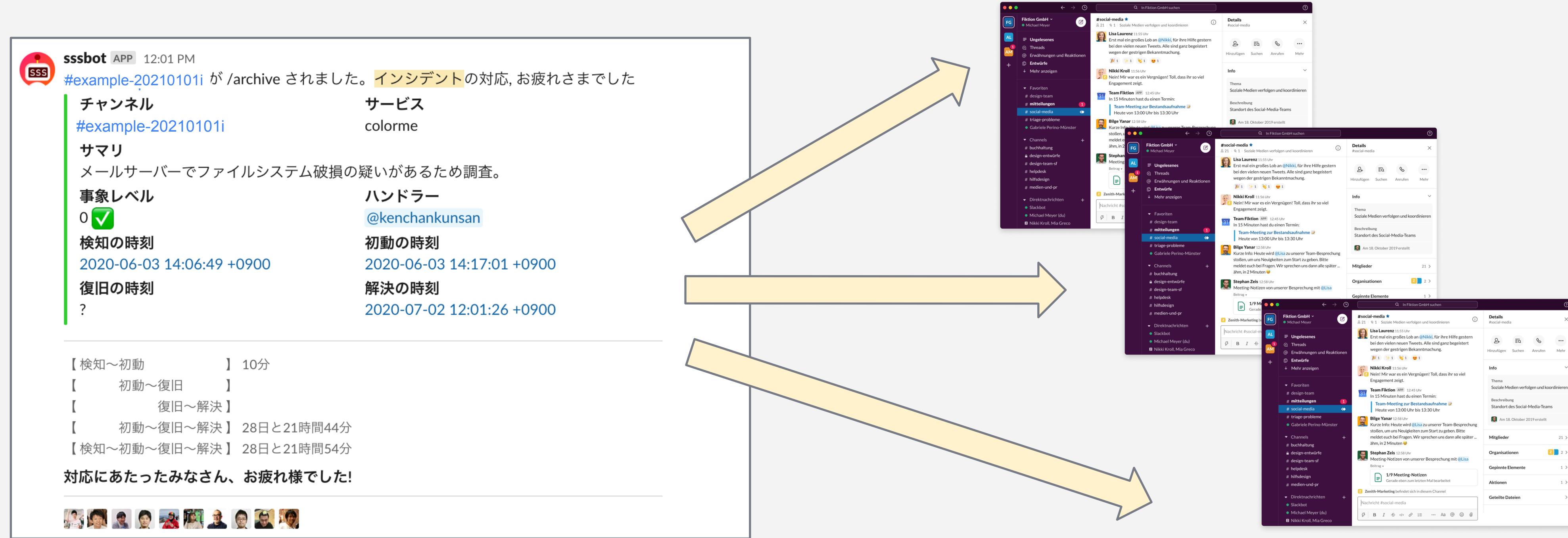
# インシデント対応のタイムライン: sssbot 呼び出しのイベントモデル図



# インシデントにエンジニアリングで立ち向かう

GMOペパボ

postmortem や再発防止の完了後は、対応チャネルを archive して、解決(完了)の宣言をブロードキャストします



💡 全ての作業を終えたら、宣言する

復旧後の残作業は長期化しやすく、時間が経過してインシデントの存在から忘れられがちになります。

最後まで作業してくれた人にお疲れ様の声もかけてあげましょう!!!!

# インシデントにエンジニアリングで立ち向かう

GMOペパボ

インシデントの解決 = チャンネルを archive するまでを後押しするリマインド機能も実装しています  
チャンネルのメッセージが一ヶ月間途絶えていたらリマインドを入れるようにしています

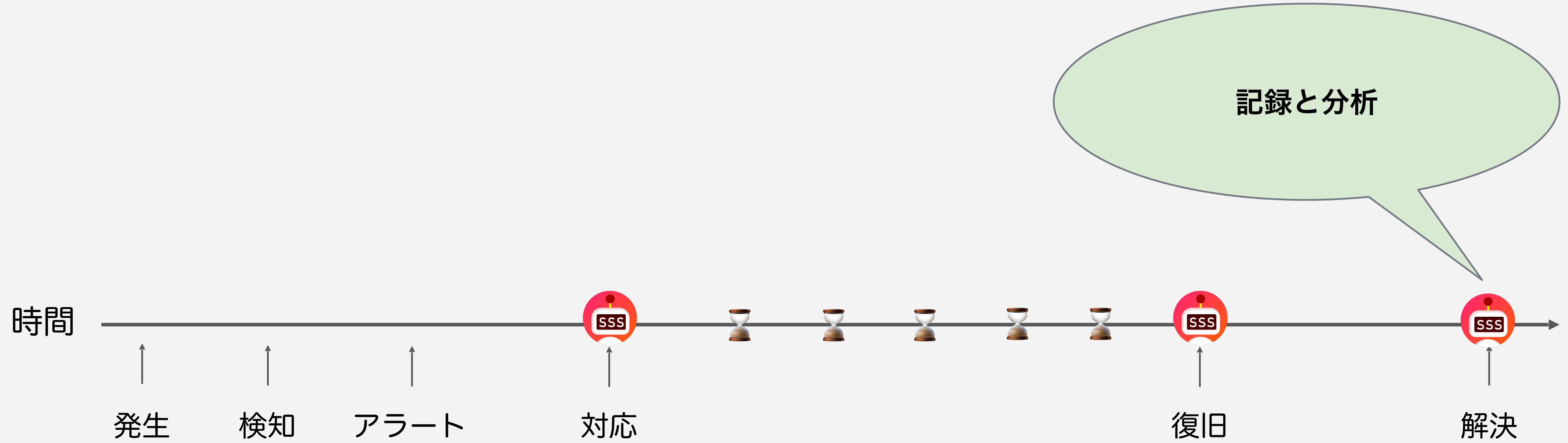
The screenshot shows a messaging app interface with three participants:

- sssbot** (APP) at 11:10 AM: 最後のメッセージから一ヶ月以上経過したのでリマインドです。インシデントの残作業はありますか? それとも /archive しますか?
- foo** at 11:51 AM: こちらの件ですが、からその後の回答はありましたでしょうか?
- bar** at 1:25 PM: 報告書に反映中となりますので、9/1 19:00までにバック予定です 🙏

## 💡 解決 = チャンネルのアーカイブまでを推し進める

インシデントによっては残作業で、数ヶ月をして長期化するケースがあり、フォローします  
単に、チャネルの archive を忘れているだけなら archive するように促します

# インシデント対応のタイムライン: sssbot 呼び出しのイベントモデル図



# インシデントにエンジニアリングで立ち向かう

GMOペパボ

インシデント発生時に自動で issue を作成し、インシデントの記録をとっています

The image consists of two side-by-side screenshots of a GitHub Enterprise interface. The left screenshot shows a list of issues in a repository named 'pepabo / postmortem'. The right screenshot shows a detailed view of a specific issue titled '#colorme-20210930i-訓練 #886'.

**Left Screenshot (Issues List):**

- Header: Enterprise, Search or jump to..., Pull requests, Issues, Trending, Explore.
- Repository: pepabo / postmortem (Internal).
- Issue Count: Issues 110, Pull requests 25.
- Filters: Filters dropdown, search bar (訓練 intitle), Labels 32, Milestones 0, New issue button.
- List of issues:

  - #886 by pepabot was closed 16 days ago. Labels: colorme, セキュリティインシデント.
  - #884 by pepabot was closed 18 days ago. Review required. Labels: goope, 事象レベル2 🔴🔴, 閉じる.
  - #883 by pepabot was closed 18 days ago. Labels: goope, 事象レベル2 🔴🔴, 閉じる.
  - #570 by pepabot was closed on 2 Feb. Review required. Labels: goope, セキュリティインシデント, 事象レベル2 🔴🔴.
  - #569 by pepabot was closed on 2 Feb. Labels: goope, セキュリティインシデント, 事象レベル1 ⚠️.
  - #568 by pepabot was closed on 2 Feb. Review required. Labels: goope, セキュリティインシデント, 事象レベル3 🔴🔴🔴.
  - #566 by pepabot was closed on 2 Feb. Labels: goope, セキュリティインシデント, 事象レベル3 🔴🔴🔴.

**Right Screenshot (Issue Detail #886):**

- Header: Enterprise, Search or jump to..., Pull requests, Issues, Trending, Explore.
- Repository: pepabo / postmortem (Internal).
- Issue: #colorme-20210930i-訓練 #886 (Closed, 2 comments).
- Comments:

  - pepabot 17 days ago: sssbot が自動でコメントしています
  - pepabot 17 days ago: インシデントの概要
    - サービス colorme
    - チャンネル #colorme-20210930i-訓練
  - pepabot 17 days ago: サマリ  
怪しいコマンドがサーバで実行されている可能性
  - pepabot 17 days ago: チャンネル作成の記録
    - 2021-09-30 15:12:41 +0900
  - pepabot 17 days ago: pepabot added colorme, セキュリティインシデント labels 17 days ago
  - pepabot 17 days ago: sssbot が自動でコメントしています
  - pepabot 17 days ago: 復旧の記録
    - 2021-09-30 15:48:32 +0900

- Side panel: Assignees - assign yourself, Labels (colorme, セキュリティインシデント), Projects, Milestone, Linked pull requests, Notifications (Customize, You're receiving notifications because you're watching this repository, Unsubscribe), 0 participants, Lock conversation, Pin issue, Transfer issue, Delete issue.

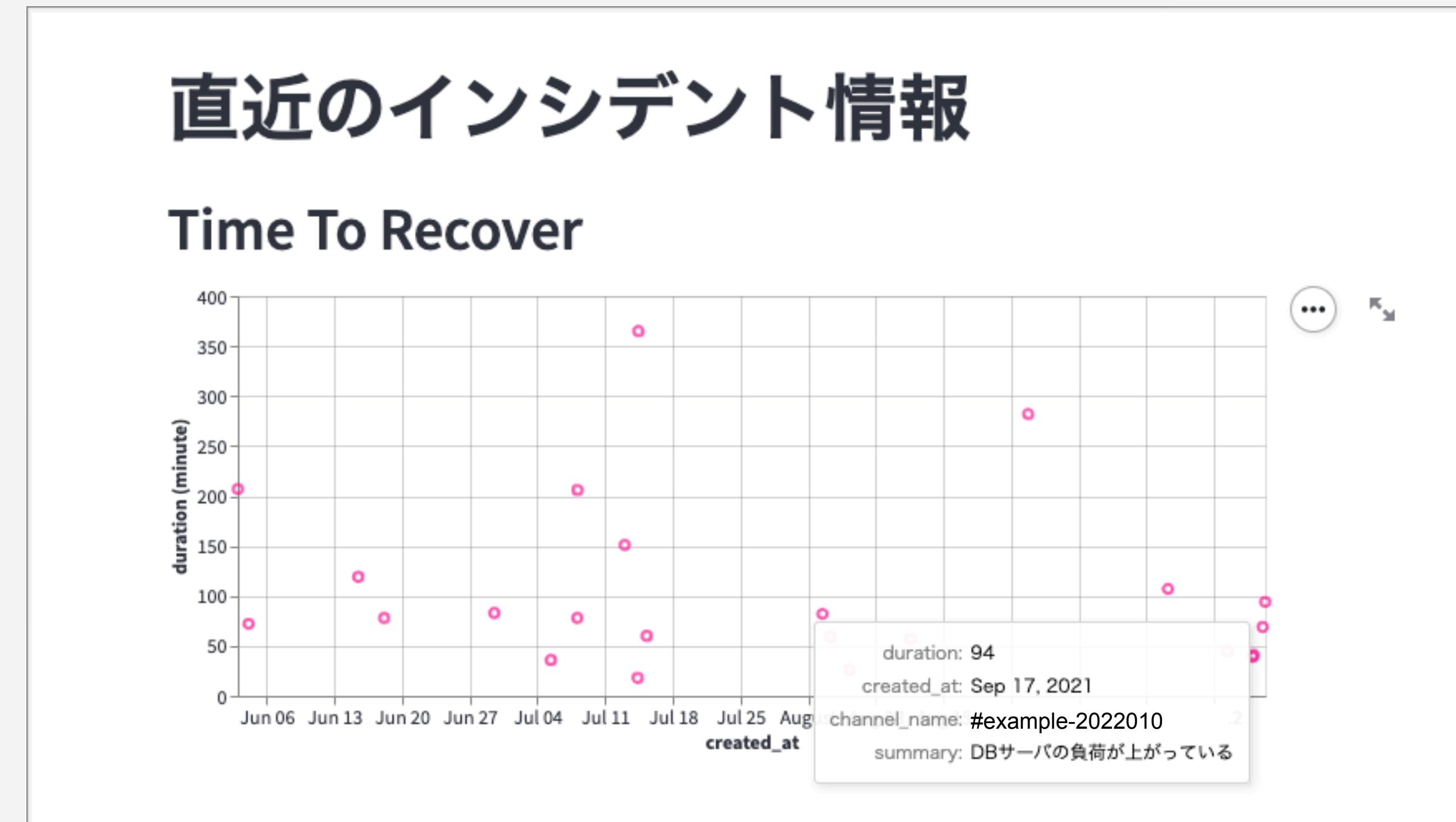
Slack の検索 UI でインシデント対応チャンネルを遡り把握するのは少々煩雑です。

GitHub Enterprise の issue にストック型のデータとして残して、参照しやすくなります。

サービス名、事象レベル、インシデントの類型でラベルを付けて、分類も自動化しています。

sssbot の記録を分析・可視化に用いることができます

対応～復旧 等に要した時間を可視化した例です (Four keys のサービス復元時間に相当する)



インシデントのレベル/温度感や、対応した人員(チャット量)など各種メタデータを採取してさらに応用ができそうですね  
ダッシュボードは弊社のデータ基盤チームが実装してくれました!

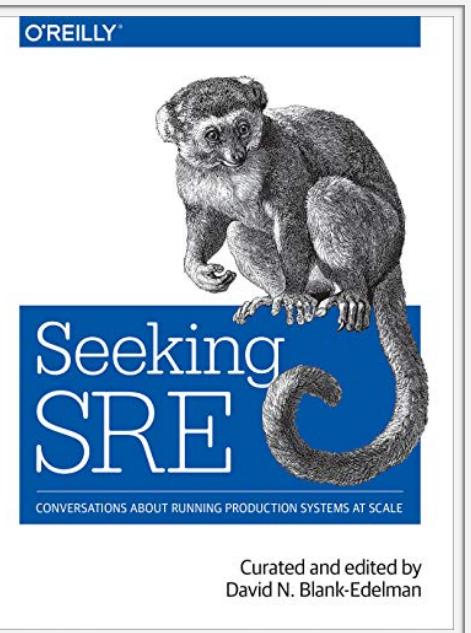
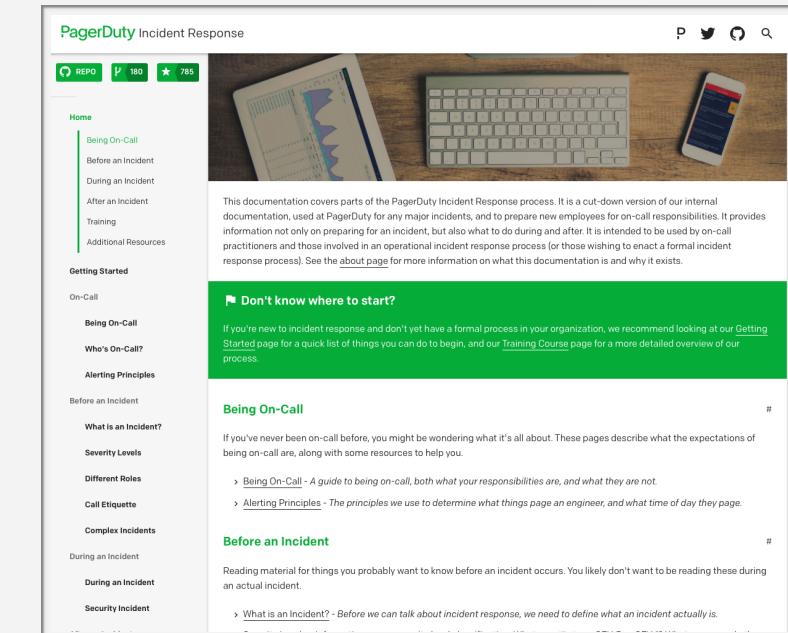
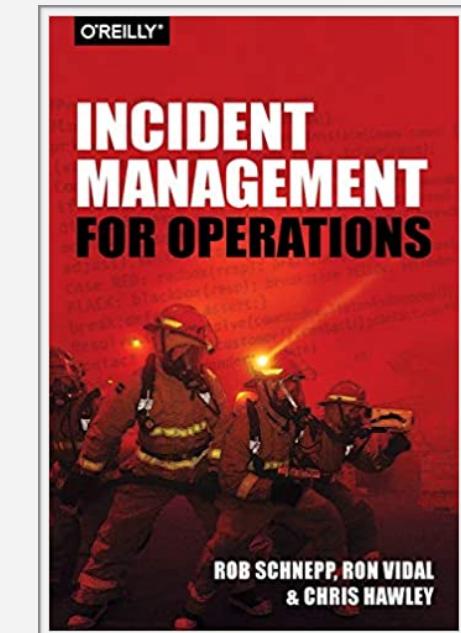


## Section 3

# インシデントマネジメントを 先人・専門家に学ぶ

## モデル・プラクティスを導入する

インシデントの際に「よい」行動をとれるよう、先人・専門家からモデル・プラクティスを学び取りましょう



????

