LOG680
Introduction à l'approche DevOps

Activer et injecter l'apprentissage dans le travail quotidien

The DevOps Handbook

Part V, Chap 19

Objectifs d'apprentissage

- Expliquer en quoi consiste le Chaos Monkey de Netflix. Pourquoi le Chaos Monkey a-t-il été développé?
- Expliquer pourquoi il est important de créer une culture d'apprentissage juste (Just Culture)
- Expliquer le rôle des réunions post-mortem. Quelles sont les activités principales d'une réunion post-mortem et qui devrait y participer?
- Pourquoi est-il important de publier les résultats des réunions post mortem à l'ensemble de l'organisation?
- Expliquer en quoi consiste Game Days. Quels sont ses avantages?

Liste des sujets

- Introduction : Partie V
- Introduction : Chap 19
- Culture d’apprentissage juste
- Réunions post-mortem
- Publication des post-mortem
- Réduire la tolérance aux incidents
- Redéfinir l'échec et encourager la prise de risques calculée
- Injection d’échec de production
- Utilisation de "Game Days"
- Conclusion

- **Introduction : Partie V**
- Introduction : Chap 19
- Culture d’apprentissage juste
- Réunions post-mortem
- Publication des post-mortem
- Réduire la tolérance aux incidents
- Redéfinir l'échec et encourager la prise de risques calculée
- Injection d’échec de production
- Utilisation de "Game Days"
- Conclusion

Introduction : Partie V

- Dans la partie III, "The **First Way**: les pratiques techniques de **flux**", nous avons discuté de la mise en œuvre des pratiques requises pour créer un **flux rapide dans notre flux de valeur**
- Dans la partie IV, "The **Second Way**: les pratiques techniques de **rétroaction**", notre objectif était de **créer le plus de rétroaction possible**, dans le plus grand nombre de domaines de notre système possible - plus tôt, plus rapidement et à moindre coût
- Dans la partie V, "**The Third Way: les pratiques techniques d'apprentissage**", nous présentons les **pratiques qui créent des possibilités d'apprentissage aussi rapidement, fréquemment que possible, à moindre coût et le plus tôt possible**
  - Cela implique de **tirer des enseignements des accidents et des échecs**, qui sont inévitables lorsque nous travaillons au sein de systèmes complexes
  - ainsi que d'**organiser et de concevoir nos systèmes de travail** de manière à constamment **expérimenter et apprendre**, tout en renforçant la sécurité de nos systèmes
  - Les résultats incluent une **plus grande résilience** et une **connaissance collective sans cesse croissante** de la manière dont notre système fonctionne réellement, afin que nous puissions mieux atteindre nos objectifs

Introduction : Partie V

- Dans les chapitres suivants, nous institutionnaliserons des rituels qui renforcent la sécurité, l'amélioration continue et l'apprentissage en procédant comme suit:
  - **Établir une culture juste** pour rendre la sécurité possible
  - **Injecter les échecs de production** pour créer de la résilience
  - **Convertir les découvertes locales en améliorations globales**
  - **Réserver du temps pour créer des améliorations et un apprentissage organisationnelle**

- Nous allons également créer des mécanismes pour que tout nouvel apprentissage généré dans un domaine de l'organisation puisse être rapidement utilisé dans l'ensemble de l'organisation
- Ainsi, **non seulement nous apprenons plus vite que nos concurrents**, ce qui nous permet de gagner sur le marché, mais **nous créons également une culture de travail plus sûre et plus résiliente**, à laquelle les gens sont ravis de participer et qui les aide à atteindre leur plus haut potentiel

- Introduction : Partie V
- **Introduction : Chap 19**
- Culture d’apprentissage juste
- Réunions post-mortem
- Publication des post-mortem
- Réduire la tolérance aux incidents
- Redéfinir l'échec et encourager la prise de risques calculée
- Injection d’échec de production
- Utilisation de "Game Days"
- Conclusion