先に紹介した sssbotは、これらのモデルやプラクティスを大いに参考にしています  
自動化に利用せずとも、同僚との学習題材としても頻繁に引用・参照しています

## 『Incident Command for IT』 by Brent Chapman (\*)

消防隊から学んだ知見を IT インシデントに応用すること勧めているプラクティス集

- インシデントコマンドシステムの導入
- 平常時と緊急時のオペレーションを分けて捉える
- “専用のチャンネルを作る”
- タイムラインモデル
- “訓練、訓練、訓練、さらに、訓練”
- … etc

すぐにも活用できる tips から マネジメントや組織文化にも言及する充実した内容です  
97ページもあり要約して紹介しきれないのでいくつかピックアップしてみます

### Incident Command for IT: What We've Learned from the Fire Department

USENIX SREcon18  
27 March 2018

PDF of these slides: <https://goo.gl/5C2M2d>

Brent Chapman

Brent@GreatCircle.com  
@brent\_chapman

# Great Circle

Copyright © 2018, Great Circle Associates, Inc. All Rights Reserved.

## "専用のチャンネルをつくる"

インシデント対応専用のチャンネルを作ることを説いています

- 平時とは別のインシデント専用チャンネルをつくる
- チャンネル名を説明的にする
- 状況やステータスドキュメントへのリンクを貼る

### 解決するアンチパターン

- アラートが溢れるチャンネルで対応を進めてしまう
- どのチャンネルで対応してるので分からぬことがある
- 対応の記録が散逸してしまうことがある

自動化とも相性がいいプラクティスです



### Tip: use a dedicated channel

- Create a channel just for this incident
  - Don't use your team's normal "chat" channel
  - Channel name should reflect incident name
  - Channel description should include one-sentence synopsis, and link to status doc
- TL and IC control the channel



## "タイムライン"のモデルを導入する

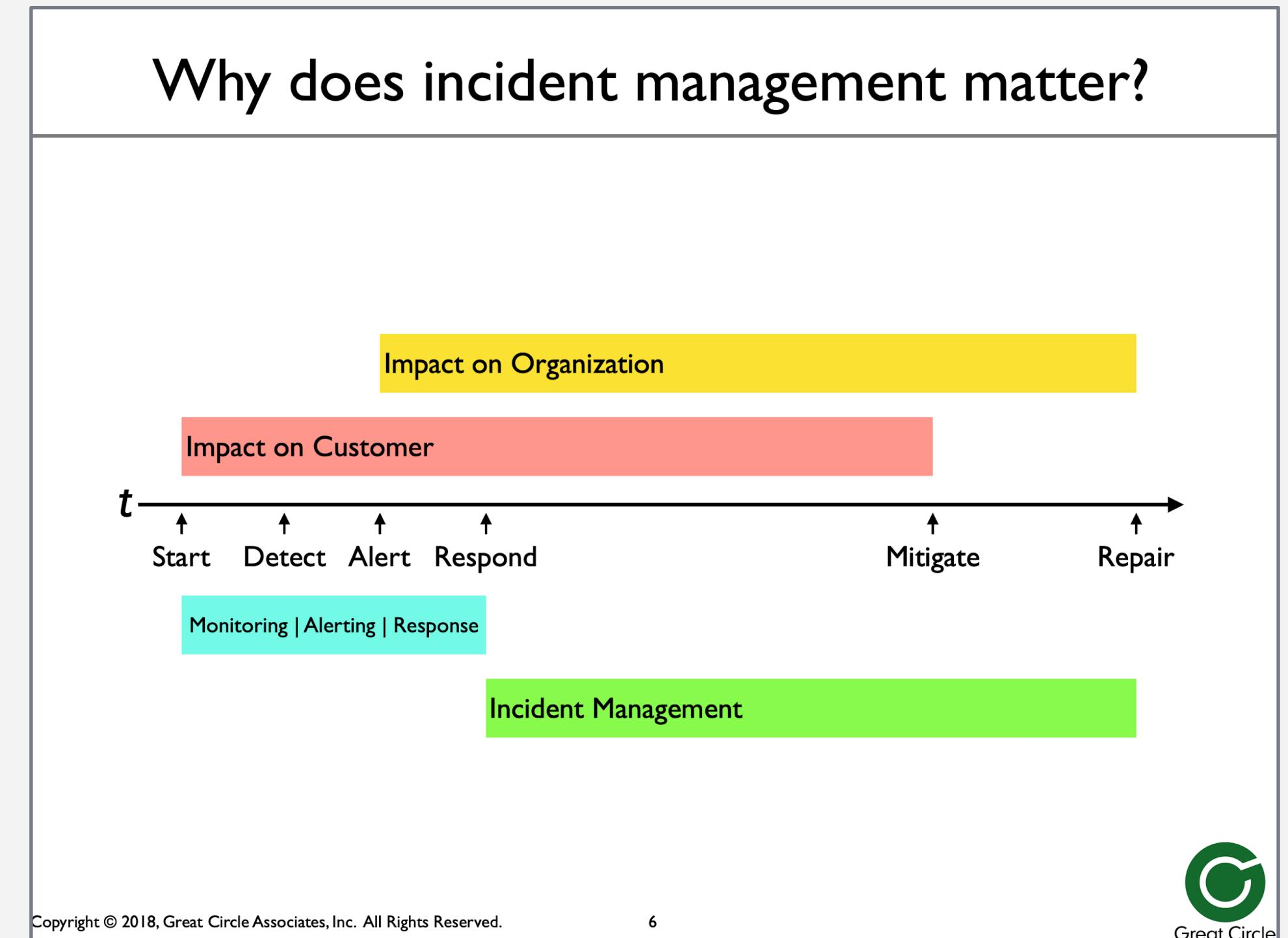
先の PDF に掲載されているタイムラインのモデルです。  
インシデントのプロセスを構造化するのに役立ちます。

- 人間のメンタルモデルにする
- 自動化の設計/実装のモデルにする

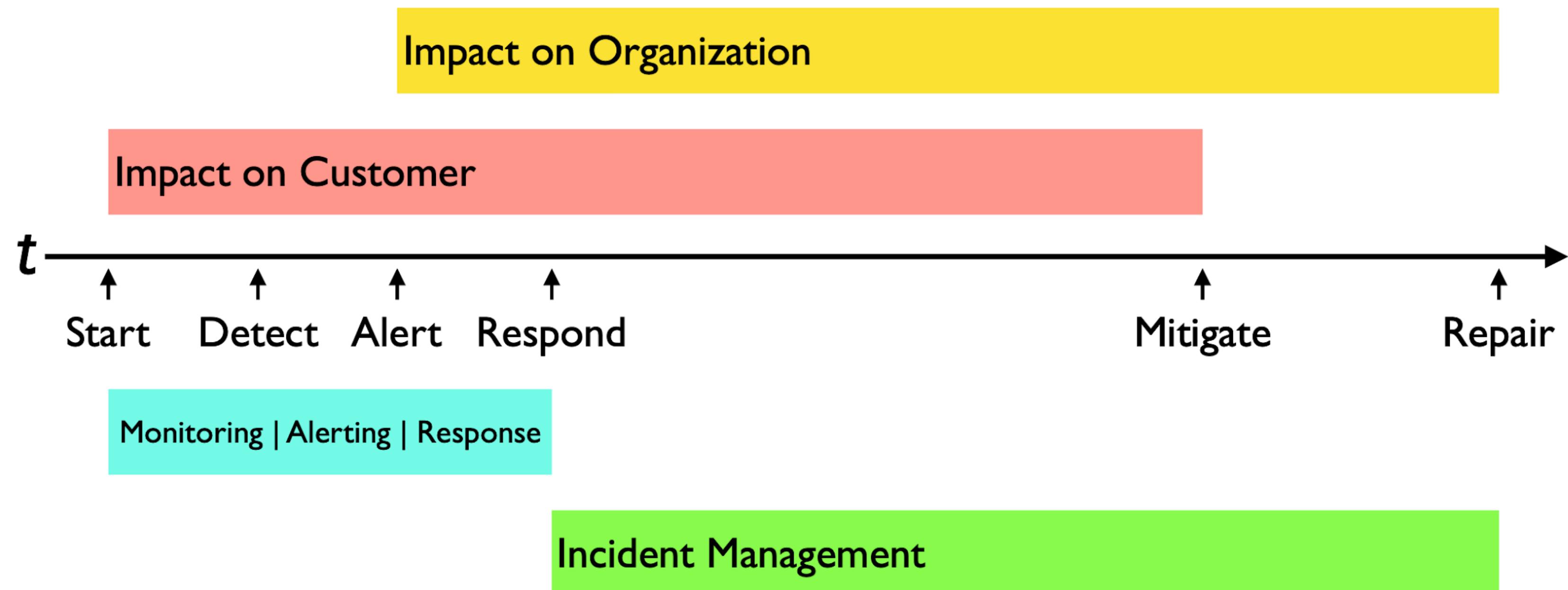
### 利用の例

- 組織内で用語・概念の統一
  - 例 「Alert ~ Respond の時間がボトルネックだから改善しよう」
- 自動化につかうとユビキタス言語

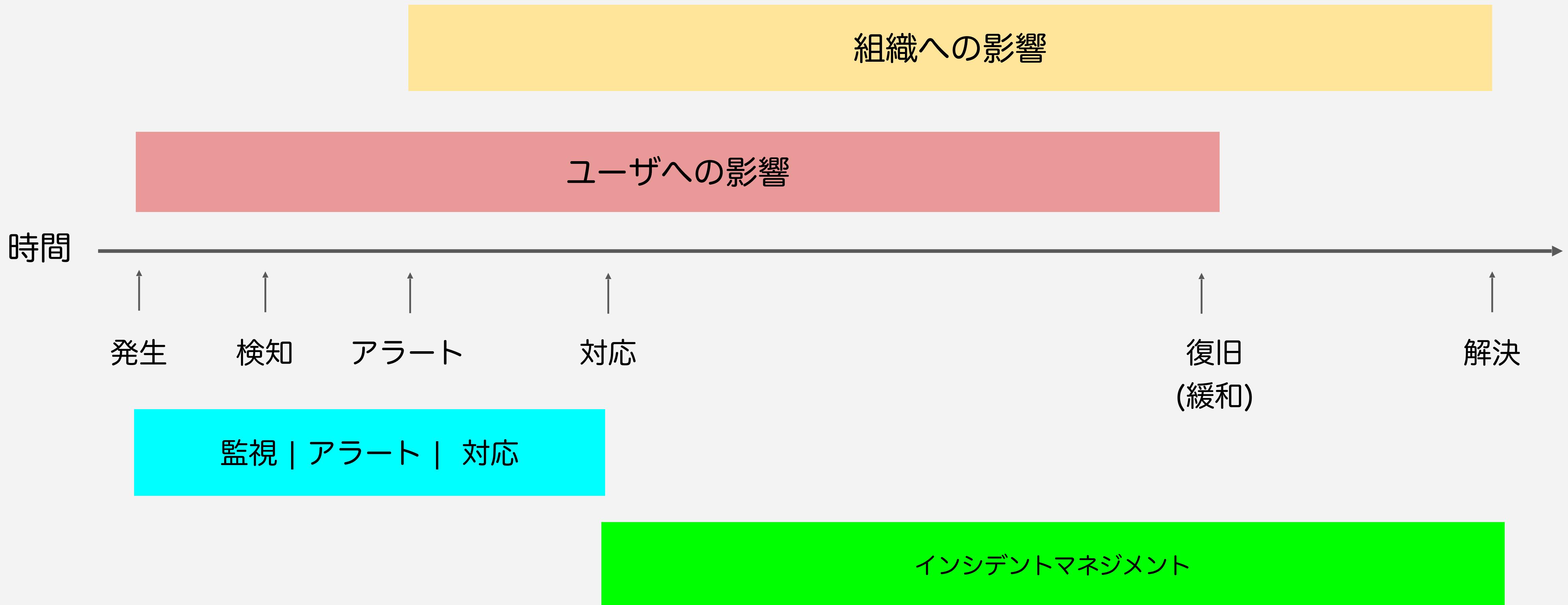
Incident Command for IT:  
What We've Learned from  
the Fire Department  
USENIX SREcon18  
27 March 2018  
PDF of these slides: <https://goo.gl/5C2M2d>  
Brent Chapman  
[Brent@GreatCircle.com](mailto:Brent@GreatCircle.com)  
[@brent\\_chapman](https://twitter.com/brent_chapman)  
Great Circle



# Why does incident management matter?

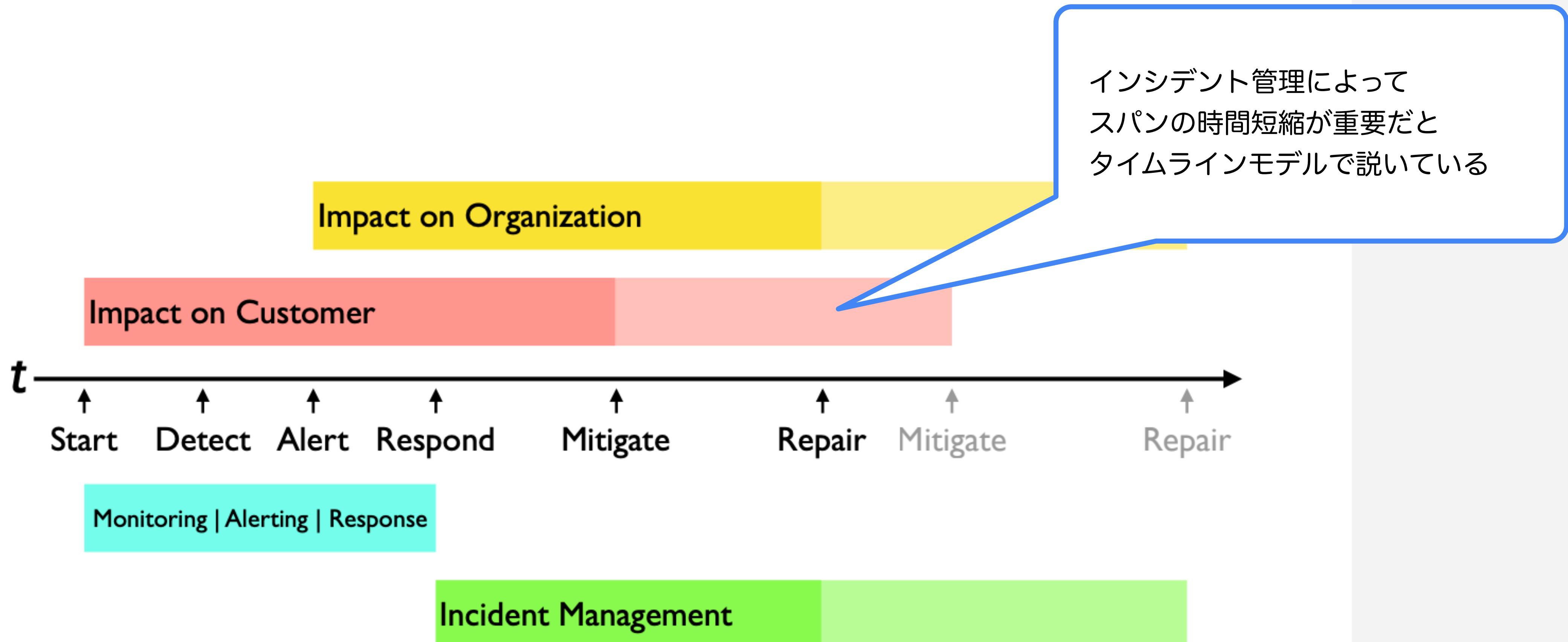


# インシデント対応のタイムライン：ペパボ翻訳版

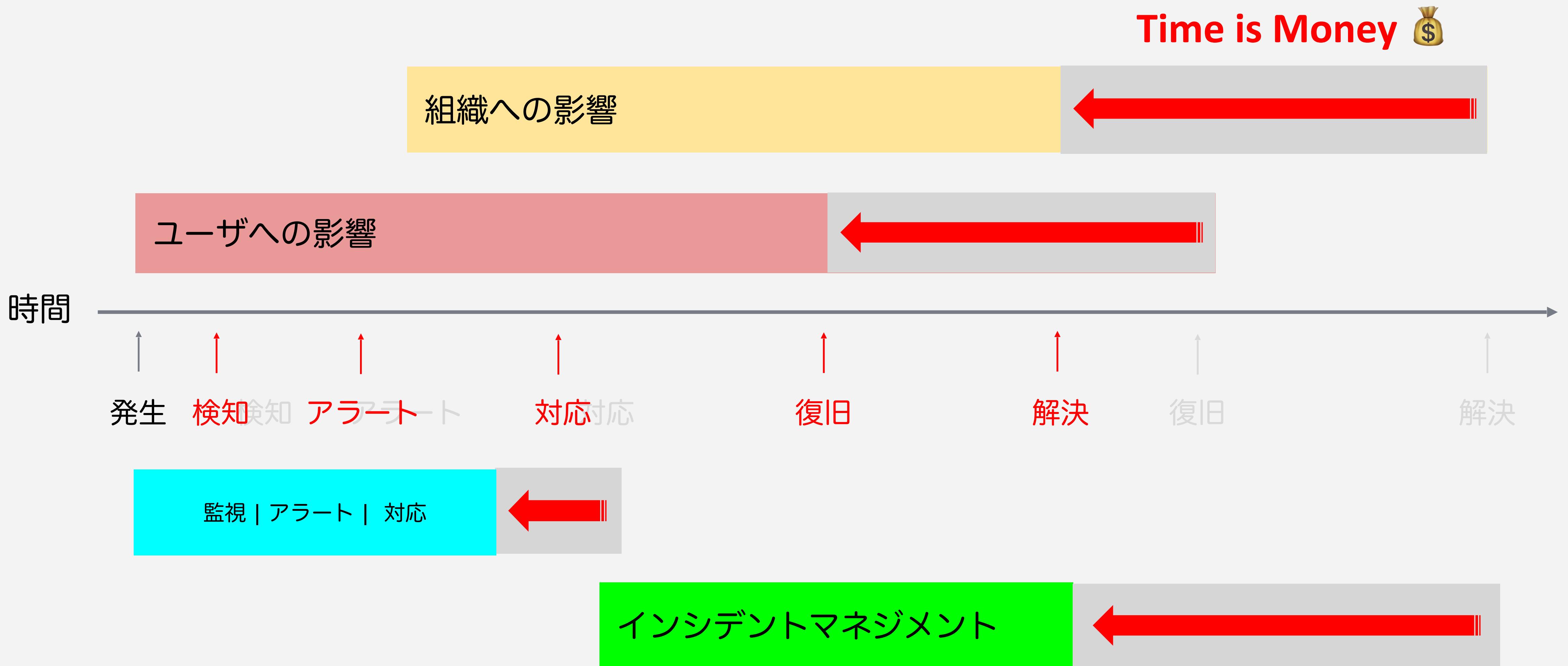


⚠️ ペパボでは可用性・完全性・機密性を回復した時に「復旧」と呼ぶ人が多いので緩和の代わりに用いている

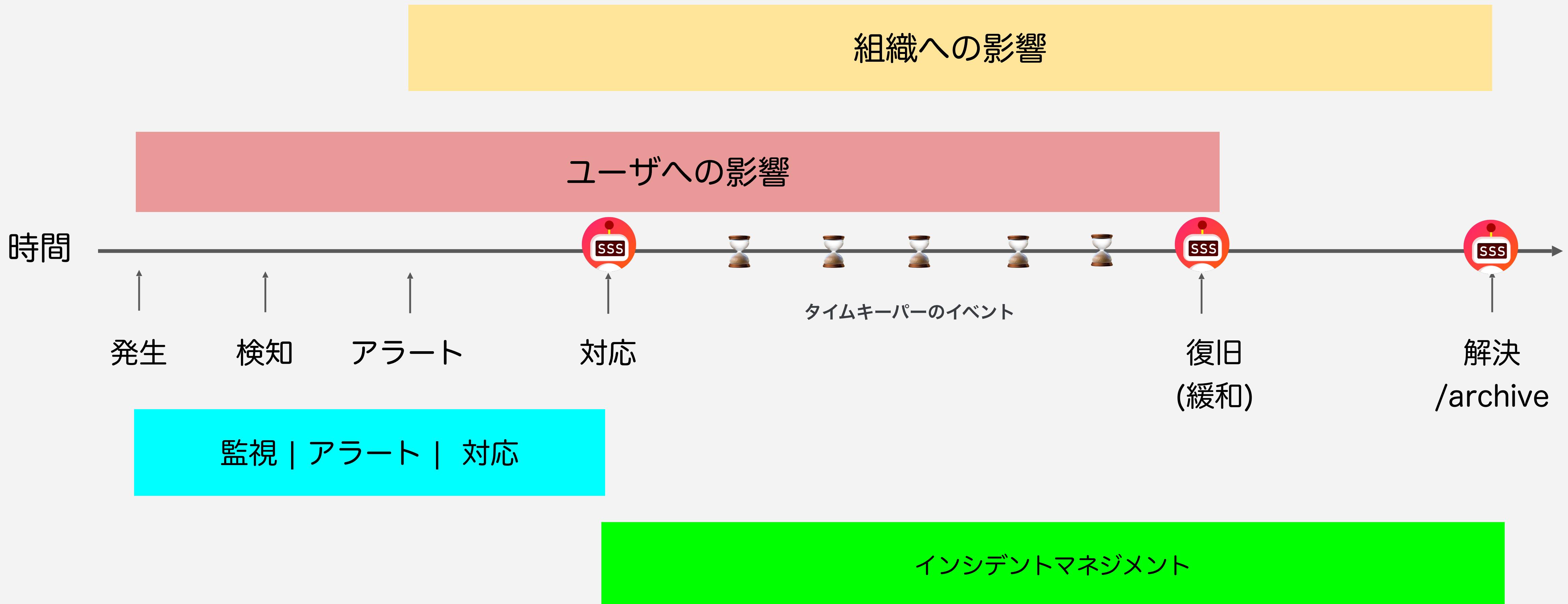
# Why does incident management matter?



# インシデント対応のタイムライン

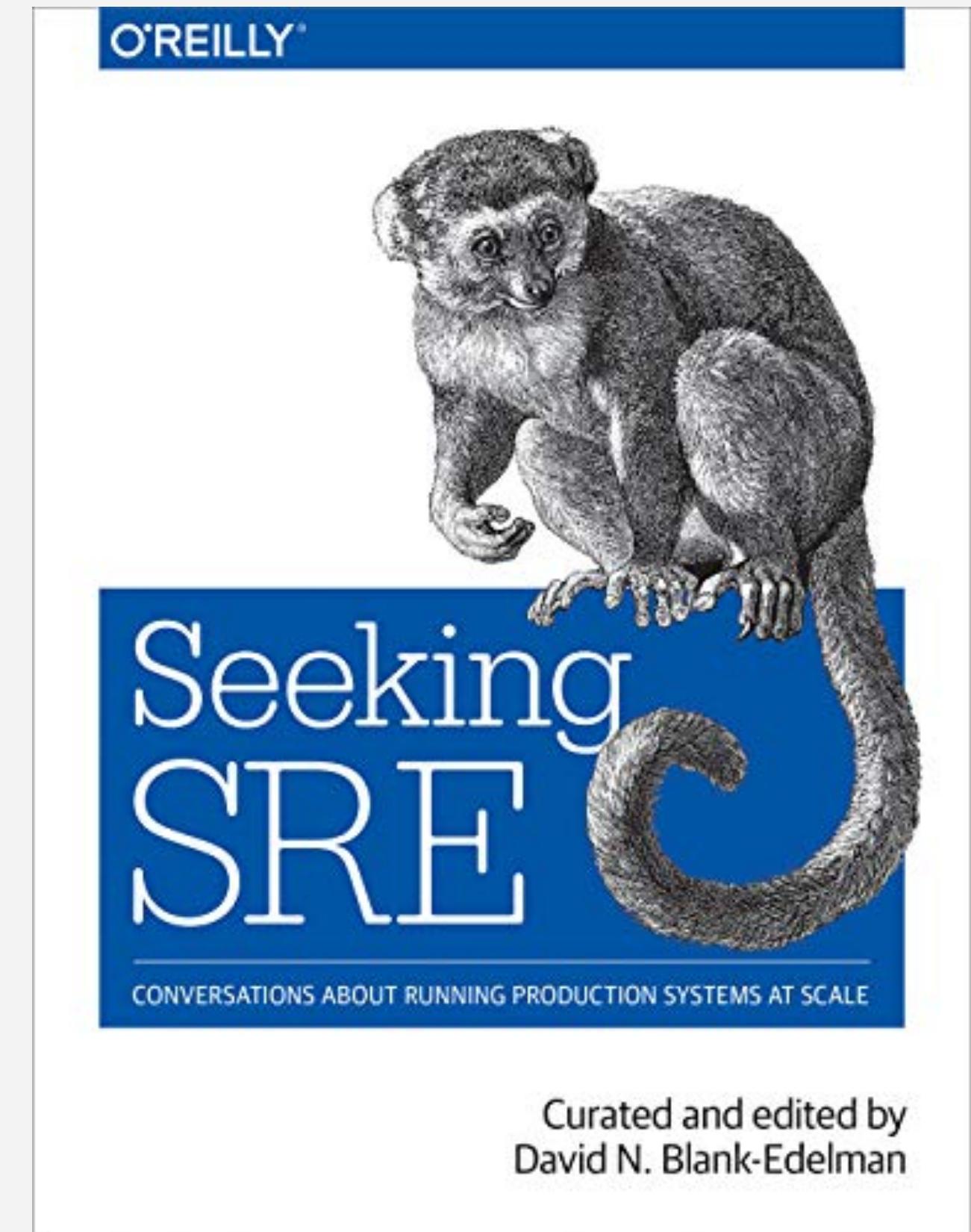
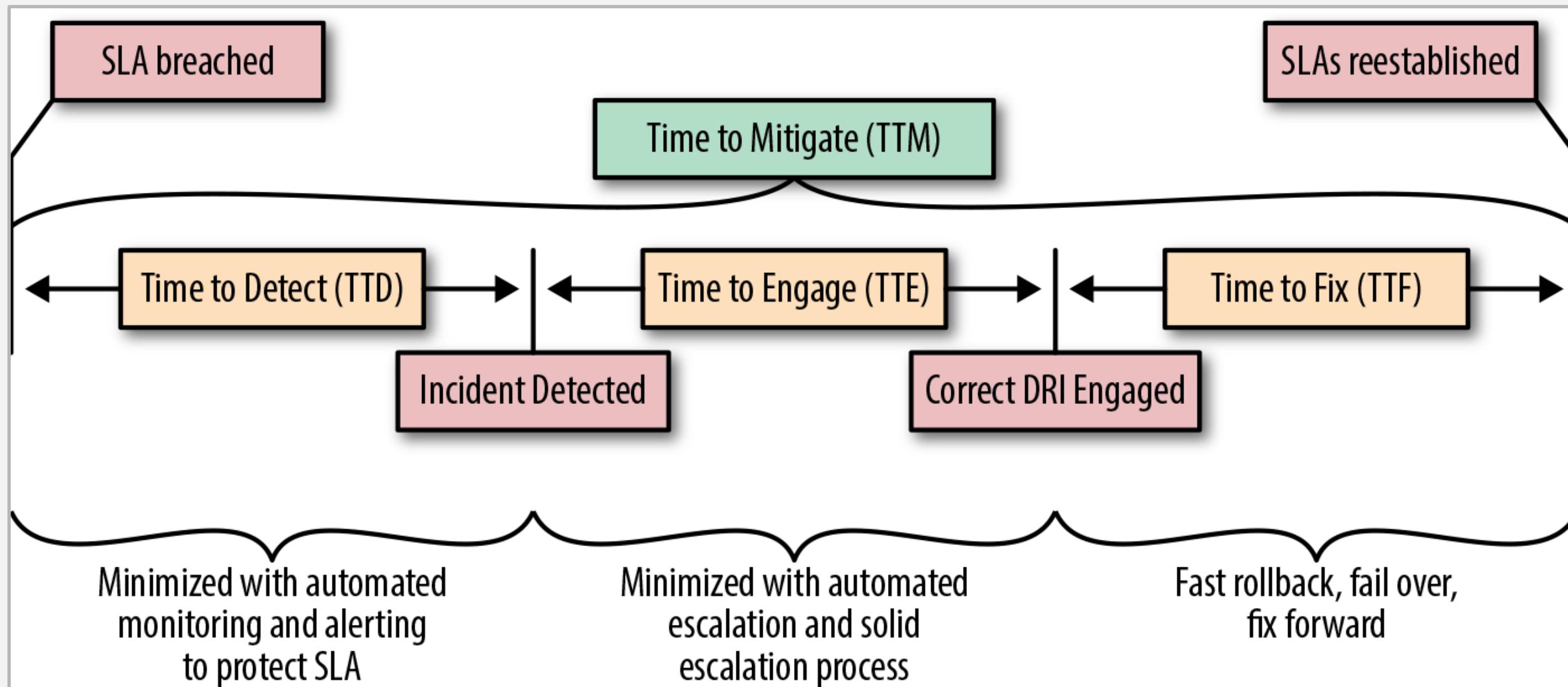


# sssbot はイベントドリブンアプリケーション



## Seeking SRE のタイムラインモデル

タイムラインのモデルは『Seeking SRE』にも載っています



先のモデルと用語の違いはありますが、フェーズの区切り方はよく似ています。

組織でのやりかたや自動化によくマッチするモデルを選択するとよいでしょう

## "危機を宣言する"

インシデントであることを明確に宣言するというプラクティスです

- 危機を宣言し、対応の優先度を上げる
- 平時と緊急時を明確に分ける
- 危機が過ぎたことも宣言し、優先度を下げる

解決する症状

- 同僚がインシデントの発生・復旧を認知していない
- 復旧したのかどうか分からず対応の優先度が曖昧になる

bot の通知で実装できるプラクティスです



### Key: Declare an Emergency

- This is not how we operate, day-to-day
- This is a special set of rules, for emergencies
- **Declare an emergency, to make it clear that you're operating under these special rules**
- Goal is to return to normal operations as quickly as possible
- Must also declare when emergency is over

Copyright © 2018, Great Circle Associates, Inc. All Rights Reserved.

23



## PagerDuty Incident Response

PagerDuty 社のインシデントレスポンスのドキュメントです

対応全般の話、トレーニング、ポストモーテム、プラクティスやアンチパターンなど

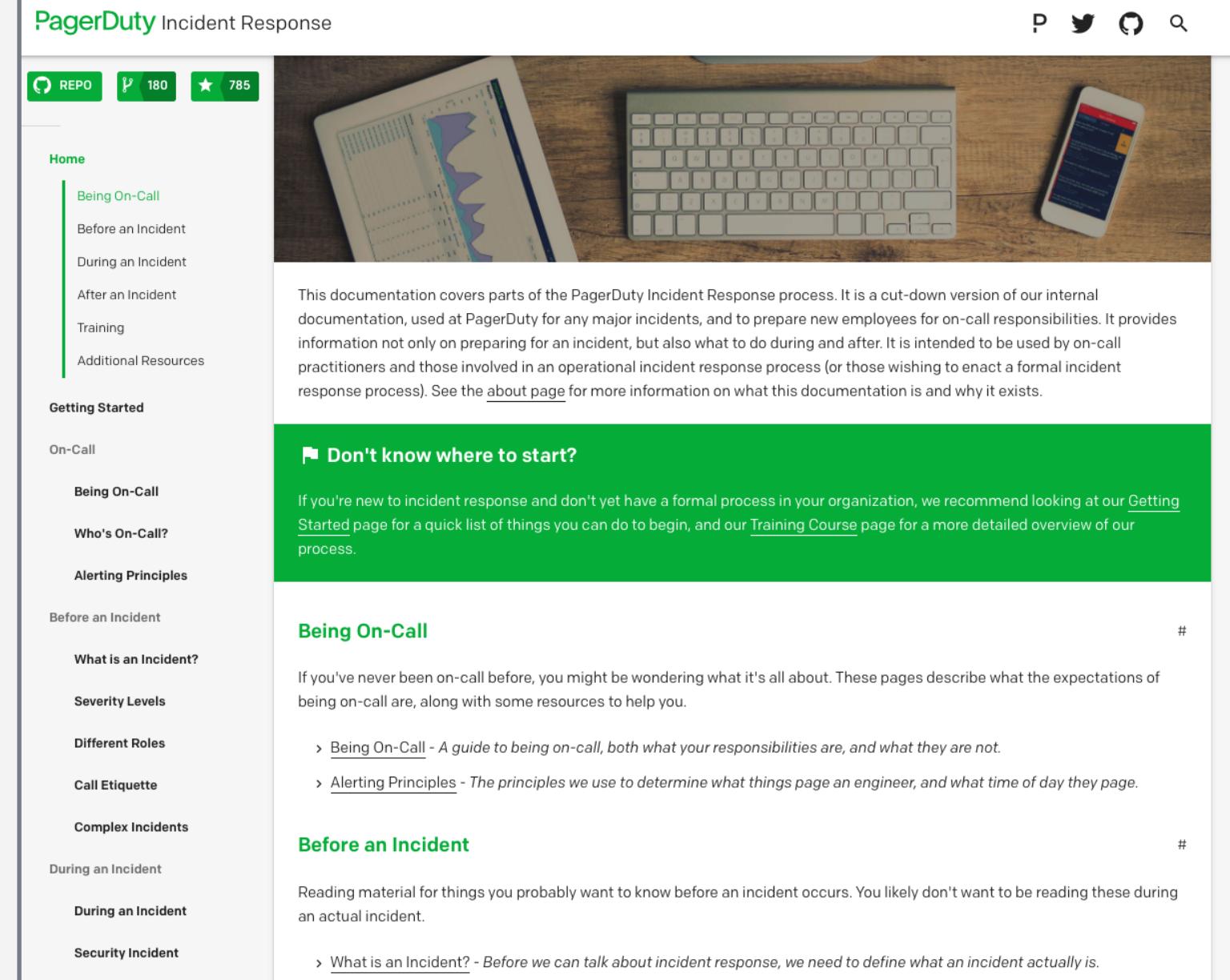
PagerDuty 社で培われたプラクティスが盛り沢山のドキュメントです。

本ドキュメントを参考にしている記事は、各所で見かけます。

slack コマンドで対応チームを呼び出すやり方も書いてあります

有志の方が邦訳を出してくださったので チームでの読み合わせにもよいテキストですね

[PagerDuty Incident Responseの邦訳版を公開しました - Folioscope](#)



The screenshot shows the homepage of the PagerDuty Incident Response documentation. At the top, there's a header with the title "PagerDuty Incident Response" and social sharing icons. Below the header is a navigation bar with links like "Home", "Getting Started", "On-Call", and "Before an Incident". The main content area features a large image of a keyboard and a smartphone on a desk, followed by a descriptive text block about the documentation's purpose. A green sidebar on the right contains sections for "Don't know where to start?", "Being On-Call", and "Before an Incident", each with a brief description and a link to more information.