Introduction

- Lorsque nous travaillons dans un **système complexe**, il nous est **impossible de prédire tous les résultats pour les actions que nous prenons**
- Cela contribue à des **accidents inattendus** et **parfois catastrophiques**, même lorsque nous utilisons des outils de précaution statiques, tels que des listes de contrôle, qui codifient notre compréhension actuelle du système
- Pour nous permettre de travailler en toute sécurité au sein de systèmes complexes, nos **organisations** doivent devenir de plus en plus **capables d'autodiagnostic et d'auto-amélioration**, et doivent être **capables de détecter les problèmes**, **de les résoudre** et d'en **multiplier les effets** en rendant les solutions disponibles dans l'ensemble de l'organisation
- Cela crée un **système d'apprentissage dynamique** qui nous permet de comprendre nos erreurs et de traduire cette compréhension en actions qui empêchent ces erreurs de se reproduire à l'avenir

Introduction

- Le résultat est ce que le Dr Steven Spear décrit comme des **organisations résilientes**, "**habiles à détecter les problèmes, à les résoudre et à multiplier les effets en rendant les solutions disponibles dans toute l'organisation**"

- Ces organisations peuvent se soigner d'elles-mêmes

  « **Pour une telle organisation, réagir aux crises n'est pas un travail idiosyncratique. C'est quelque chose qui se fait tout le temps. C'est cette réactivité qui est leur source de fiabilité.** »

Cloud Native Netflix

- Un exemple frappant de l'incroyable résilience pouvant résulter de ces principes et pratiques a été vu le 21 avril 2011, lorsque toute la zone de disponibilité Amazon AWS US-EAST est tombé, éliminant la quasi-totalité de leurs clients qui en dépendaient, y compris Reddit et Quora
- Netflix était une exception surprenante, apparemment insensible à cette panne massive d'AWS
- Après l'événement, de nombreuses spéculations ont eu lieu sur la manière dont Netflix assurait le bon fonctionnement de leurs services
  - Selon une théorie répandue, Netflix étant l'un des plus gros clients d'Amazon Web Services, un traitement spécial leur a permis de continuer à fonctionner
  - Cependant, un article de blog de Netflix Engineering expliquait que **c'était leur décision de conception architecturale en 2009 qui leur permettait une résilience exceptionnelle**

Cloud Native Netflix

- En 2008, le service de diffusion vidéo en ligne de Netflix s'exécutait sur une application J2EE monolithique hébergée dans l'un de leurs centres de données
- Cependant, à partir de 2009, ils ont commencé à réorganiser ce système pour devenir ce qu'ils ont appelé le "cloud native »
  - Conçu pour **fonctionner entièrement dans le cloud public Amazon** et pour **être suffisamment résilient pour survivre à d'importantes défaillances**
- L'un de leurs **objectifs de conception** spécifiques était de **garantir le fonctionnement continu des services Netflix, même en cas de panne complète d'une zone de disponibilité AWS, comme c'est le cas avec US-EAST**
  - Pour ce faire, il fallait que leur système soit faiblement couplé, chaque composant ayant des délais d'attente ("timeout") agressifs afin de garantir que les défaillances de composants ne fassent pas tomber tout le système
  - Au lieu de cela, **chaque fonctionnalité** et **chaque composant** ont été **conçus pour se dégrader en douceur ("gracefully degrade")**
    - Par exemple, lors des pics de trafic générant des pics d'utilisation du processeur, au lieu d'afficher une liste de films personnalisés à l'utilisateur, ils affichaient un contenu statique, tel que des résultats en cache ou non personnalisés, qui nécessitait moins de calcul