## SRE サイトリライアビリティエンジニアリング

改めて紹介するまでもないですが、みんなで読んでおきたい書籍ですね。

本書を呼んでインシデント対応を見直した、ふりかえりや postmortem を始めた  
という組織が多いのではないでしょうか。

ペパボでは 2017年12月ごろから postmortem の取り組みを始めました。

本書を参考にしてpostmortem のテンプレートを作り、markdown 化しています。

後に、sssbot で自動で postmortem (の下書き) を作れるようになりました

O'REILLY®  
オライリー・ジャパン



SRE  
サイトリライアビリティ  
エンジニアリング

Googleの信頼性を支えるエンジニアリングチーム

Betsy Beyer, Chris Jones 编  
Jennifer Petoff, Niall Richard Murphy  
澤田 武男、関根 達夫、細川 一茂、矢吹 大輔 監訳  
Sky株式会社 玉川 雅司 訳

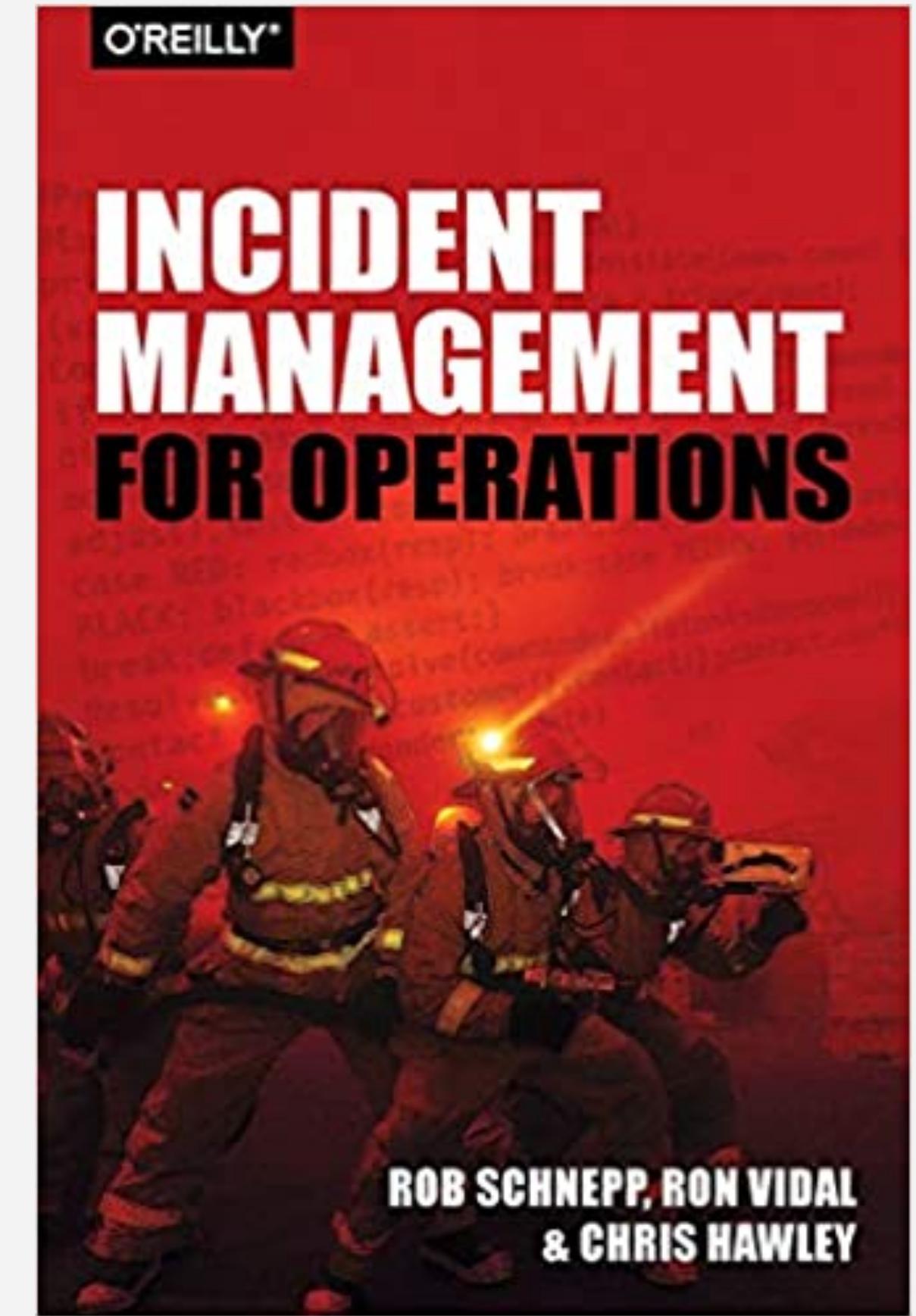
## Incident Management For Operations

元消防士が災害の現場で培った経験・方法論を ITインシデントマネジメントへ導入を説く書籍

- 組織のインシデント対応をアセスメントする質問集
- Peacetime / Wartime の組織構造
- Incident Command System の応用
- インシデント対応をスケールさせる方法
- Serverity をインシデントの指標とする
- … etc

ssbot では Serverity = 「温度感」として UI/UX に取り込みました。

人命救助に携わっている方なので「postmortem」でなく「After Action Review」と呼ぶことを  
勧めるくだりでもハッと気が付きがありました。



## Peacetime の組織 / Wartime の組織

Peacetime = 平時ビジネスを進める組織と

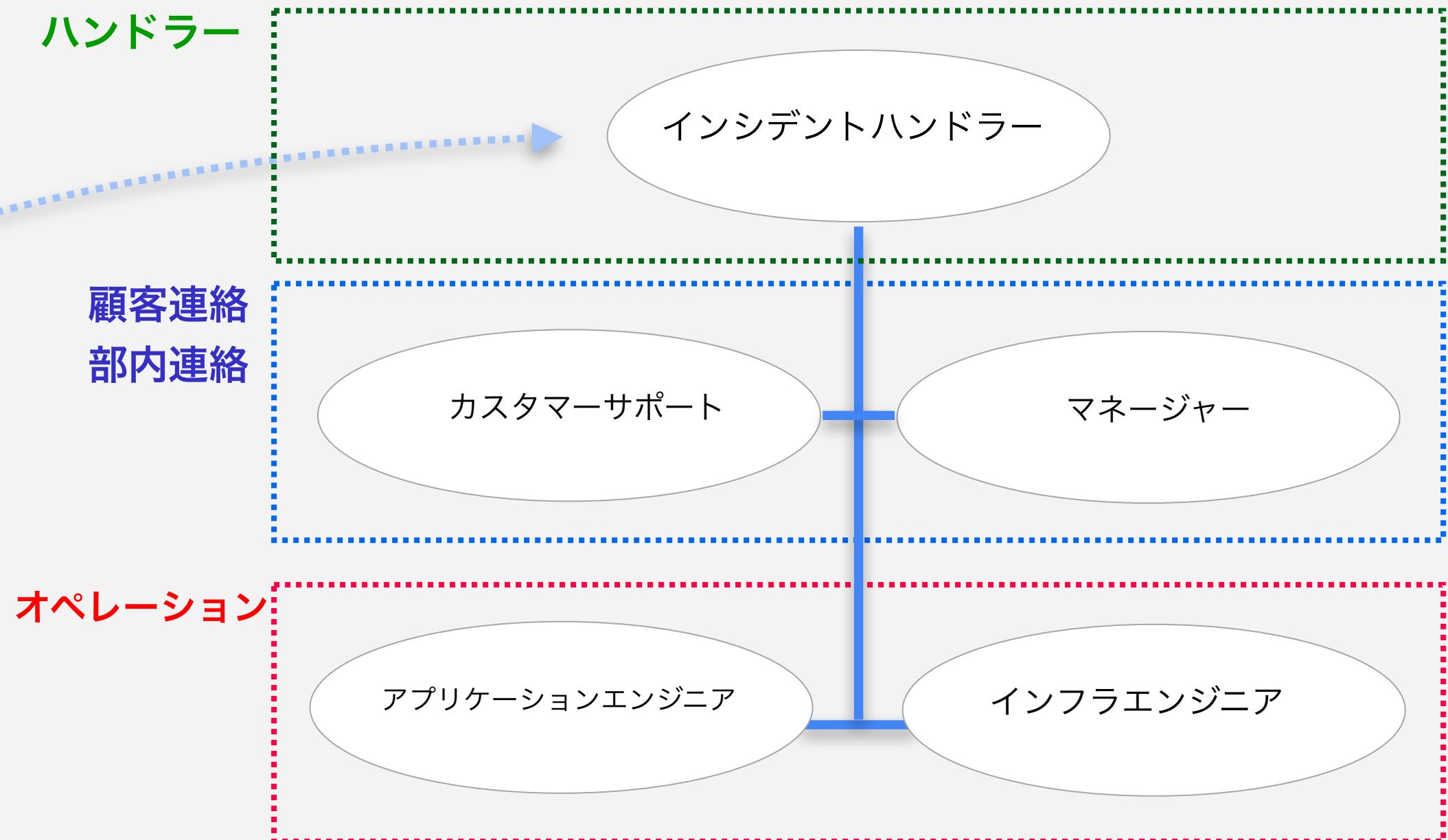
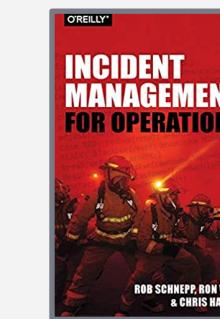
Wartime = インシデント対応時の組織の考え方方が書いてあります。



Peacetime の組織モデル

経営層・ミドル・現場 / 職位・職能での階層 (\*)

静的な組織構造 / 中・長期的な事業目標を優先する意思決定



Wartime の組織モデル

インシデントハンドラーを筆頭にロールで構成

テンポラリな組織構造 / 復旧 = 超短期を最優先する意思決定

1) <https://www.pagerduty.com/blog/peacetime-wartime-devops/> にも同様の話が載っている。日付からすると こちらが原典かもしれない。

2) Peacetime と Wartime の話を論じるために出した「会社」のモデルです

## システム障害対応の教科書

本スライドにも書かれているモデルやプラティクスまでも含めて網羅的に「障害対応」を体系化した書籍です。日本語で読めるのも嬉しいですね

GMO ペパボでは本書の「インシデントレベルのアセスメント」を取り込みました。

postmortem 後の対応、インシデント対応訓練にアセスメント評価を行います。  
アセスメントをもとに対応レベルの評価や、訓練計画の立案、ふりかえりの題材とします



## 🔥 誰も... 消防車を呼んでいないのである!!!

漫画『アフロ田中』のツイートを読んで**傍観者効果**を理解しましょう  
火事が起きているのに誰も消防車を呼んでいないという一コマです

- 傍観者効果を避けよう
  - 「障害? 誰かが対応してるだろう」
  - 「アラート? きっと誰か気がつくだろう」
  - ...

あなたが「消防車 = 問題を解決できる人」になろう/呼びましょう!!

新しく入社された同僚向けのオンボーディングで漫画を読んでもらって  
(笑えないけど、笑いながら) インシデント対応のマインドを啓蒙します

のりつけ雅春。「アフロ田中」描いてます。 @zen... · 2019年2月18日 ...  
返信先: @zenbutukawareteさん  
消防車が来ない話 ⑥





## Section 4

# インシデントに 立ち向かう組織づくり

## 人機一体 = 組織文化 + 技術

sssbot を通じて達成したいことは、自動化したインシデントマネジメント支援を用いて、いちはやくサービスの危機を乗り越えることです

しかしながら、自動化できることにも限界があるのが実情です。

組織内で継続的な学習・訓練・啓蒙を続けていくことにより、逆境に強い組織をつくりあげレジリエンスを獲得できると考えています



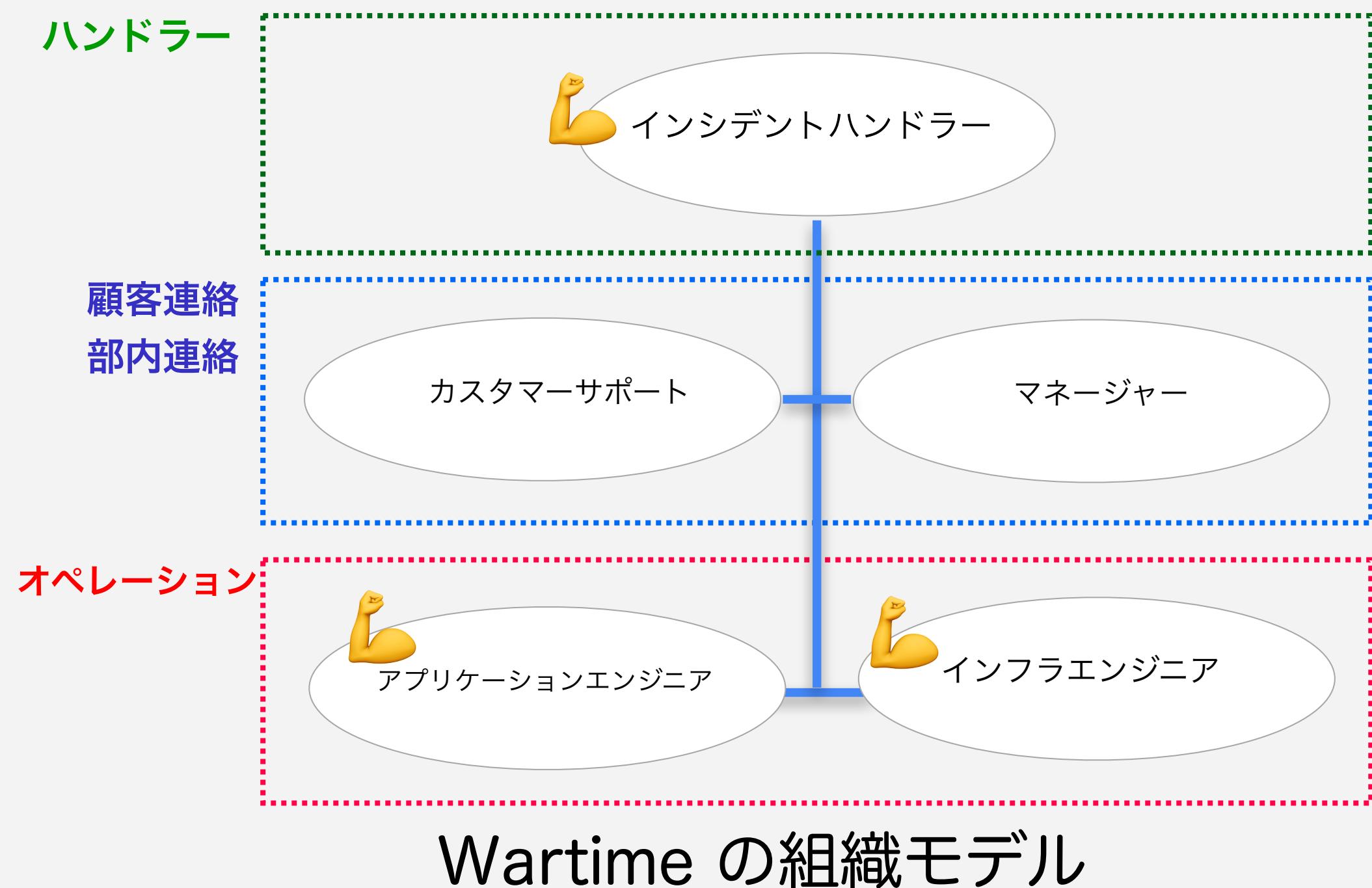
## 持論: インシデント対応と技術者のリーダーシップ

bot が支援できるのはプロセス支援に留まります。

インシデントの技術的問題まで「自動で解決する」のは困難です (\*)

- 問題が解決されなければ、インシデントから復旧できない
  - 問題解決は技術者に委ねられる (\*)
  - 技術者はインシデント対応でリーダーシップ(\*) をとる責務を負う

Wartime の組織モデルでは、インシデントハンドラーに加えて  
技術者もリーダーシップを発揮することが重要だと考えています。



1) 自動で解決できる問題は システムにあらかじめ設計・実装して対応する

2) セキュリティインシデント(の疑いがある事象) では、カスタマーサポート職がユーザ様とのやり取りを進めてリードする事例もあります

3) リーダーシップと言ってもカリスマ性・英雄像を求めるといった話ではない。「能動的に問題を解決しようとアクションを起こして他の人をひっぱる」というニュアンスです

## 平時からの学習・訓練・啓蒙

リーダーシップやレジリエンスのマインドセット = 組織文化は一朝一夕で作られません

DevSecOps Cycle と情報セキュリティマネジメントも併せてインシデントに備えます

- インシデントハンドラーの勉強会
- インシデント対応訓練
- セキュリティ対策室のオンボーディング ( sssbot チュートリアル )
- ヒヤリハット共有会
- Pepabo Tech Friday (社内技術MTG) での宣伝
- 脅威モデリング



SSS Tuesday SSS SSS

セキュリティ対策室の紹介とオンボーディング

ペパボテックブログ  
Technology, Engineering, Creative, and Human-Centered Design

2019-03-04 ロリポップに対してインシデント訓練を行いました

engineering | 研修 | ホスティング事業部 | ロリポップ

[サイト](#) [いいね!](#) [シェア](#) [壁紙マーク](#) [Pocket](#)

こんにちは、@kunitaです。

2月の後半に、GMOペパボ株式会社(以下、ペパボ)のホスティング事業部で開発・運営しているレンタルサーバーのサービス「ロリポップ」に対して、インシデントの対応訓練を行いました。どんな感じで実施したかをご紹介したいと思います。

### 訓練の目的

ペパボでは2018年の1年間をかけて、全社でセキュリティ強化の対策を行ってきました。それに伴って導入した仕組みやフローをインシデント発生時に実際に正しく運用できるのか？現在のフローをより良くしていくためにどうしたらいいのか？というのを日頃から確認していくのが、インシデント対応訓練の目的です。

### 訓練のながれ

### 最新記事

2021-10-06 初見連想でより意味を伝えるデザイン～minneの事例～

2021-10-04 新卒エンジニア研修2021を受けました（前半）

2021-09-22 DBモーティングとRSpecのワークショップを行いました

2021-09-21 GMOペパボのエンジニア研修2021の資料を公開します

2021-09-17 minneにおけるスクラム開発改善例のご紹介

2021-09-16 minneのフロントエンドをNext.jsで刷新してわかったこと

2021-09-10 GitHub Enterprise ServerのGitHub Packagesで社内用gemをホストする

タグ

## インシデント対応訓練

インシデント対応訓練を定期的に実施しています。サービスごとにシナリオを考えて実施します。

sssbot も本番同様に用います。インシデントハンドラーを練習する機会にもなります。

#colorme-20210930i-訓練 #886

**Closed** 2 comments

sssbot 20 days ago  
sssbot が自動でコメントしています

**インシデントの概要**

- サービス colorme
- チャンネル #colorme-20210930i-訓練

**サマリ**

怪しいコマンドがサーバで実行されている可能性

**チャンネル作成の記録**

- 2021-09-30 15:12:41 +0900

Assignees – assign yourself  
Labels  
**colorme** **セキュリティインシデント**  
Projects  
Milestone  
Linked pull requests  
Notifications  
Customize  
Unsubscribe  
0 participants

2019-03-04  
ロリポップに対してインシデント訓練を行いました

engineering 研修 ホスティング事業部 ロリポップ

こんちは、@kunitです。  
2月の後半に、GMOペパボ株式会社(以下、ペパボ)のホスティング事業部で開発・運営しているレンタルサーバーのサービス「ロリポップ」に対して、インシデントの対応訓練を行いました。どんな感じで実施したかをご紹介したいと思います。

**訓練の目的**

ペパボでは2018年の1年間をかけて、全社でセキュリティ強化の対策を行ってきました。  
それに伴って導入した仕組みやフローをインシデント発生時に実際に正しく運用できるのか? 現在のフローをより良くしていくためにどうしたらいいのか? というのを日頃から確認していくのが、インシデント対応訓練の目的です。

**訓練のながれ**

タグ

## SSS Tuesday ~セキュリティ対策室のオンボーディング

中途で入社された方向けに、セキュリティ対策室でオンボーディングを行います（隔月開催）

セキュリティ対策室のミッションの説明から始まり、sssbot のチュートリアル（実際に動かしてもらう）もります



### セキュリティと組織文化

なんでもコミュニケーションしやすい空気の情勢

- ヒューマンエラーを責めない
  - 誰かの失敗には「Nice Try！」で声をかける
- 我々は技術の専門家なのでディスカッションしよう
  - 等級・年齢・所属年数に縛られず議論しましょう
- 解決できないときは誰かに頼る
  - #dev や #sss で相談してみましょう
  - 「誰も消防車を～」
- みんなと何かよくすること
  - オンラインでもオフラインでもコミュニケーションとりましょう！

NT! :nicetry: NT!

SSS SSS

27

## トップダウンのマネジメント: マニュアルの作成

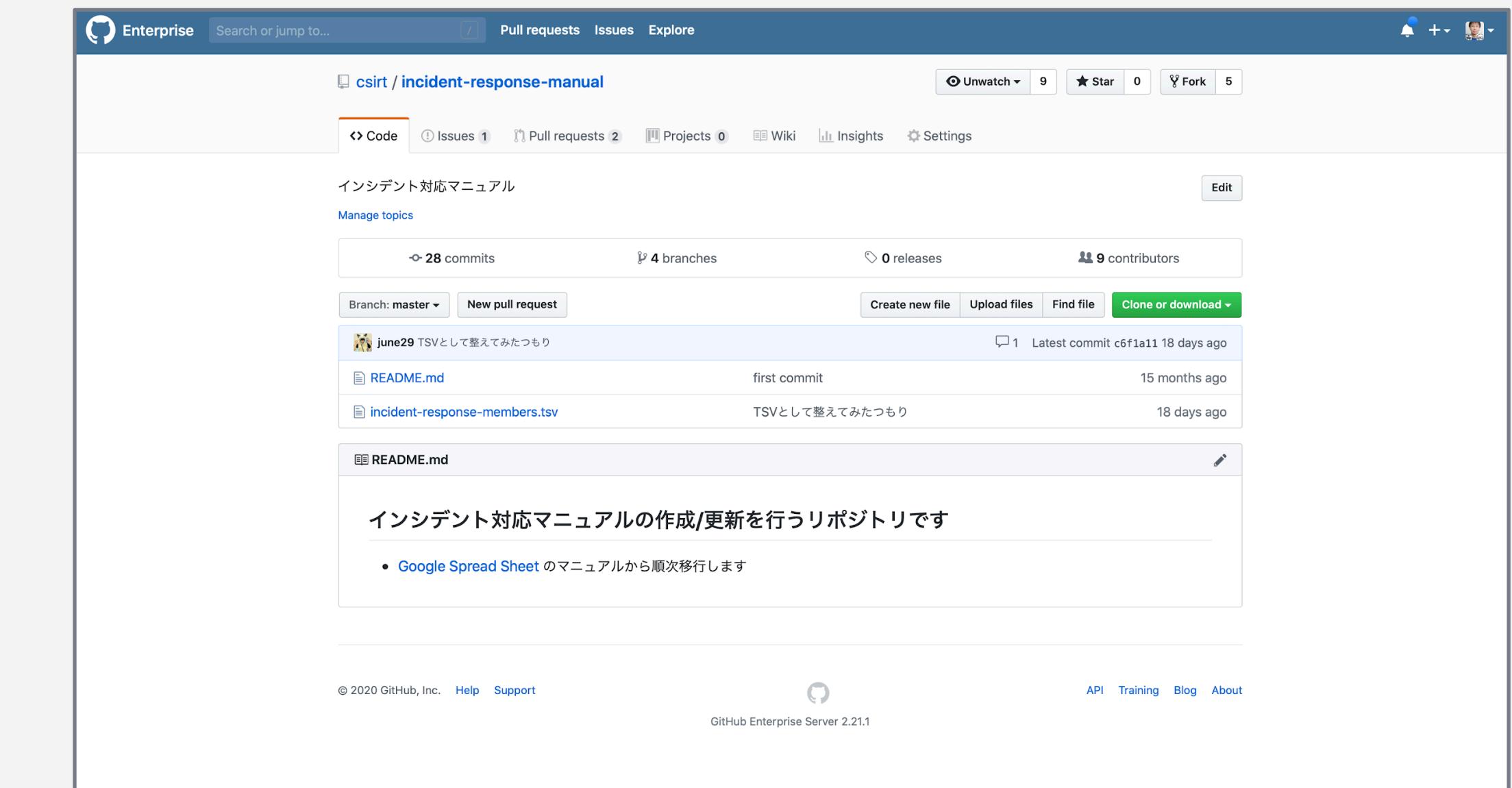
ペパボでは「インシデント対応マニュアル」(内製)が存在します。

セキュリティ対策室が情報セキュリティマネジメントの一貫で作成しています

- インシデントの定義
- インシデント発生時の対応フロー
- 初動対応チームの発足
- 組織内外への連絡フロー
- ... etc

そもそも sssbot が本マニュアルをなぞるように作成されたものでした 

自動化システムをマニュアル・規定・制度と整合性を保つようにして  
マネジメントサイドと並走した利用を目指します



## トップダウンのマネジメント: アセスメントの作成

『システム障害対応の教科書』の「インシデント対応レベルのアセスメント」をペパボ用語に書き直したアセスメントを作っています  
postmortem やインシデント対応訓練のタイミングでアセスメントの実施を薦めています

hiboma 大見出しをつける

Latest commit d7faf2d on 26 Feb History

1 contributor

150 lines (86 sloc) | 4.42 KB

Raw Blame

### インシデント対応レベルのアセスメント

#### 人の動きに関するインシデント対応レベル

##### レベル1

- インシデントハンドラー不在で場当たり的な対応
- 個々の能力がバラバラで、個人の能力に頼る
- インシデント対応時の自チームの役割が決まっていない / ルールが認識されていない
- 各個人がバラバラの情報を保持し、噛み合わないことが多い

##### レベル2

- インシデントハンドラーを中心とした組織的な対応
- 誰でも標準以上の能力を発揮する
- インシデント対応時の他部門との連携方法が決まっていない / ルールが認識されていない
- 各組織間でバラバラの情報を保持し、全体方針を決めるのに時間がかかる

##### レベル3

- インシデントハンドラーと作業担当の高度な情報共有に基づく意思決定
- 作業担当も全体状況を把握できる



## トップダウンのマネジメント：評価に組み入れる

ペパボのエンジニア職の評価制度では、専門性を評価する軸として、以下の三つの力 (\*) を掲げています

- 作り上げる力
- 先を見通す力
- 影響を広げる力

次期の改定では、例外的事態への対応（例：インシデントに対応できること）を評価要件に盛り込むことを計画しています。



執行役員 VP of Engineering/技術部長 柴田博志

### エンジニア評価のポイント(1)

エンジニアは専門職として、エンジニア組織内で評価の基準と体系を持ち、専門職上長が全体の 5/9 (シニア以上は 2/3) の評価を行う。

Grade評価者及び被評価者（エンジニア）		GMOペパボ株式会社					
エンジニア職については、要件ごとに評価者を以下のように設定します		1～3等級		4等級		5～6等級	
被評価者		専門職上長	所属上長	専門職上長	所属上長	専門職上長	所属上長
作り上げる力	専門力	1次：シニアorEL 2次：SEL	—	1次：SEL 2次：VPoE	—	1次：VPoE 2次：CTO	—
	完遂力	—	1次：MGR 2次：部長	—	—	—	—
	行動力	—	—	—	—	—	—
先を見通す力	予見力	1次：シニアorEL 2次：SEL	—	1次：SEL 2次：VPoE	—	1次：VPoE 2次：CTO	—
	課題抽出力	—	—	—	—	—	—
	改善力	—	—	—	—	—	—
影響を広げる力	影響力	—	—	1次：MGR 2次：部長	—	1次：MGR 2次：部長	—
	主体性	—	—	—	—	—	—
	影響力	—	—	—	—	—	—

CTO=技術担当取締役、VPoE=技術担当執行役員、SEL=シニアエンジニアリングリード、EL=エンジニアリングリード

58

2020年人事制度の改訂（2019年11月5日経営会議用）

1) 現行の制度については [ペパボのエンジニアの各種制度 2020 夏 - ペパボテックブログ](#) を参照してください

2) さらに 9項目に細分化された評価軸も存在する。等級によって適用が異なる

3) 従来の評価軸に例外的事態への対応が追加されて 専門性の主張軸が拡張されるのである。短絡的に「インシデント対応をしなければ評価が下がる」と捉えるものではない

## アウトプットに昇華する

技術的な難易度の高かったインシデントでは、問題を抽出・一般化してアウトプットとし、学びに昇華します

**ペパボテックブログ**  
Technology, Engineering, Creative, and Human-Centered Design

#トラブルシューティングに関する記事一覧

最新記事

- 2020-10-29 ペパボトラブルシュート伝 - 第12回 コンテナ技術の情報交換会@オンラインで「cgroup と sysfs ファイル トラブルシューティング事例から cgroup を深追いする」を発表しました
- 2020-06-26 ペパボトラブルシュート伝 - TCP: out of memory - consider tuning tcp\_mem の dmseg から辿る 詳解 Linux net.ipv4.tcp\_mem
- 2020-06-11 ペパボトラブルシュート伝 - node プロセスの general protection fault を追う - abort(3) の意外な実装

タグ

- デザイン (67)
- イベントレポート (42)
- 勉強会 (39)
- 研修 (33)
- pepabo (27)
- ECプログラマー (22)
- カラーミーショップ (21)
- CTO (19)
- minne (19)
- プロダクトマネジメント (18)
- ProductManagement (17)

1 / 1 ページ

### 4. Linux カーネルの TCP メモリ管理を俯瞰する

TCP oom を理解するにあたって、Linux カーネルのメモリ管理に踏み込んでいきましょう。次の図は、Linux カーネルの TCP メモリ管理をモデル化したものです。

Linux カーネルは、TCP ソケットでデータを送信する・データを受信する際にデータをバッファリングします。バッファはカーネル内に確保します。バッファ用のメモリは、メモリ管理サブシステムから割り当てます。



トラブルシューティングをペパボテックブログの題材に書くのは、今のところ私個人での取り組みにとどまります  
postmortem を延長したとしてアクションとして広まってくれるといいなと思っています

## RE: SRE の書籍

改めて SRE 本を見直して「人間」にフォーカスした章を再発見しましょう

- 12章 効果的なトラブルシューティング
- 13章 緊急対応
- 14章 インシデント管理
- 15章 ポストモーテムの文化：失敗からの学び
- 16章 サービス障害の追跡

Google のような高い技術をもつエンジニアリングチームが  
インシデントマネジメントを行っていることを反芻したい

編  
澤田 武男、関根 達夫、細川 一茂、矢吹 大輔  
監訳  
Sky 株式会社 玉川 竜司

O'REILLY®  
オライリー・ジャパン



Betsy Beyer, Chris Jones 編  
Jennifer Petoff, Niall Richard Murphy  
澤田 武男、関根 達夫、細川 一茂、矢吹 大輔 監訳  
Sky 株式会社 玉川 竜司 訳

Section 5

# 展望と課題



## 今後の展望や課題

sssbott を作ってから 2年半ほど経過していますが、やりたいこと / 課題は様々あがっています

- SRE の取り組みを支援したい
  - SLA/SLI/SLO の計測に応用する
  - インテグレーションを充実させたい
    - PagerDuty, DataDog 等の SaaS と連携
    - システムのメトリクス (Prometheus) と連携
  - Slack UI をもっと活用したい
    - 例) チェックリスト UI でプロセスの進行を把握できるとか
  - ... etc

(後述する) SRE / Incident Response Platform を提供する SaaS も出ており

内製でどこまで開発するかも悩ましいところです



## 海外の SRE / Incident Response の SaaS

海外では SRE / インシデント対応にフォーカスした SaaS も登場している

The screenshot shows the Rootly.io homepage with a dark blue header featuring the logo and navigation links: Home, Product, Resources, Pricing, My Account, and a green 'BOOK A DEMO' button. Below the header, there's a large callout box with the text 'Manage incidents directly from Slack'. It includes a subtext: 'Focus on what you do best, putting out 🔥 Get started in 5min or less' and a 'Sign in with Slack' button. In the center, there's a screenshot of a Slack interface titled '#incident-20210412-customers-unable-to-place-order...'. The Slack interface shows integrations with Zoom room and Jira issue links. A modal window titled 'Select responders' is open, showing a dropdown menu with 'Default' selected under 'Escalation Policies'. Other options include 'Users' with entries 'JJ Tang' and 'Quentin Rousseau'. A 'View on rootly.io' button is visible at the bottom of the dropdown. At the bottom of the screenshot, there's a message input field: 'Send a message to #incident-20210412-customers-una...'.