Cloud Native Netflix

- En plus de mettre en œuvre ces modèles architecturaux, **Netflix avait également créé et exécutait un service surprenant et audacieux appelé Chaos Monkey**
  - Simulait les défaillances d'AWS en éliminant de manière constante et aléatoire des serveurs de production
  - Souhaitaient que «toutes les équipes d'ingénierie s'habituent à un niveau constant de défaillance dans le cloud» afin que les services puissent «se rétablir automatiquement sans aucune intervention manuelle»
- En d'autres termes, **l'équipe Netflix a utilisé Chaos Monkey pour s'assurer qu'elle avait atteint ses objectifs de résilience opérationnelle**, en injectant constamment des défaillances dans ses environnements de pré-production et de production
- Comme on pouvait s'y attendre, lorsqu'ils ont utilisé Chaos Monkey pour la première fois dans leurs environnements de production, les services ont échoué comme ils ne l'auraient jamais prévu ou imaginé
  - En recherchant et en corrigeant constamment ces problèmes pendant les heures de travail normales, les ingénieurs de Netflix ont rapidement et de manière itérative créé un service plus résistant
  - tout en créant simultanément des apprentissages organisationnels (pendant les heures de travail normales!) qui leur ont permis de faire évoluer leurs systèmes bien au-delà de leurs concurrents

Introduction

- Le Chaos Monkey n'est qu'un exemple de la manière dont l'apprentissage peut être intégré au travail quotidien
- L'histoire montre également comment **les organisations qui apprennent considèrent les échecs, les accidents et les erreurs comme une opportunité d'apprentissage et non comme une punition**
- Ce chapitre explore les moyens de créer un système d'apprentissage et d'établir une culture juste, ainsi que de répéter et de créer de manière routinière des échecs pour accélérer l'apprentissage

- Introduction : Partie V
- Introduction : Chap 19
- **Culture d’apprentissage juste**
- Réunions post-mortem
- Publication des post-mortem
- Réduire la tolérance aux incidents
- Redéfinir l'échec et encourager la prise de risques calculée
- Injection d’échec de production
- Utilisation de "Game Days"
- Conclusion

Établir une culture d'apprentissage juste

- L'une des **conditions préalables à une culture d'apprentissage** est que, **lorsqu'un accident survient** (ce qui va certainement arriver), **la réaction à cet accident est considérée comme «juste»**
- Le Dr Sidney Dekker, qui a contribué à codifier certains des éléments clés de la culture de la sécurité et défini le terme "Just Culture" (culture juste), écrit:

  «**Lorsque les réactions aux incidents et aux accidents sont considérées comme injustes**, elles peuvent **entraver les enquêtes de sécurité**, **susciter la peur** plutôt que la vigilance chez les personnes effectuant un travail essentiel pour la sécurité, rendant les organisations plus bureaucratiques plutôt que plus prudentes, et **favoriser une culture de secret professionnel, d'évasion et d'autoprotection**. »

- Cette notion de punition est présente, de manière subtile ou évidente, dans la façon dont de nombreux gestionnaires ont fonctionné au cours du siècle dernier
  - Selon cette façon de penser, pour atteindre les objectifs de l'organisation, les dirigeants doivent commander, contrôler, établir des procédures permettant d'éliminer les erreurs et d'appliquer le respect de ces procédures

Établir une culture d'apprentissage juste

- Dr. Dekker appelle cette notion "d'élimination de l'erreur en éliminant les personnes qui ont causé l'erreur", la **théorie de la pomme pourrie** (*Bad Apple Theory*)
- Il affirme que cela est invalide, car

  «**L'erreur humaine n'est pas notre cause de problèmes; l'erreur humaine est plutôt une conséquence de la conception des outils que nous leur avons fournis**. »

- Si les accidents ne sont pas causés par des pommes pourries, mais plutôt par des problèmes de conception inévitables dans le système complexe que nous avons créé, alors **au lieu de «nommer, blâmer et humilier»** la personne qui a causé la défaillance, notre objectif devrait toujours être de **maximiser les possibilités d'apprentissage organisationnel**, en renforçant continuellement le fait que nous valorisons les actions qui exposent et partagent plus largement les problèmes de notre travail quotidien
  - C'est ce qui nous permet d'améliorer la qualité et la sécurité du système dans lequel nous opérons et de renforcer les relations entre tous les utilisateurs de ce système

Établir une culture d'apprentissage juste

- En **transformant l'information en connaissances** et en **intégrant les résultats de l'apprentissage dans nos systèmes**, **nous commençons à atteindre les objectifs d'une culture juste**, en équilibrant les besoins de sécurité et de responsabilité
  - Comme le déclare John Allspaw (CTO d'Etsy) «Our goal at Etsy is to view mistakes, errors, slips, lapses, and so forth with a perspective of learning. »

- Lorsque les ingénieurs commettent des erreurs et se sentent en sécurité lorsqu'ils fournissent des détails à ce sujet, ils ne sont pas seulement disposés à être tenus pour responsables, ils sont également enthousiastes à aider le reste de l'organisation à éviter la même erreur à l'avenir

  => C'est ce qui crée l'apprentissage organisationnel.

- D'un autre côté, si nous punissons cet ingénieur, tout le monde n'est pas incité à fournir les détails nécessaires pour comprendre le mécanisme, la pathologie et le fonctionnement de la défaillance, ce qui garantit que la défaillance se reproduira

Établir une culture d'apprentissage juste

- **Deux pratiques efficaces** qui contribuent à **créer une culture juste basée sur l'apprentissage** sont
  - **Post-mortem sans-reproche** (Blameless post-mortem)
  - **Introduction contrôlée d'échecs** (Control introduction of failures) dans la production afin de créer des opportunités de résoudre les problèmes qui se produisent de façon inévitable dans les systèmes complexes

- Nous allons d'abord examiner les post-mortem sans reproche et examiner ensuite les raisons pour lesquelles un échec peut être une bonne chose

- Introduction : Partie V
- Introduction : Chap 19
- Culture d’apprentissage juste
- **Réunions post-mortem**
- Publication des post-mortem
- Réduire la tolérance aux incidents
- Redéfinir l'échec et encourager la prise de risques calculée
- Injection d’échec de production
- Utilisation de "Game Days"
- Conclusion

Réunions post-mortem après les accidents

- Pour aider à instaurer une culture juste, lorsque des accidents et des incidents importants se produisent (e.g., déploiement défaillant, problème de production affectant les clients), nous **devons procéder à une évaluation post-mortem sans-reproche après la résolution de l'incident**
  - Une évaluation post-mortem sans-reproche (terme défini par John Allspaw) nous aide à examiner «les erreurs de manière centrée sur les aspects situationnels du mécanisme de l'échec et sur le processus de prise de décision des personnes proches de l'échec»

- Pour ce faire, nous **planifions le post-mortem le plus tôt possible après l'accident** et avant que les souvenirs et les liens entre la disparition de cause à effet et les circonstances ne changent
  - Nous attendons bien sûr que le problème soit résolu pour ne pas distraire les personnes qui travaillent encore activement sur le sujet

Réunions post-mortem après les accidents

- Lors de la réunion post-mortem sans reproche, **nous ferons ce qui suit**:
  - **Construire une chronologie et rassembler les détails de plusieurs points de vue** sur les échecs, en veillant à ne pas punir les gens qui font des erreurs
  - **Permettre à tous les ingénieurs d'améliorer la sécurité** en leur permettant de rendre compte en détail de leurs contributions aux défaillances
  - **Permettre et encourager les auteurs d'erreurs à devenir des experts qui éduquent** le reste de l'organisation sur la façon de ne pas les commettre à l'avenir
  - **Accepter qu'il existe toujours un espace discrétionnaire** où les humains peuvent décider d'agir ou non, et que le jugement de ces décisions repose sur le recul
  - **Proposer des contre-mesures** pour empêcher qu'un accident similaire ne se reproduise et **assurer que ces contre-mesures sont enregistrées** avec une date cible et un propriétaire pour suivi