<https://rootly.io/>

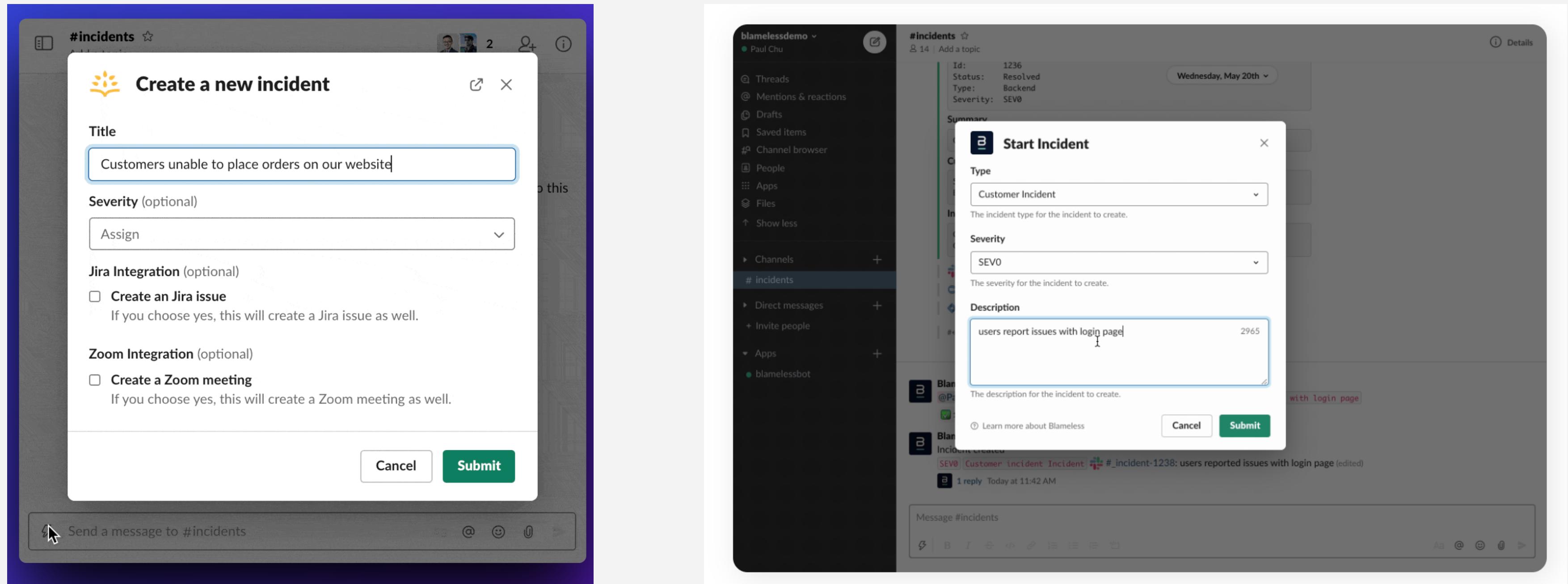
The screenshot shows the Blameless website with a dark blue header featuring the logo and navigation links: Product, Blog, Customers, Resources, Company, Docs, and a green 'Schedule Demo' button. Below the header, there's a main heading 'Blameless: The End-to-End SRE Platform' and a sub-section 'Advance Teams to a Culture of Resilience'. A descriptive text states: 'Blameless is the industry's first end-to-end SRE platform, empowering teams to optimize the reliability of their systems without sacrificing innovation velocity.' Below this, there are two buttons: 'Schedule Demo' and 'Try It Now'. To the right, there's a 3D illustration of a person standing on a glowing blue platform surrounded by multiple screens and data visualizations. At the bottom, there's a section titled 'Trusted by Leading Teams' with logos for MAMBA, ZAPIER, PROCORE, and CITRIX.

<https://www.blameless.com/>

この他にもサービスがありましたらフィードバックをいただけます

## 海外の SRE / Incident Response の SaaS

Slack インテグレーションを備え、インシデント対応のプロセスを自動で支援する UI/UX も提供する



<https://rootly.io/>

<https://www.blameless.com/>

1) デモを眺めただけで、実際に詳細は把握しておりません。利用している方がいらっしゃたらフィードバックをいただけると幸いです

2) Severity を入れるのはどこも採用しているプラクティスになっている。真似したわけではないので一応



終わり



エンジニアリングからマネジメントまでテーマを広げたGMOペパボの事例でした  
IT企業でのインシデント対応にまつわる話をさらにオープンにディスカッションしていきましょう



# Appendix

録画の際に掲載を見送ったページです

## sssbot 開発から普及まで

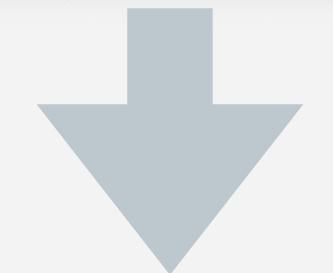
sssbot はチャンネルを作るだけの簡素な CLI ツールから徐々に slack app に作り込んでいった

- 社内で前乗りで使ってくれる人にフィードバックをもらう
  - ペパボの在籍が長く 誰に勘所が利いたという面もあるかと思う
  - ツールの進化とともに、従来のやり方を徐々に変えていくように努めた

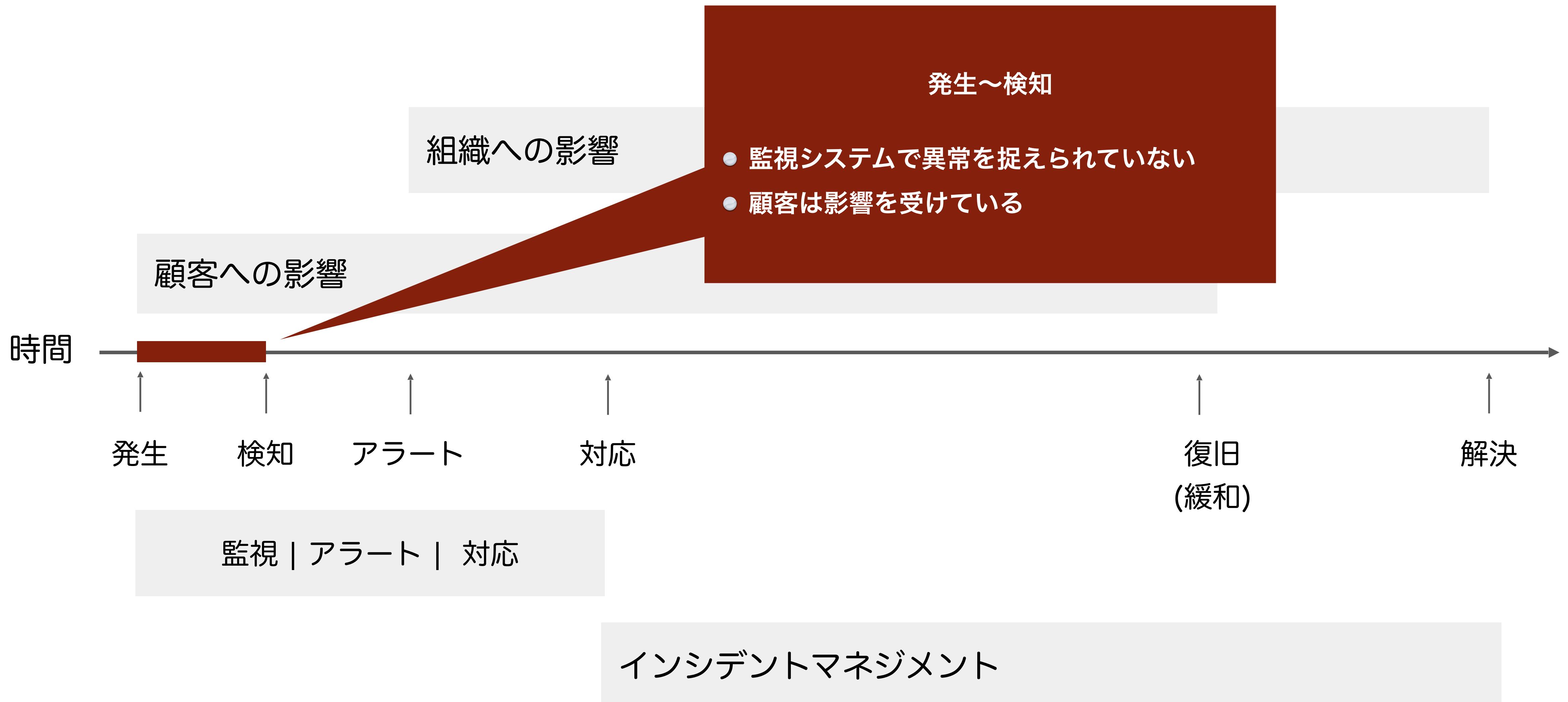
また、ツールの開発・普及と合わせて セキュリティ対策チームとして信頼を獲得にも努めてました

- 全てのインシデント対応チャンネルに join する
  - CTO, VPoE, セキュリティ対策チームは必ず invite される
- 自分がトラブル解決できる機会を拾っていく

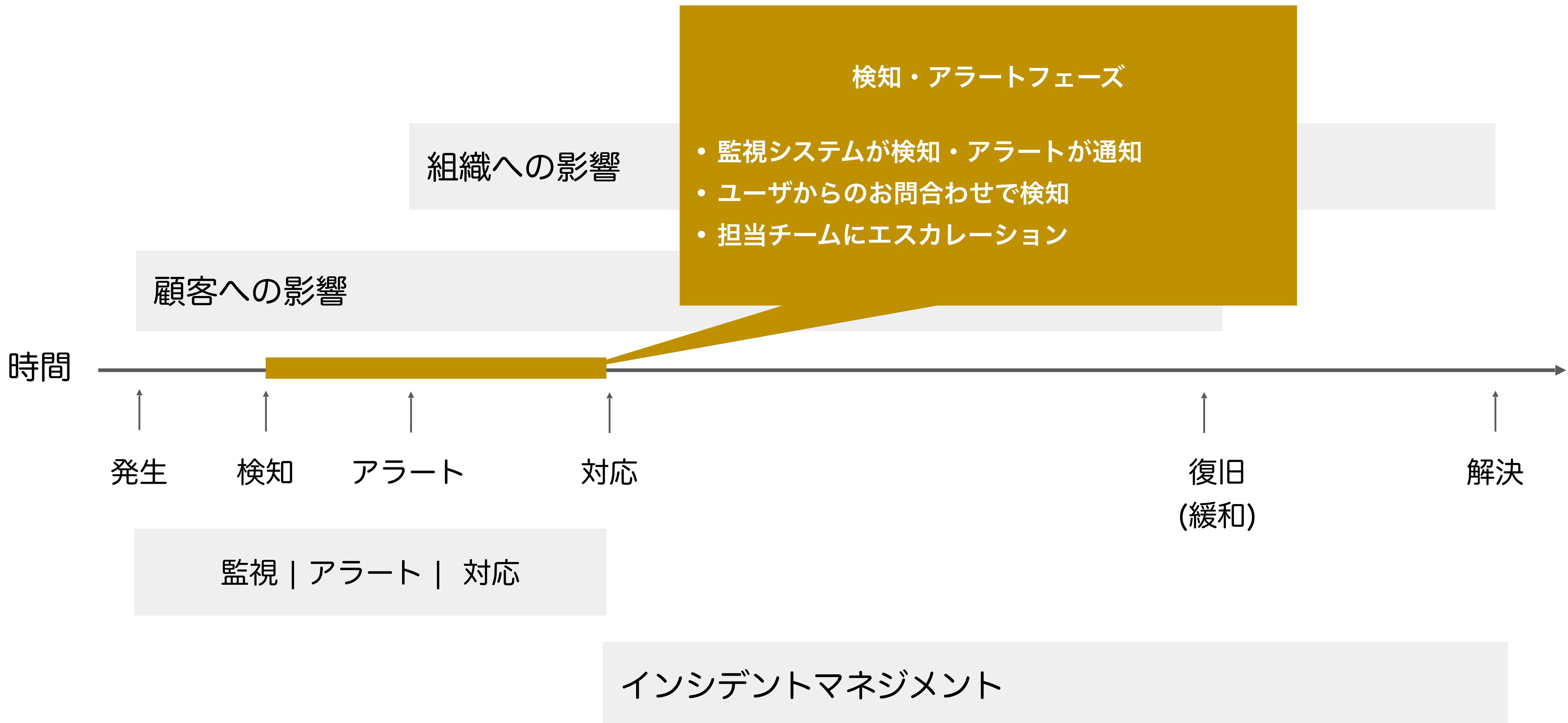
インシデント対応ツールを一気に導入してたところで、モデルやプラクティスも含めて広めていけたか、あるいは私自身が学び得ていたか、は考えてしまうところがあります。



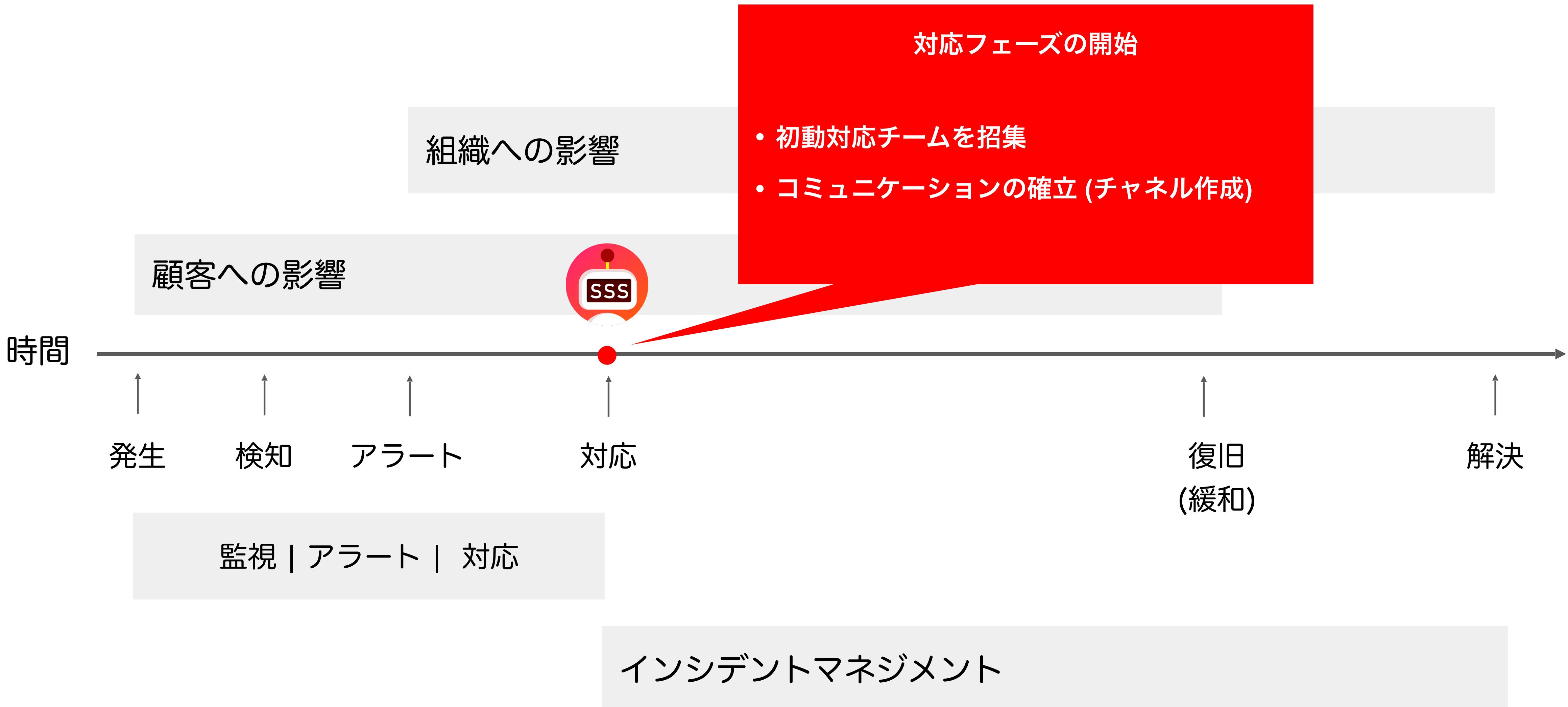
# インシデント対応のタイムライン



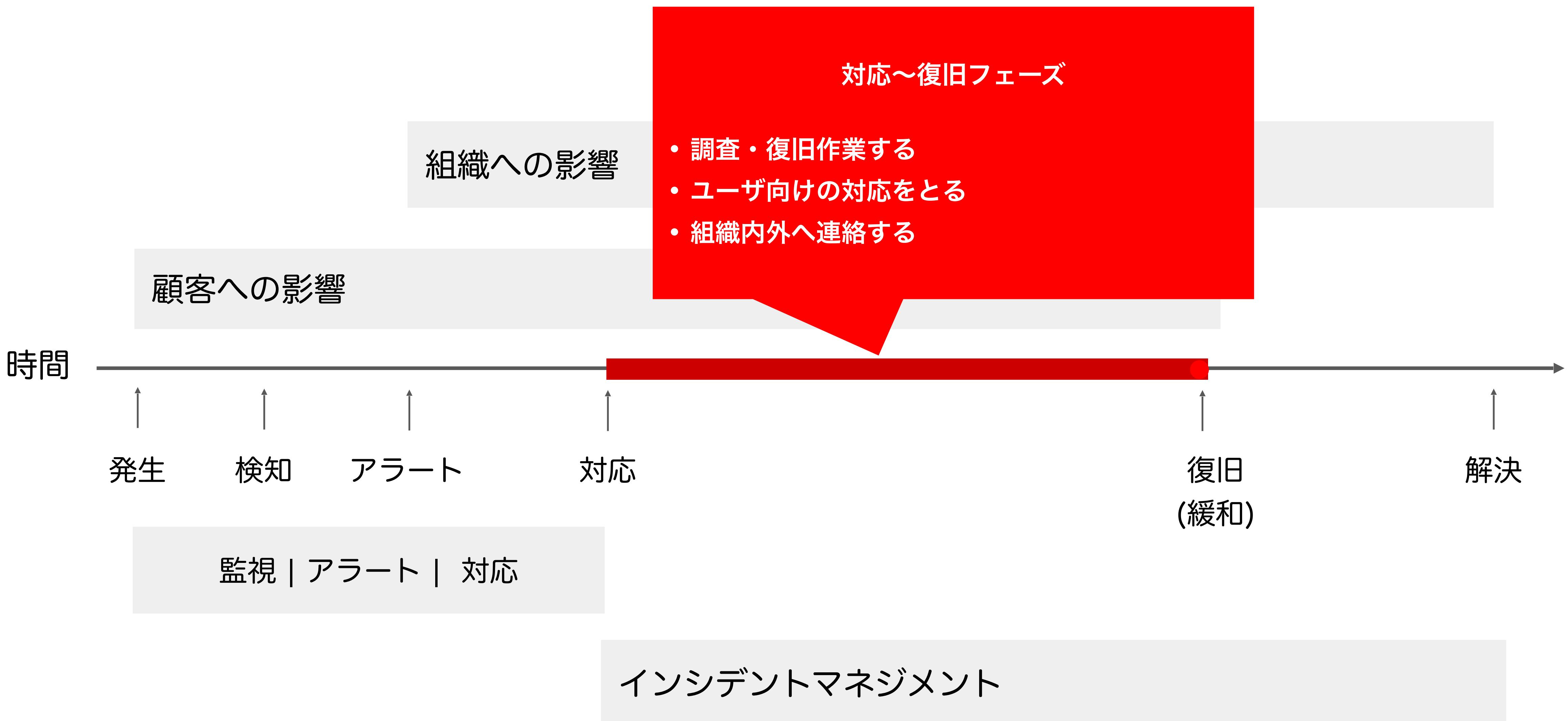
# インシデント対応のタイムライン



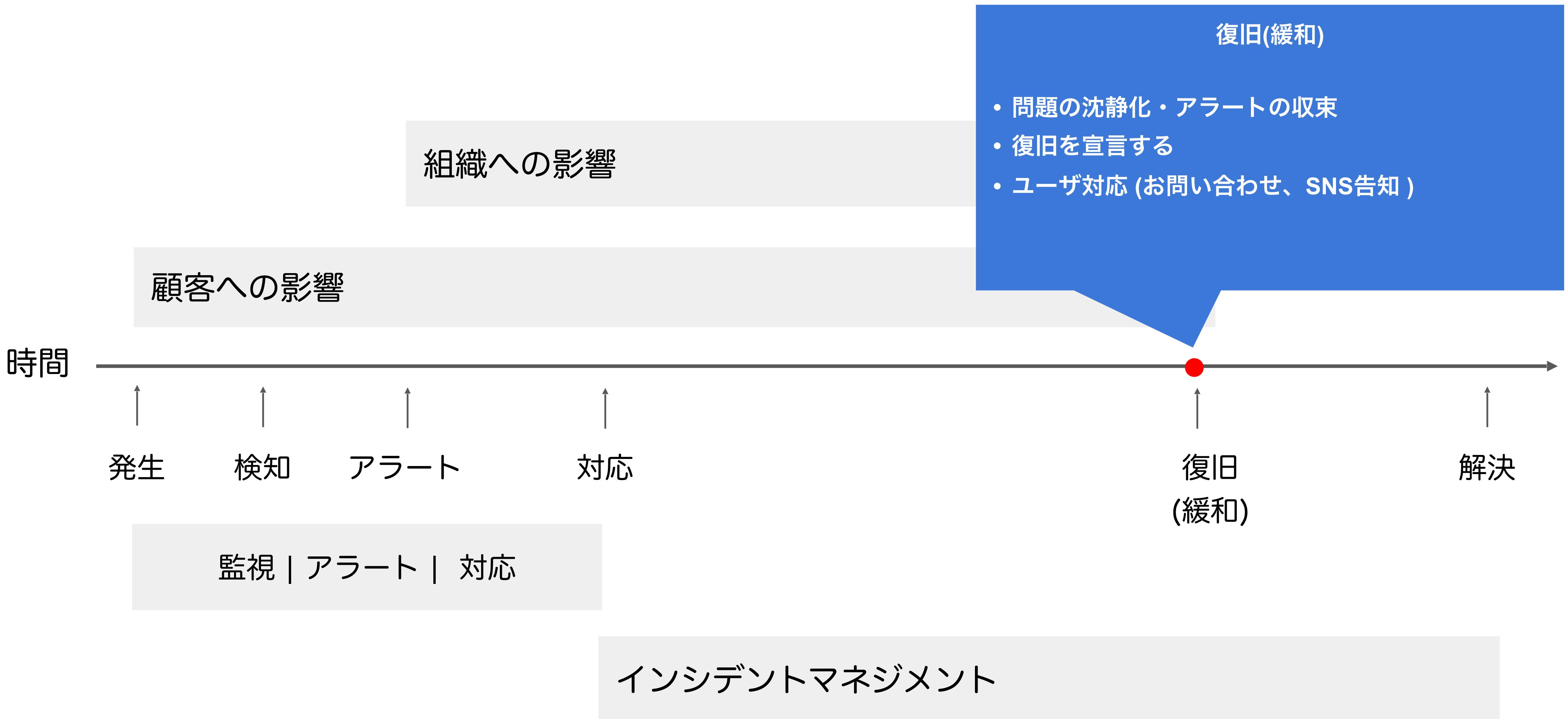
# インシデント対応のタイムライン



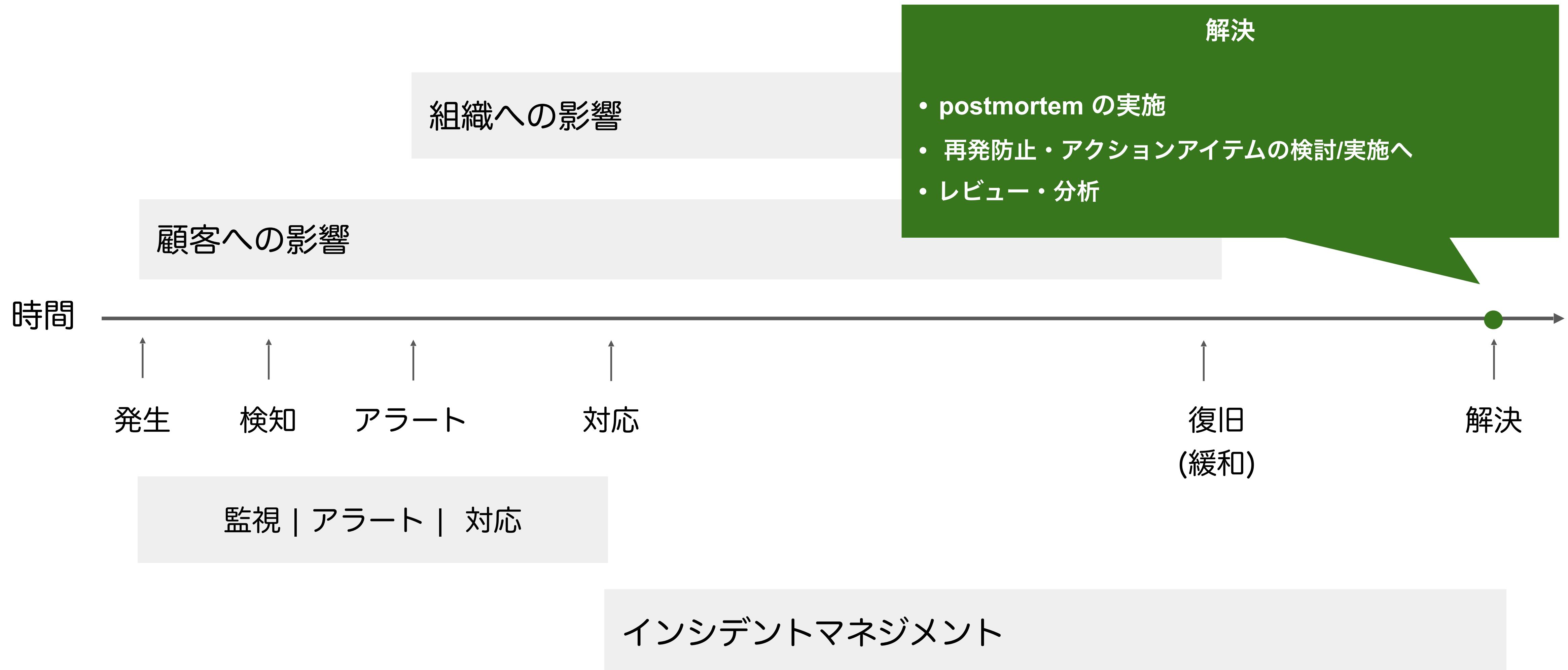
# インシデント対応のタイムライン



# インシデント対応のタイムライン



# インシデント対応のタイムライン



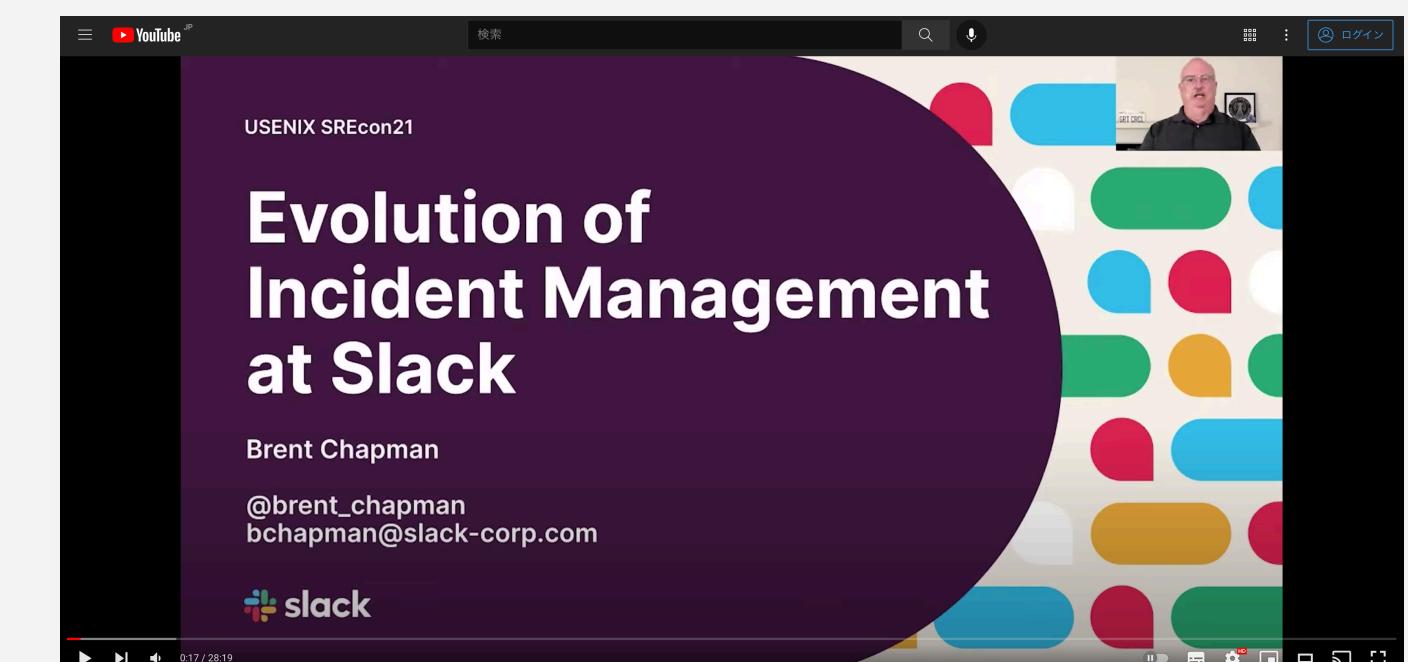
## (補足) Brent Chapman さん

“Google社の伝説的なSRE組織に所属していたとき、Incident Management at Google (IMAG)システムを構築・立ち上げ、現在は全社的に緊急事態管理に使用されている。”

“ブレントはまた、元航空捜索救助パイロットおよび事件指揮官、主要な芸術および音楽祭およびイベントの緊急派遣および派遣監督者、ならびにコミュニティ緊急対応チーム（CERT）のメンバーおよびインストラクターでもあります。”

“Majordomo（世界中の何十万ものサイトで使用されているメーリングリスト管理ソフトウェア）…などのオープンソースソフトウェアの開発者でもある。

現在は Slack のインシデントマネジメント携わっているようです。  
Slackについて話をした SRECon21 の動画が 2021/10/15 付けて公開されています



出典: <https://greatcircle.com/im/>

出典: <https://www.youtube.com/watch?v=FYYTgIQoS3w>

“システム設計がより複雑になり、様々なタイプの専門家の協調と統合が必要になるにつれて、システムの動作や行動を理解し、結果を予想することはますます困難になってきた”



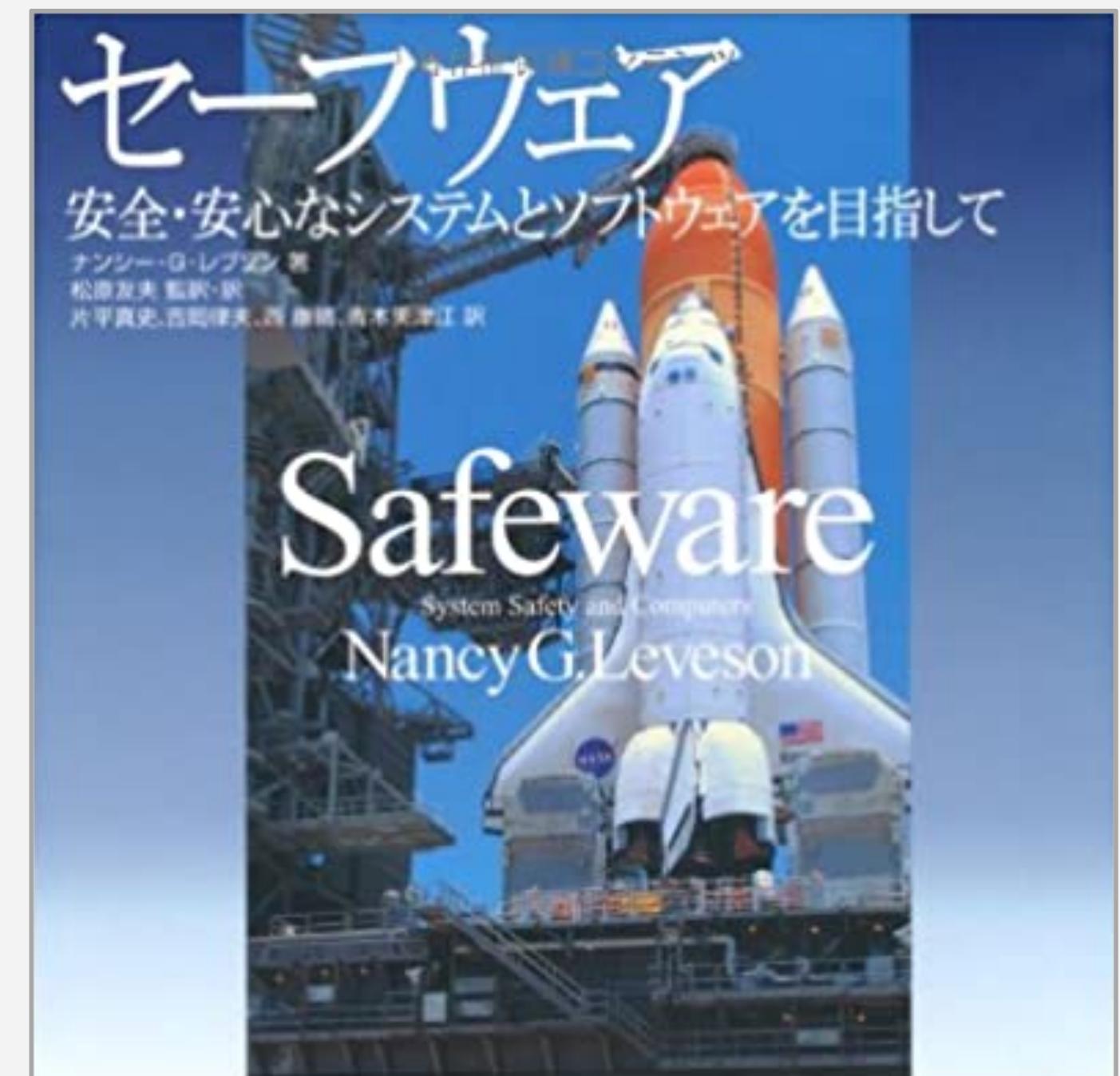
システムハザード＝コンピュータ制御システムに  
忍び寄る危険。解決の処方箋はここに！



NASAのスペースシャトル事故調査委員会で貴重な改善点を数多く提供してきた  
ナンシー・G・レブソン氏が、システムとソフトウェアの安全に携わるすべての読者に贈る珠玉の1冊

進成を誇る西田徹也著『システム工学』に並ぶ世界的大名著 SE Architects Archive

“自動化によってオペレータのエラーのリスクは低減されると思うかもしれないが、保守や修理といった業務へシフトしたり、より高度な監視制御や意思決定へとシフトしていくに過ぎず、自動化によってシステムから人間がいなくなることないというのが真実である”



システムハザード＝コンピュータ制御システムに  
忍び寄る危険。解決の処方箋はここに！



NASAのスペースシャトル事故調査委員会で貴重な改善点を数多く提供してきた  
ナンシー・G・レブソン氏が、システムとソフトウェアの安全に携わるすべての読者に贈る珠玉の1冊

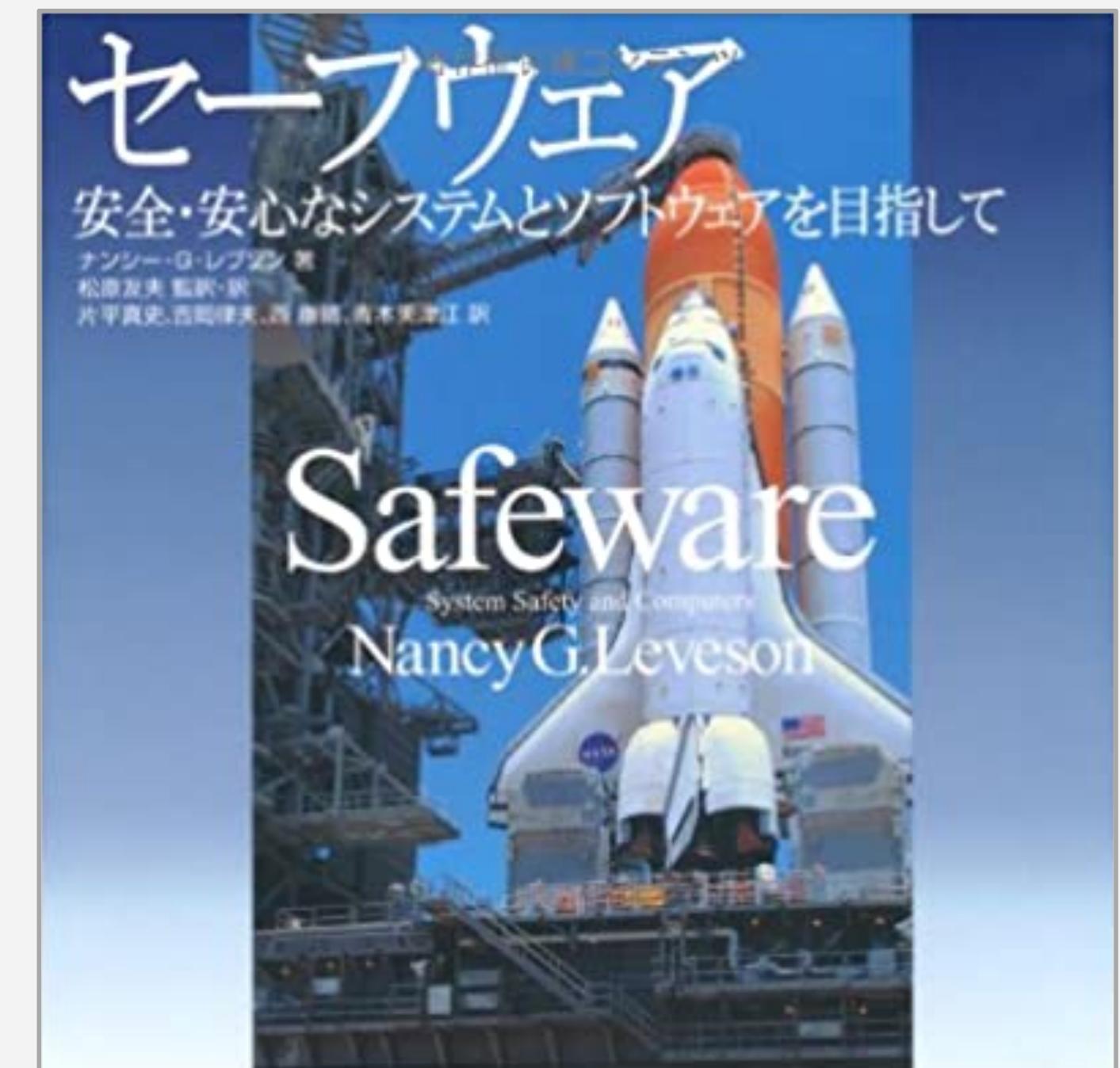
進成を誇る西田徹・久保義典著『System Safety』に並ぶ世界的名著

SE

Architects

Archive

“設計者は、システムの特性を完全に理解できないか、システムが稼働すべき環境条件を完全に予想できないことがある。事象が予見できなかつたか、不適切に扱われたか、または設計者の切り捨てレベル以下であったために、普段は確率が低い事象が、自動化システムが防御措置を講じないまま残されるだろう”



システムハザード＝コンピュータ制御システムに  
忍び寄る危険。解決の処方箋はここに！



NASAのスペースシャトル事故調査委員会で貴重な改善点を数多く提供してきた  
ナンシー・G・レブソン氏が、システムとソフトウェアの安全に携わるすべての読者に贈る珠玉の1冊

進成を誇る西田徹・堀潤・コンイケ元に幸ひも世界的名著 SE Architects Archive

## オペレータはしばしば極限状態で介入せねばならない(1)

“オペレータは、起こり得る緊急事態の克服を期待されるだけでなく、成功しない場合の影響が重大になり得る場合は、しばしばシステムが対応できる極限状態での介入が求められる。緊急事態には設計者が想定しなかった状況や、オペレータの訓練では対処できない状況が含まれる。”



## オペレータはしばしば極限状態で介入せねばならない(1)

“オペレータは、状況を速やかに診断し対応せねばならないだけでなく、極度の緊張の下で、システムの状態について限られた情報しかない状況下で、創造性と工夫の才を用いてそれをやらなければならない。われわれは、彼らが失敗した時に驚くのではなく、こうした状況の下で、人間がしばしばきわめてうまくやっていることに驚嘆すべきなのである。”

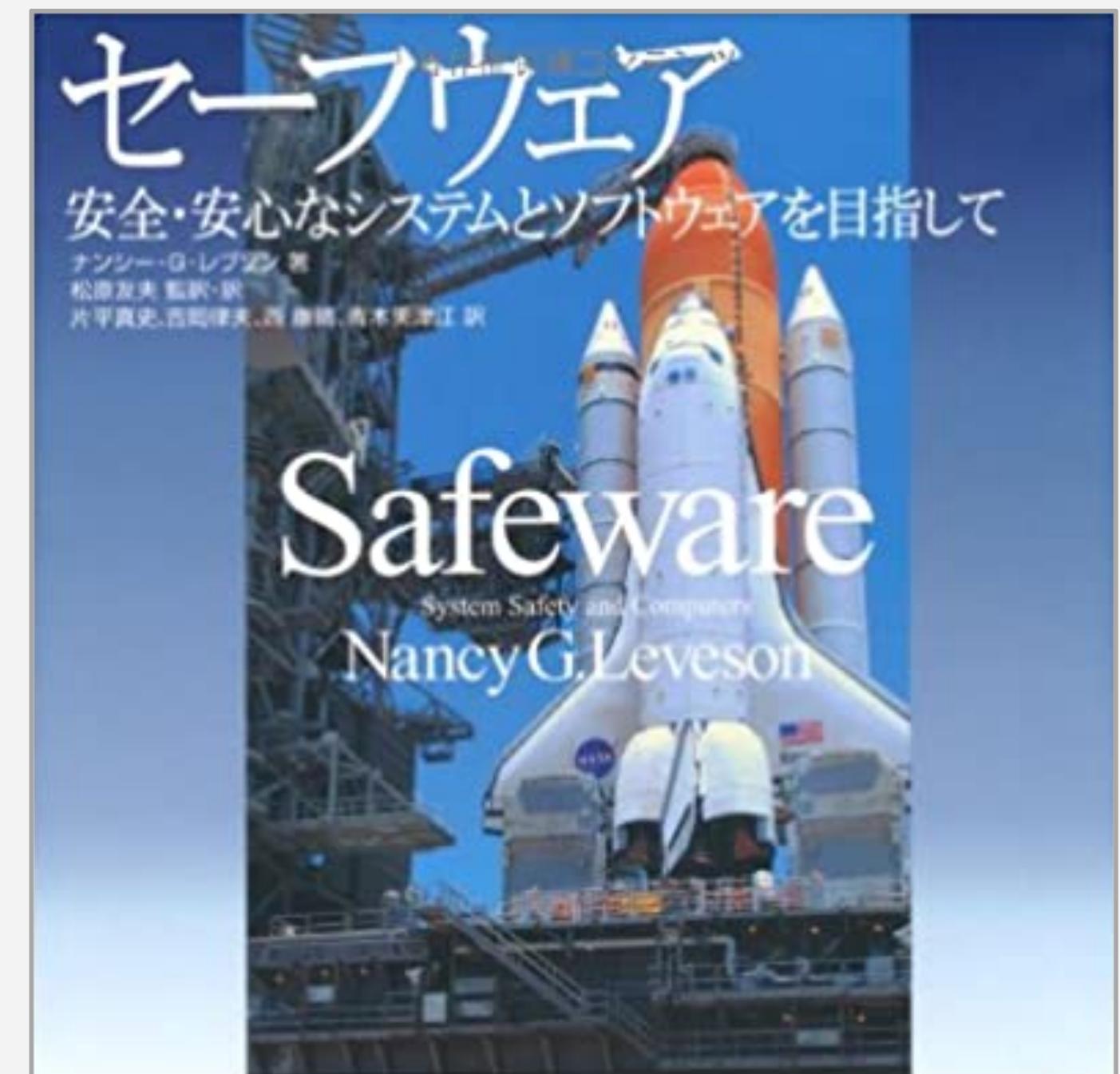


# 自動化システムにおける人間の必要性

- ① めったに起こらない事象の確率を評価するのが難しいこと
- ② 先入観を持って副作用について考えること
- ③ 偶発事象を見落とす傾向があること
- ④ システムの限られた数の観点だけに集中して複雑さを抑える傾向がある
- ⑤ 複雑な関係を把握する能力に限界があること



“自動化によってオペレータのエラーのリスクは低減されると思うかもしれないが、保守や修理といった業務へシフトしたり、より高度な監視制御や意思決定へとシフトしていくに過ぎず、自動化によってシステムから人間がいなくなることないというのが真実である”



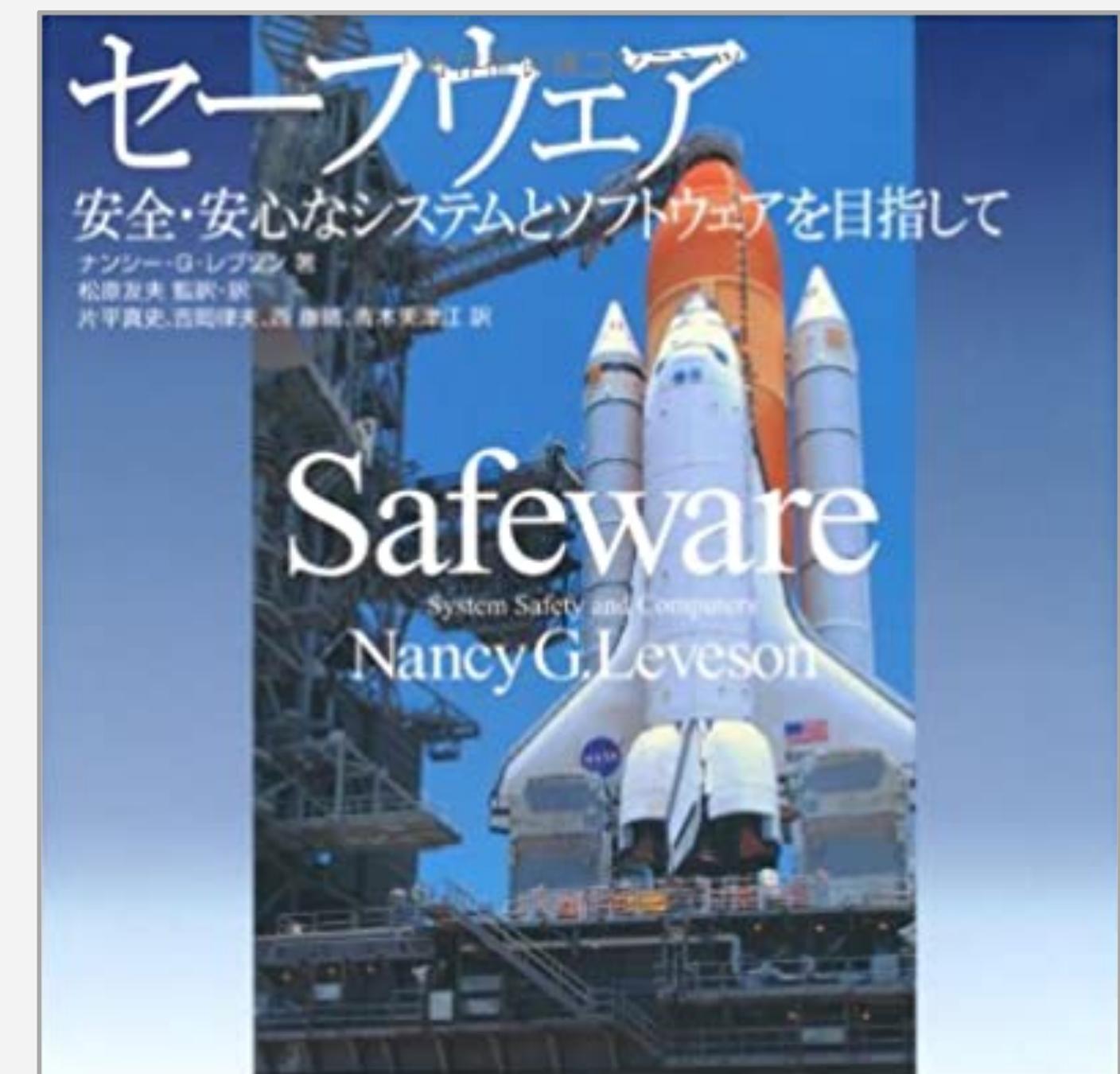
システムハザード＝コンピュータ制御システムに  
忍び寄る危険。解決の処方箋はここに！



NASAのスペースシャトル事故調査委員会で貴重な改善点を数多く提供してきた  
ナンシー・G・レブソン氏が、システムとソフトウェアの安全に携わるすべての読者に贈る珠玉の1冊

進成を誇る西田徹・久保義典著『世界的名著 SE Architects Archive』

“設計者は、システムの特性を完全に理解できないか、システムが稼働すべき環境条件を完全に予想できないことがある。事象が予見できなかつたか、不適切に扱われたか、または設計者の切り捨てレベル以下であったために、普段は確率が低い事象が、自動化システムが防御措置を講じないまま残されるだろう”



システムハザード＝コンピュータ制御システムに  
忍び寄る危険。解決の処方箋はここに！



NASAのスペースシャトル事故調査委員会で貴重な改善点を数多く提供してきた  
ナンシー・G・レブソン氏が、システムとソフトウェアの安全に携わるすべての読者に贈る珠玉の1冊

進成を誇る西田徹・堀潤・コンイケ元に幸ひも世界的名著 SE Architects Archive