Réunions post-mortem après les accidents

- Pour nous permettre d'acquérir cette compréhension, **les parties prenantes suivantes doivent être présentes à la réunion**:
  - Les personnes impliquées dans les décisions qui ont pu contribuer au problème
  - Les personnes qui ont identifié le problème
  - Les personnes qui ont répondu au problème
  - Les personnes qui ont diagnostiqué le problème
  - Les personnes touchées par le problème
  - Et toute autre personne intéressée à assister à la réunion.
- Notre **première tâche** dans la réunion post-mortem sans reproche est d'**enregistrer notre meilleure compréhension de la chronologie des événements pertinents tels qu'ils se sont produits**. Cela inclut
  - Toutes les **actions que nous avons entreprises** et **à quelle heure** (idéalement soutenu par des journaux de discussion, tels que IRC ou Slack)
  - Quels **effets nous avons observés** (idéalement sous la forme de métriques spécifiques de notre télémétrie de production, par opposition à des récits simplement subjectifs)
  - **Toutes voies d'investigation que nous avons suivies** et quelles **solutions ont été envisagées**

Réunions post-mortem après les accidents

- Pour obtenir ces résultats, nous devons faire preuve de **rigueur dans l'enregistrement des détails** et renforcer une **culture selon laquelle l'information peut être partagée sans crainte de représailles**
  - Pour cette raison, en particulier pour nos quelques premières missions post-mortem, il peut être utile que la réunion soit dirigée par un animateur qualifié qui n'a pas été impliqué dans l'accident
  - Au cours de la réunion et de la résolution ultérieure, nous devrions **explicitement interdire les expressions «aurait» ou «aurait pu»**, car il s'agit d'énoncés contrefactuels résultant de notre tendance humaine à créer des alternatives possibles aux événements déjà survenus
    - Des déclarations contrefactuelles, telles que **«J'aurais pu…» ou «Si j'avais su cela, j'aurais dû…»,** décrivent le problème en termes de système tel qu'il est imaginé plutôt qu'en termes de système qui existe réellement, qui est le contexte auquel nous devons nous limiter. Voir annexe 8

Réunions post-mortem après les accidents

- L'un des résultats potentiellement surprenants de ces réunions est que les gens se reprochent souvent des choses indépendantes de leur volonté ou remettent en question leurs propres capacités
  - Ian Malpass (ingénieur chez Etsy) observe:

    «À ce moment-là, lorsque nous faisons quelque chose qui provoque la chute de tout le site, nous obtenons un sentiment "d'eau glacée sur la tête", et probablement notre première pensée est "I suck and I have no idea what I'm doing" . Nous devons nous en empêcher, car c'est un chemin vers la folie, le désespoir et le sentiment d'être un imposteur, ce qui nous ne doit pas arriver à de bons ingénieurs. La meilleure question à se poser est la suivante: «Pourquoi cela me semblait la chose logique à faire lorsque j'ai pris la décision?»

- Lors de la réunion, nous devons **prévoir suffisamment de temps pour réfléchir et décider des mesures à prendre**
- Une fois que les contre-mesures ont été identifiées, elles doivent être **classées par ordre de priorité, assignées à un responsable et associées à un calendrier de mise en œuvre**
- Cette démarche démontre que nous attachons plus d'importance à l'amélioration de notre travail quotidien qu'au travail quotidien

Réunions post-mortem après les accidents

- Dan Milstein (ingénieurs chez Hubspot) écrit qu'il commence toutes les réunions post-mortem sans reproche en disant:

  « **Nous essayons de nous préparer pour un avenir où nous sommes aussi stupides que nous le sommes aujourd'hui.**

  En d'autres termes, il n'est pas acceptable d'avoir une contre-mesure qui consiste simplement à «faire plus attention» ou à «être moins stupide».

  Au lieu de cela, nous **devons concevoir de véritables contre-mesures pour éviter que de telles erreurs ne se reproduisent**. »

- Parmi ces **contre-mesures**, citons
  - **Création de nouveaux tests automatisés** pour détecter les conditions dangereuses dans notre pipeline de déploiement
  - **Ajout de télémétrie de production**
  - **Identification de catégories de modifications nécessitant une évaluation supplémentaire par les pairs**
  - **Répétition de cette catégorie d'échec dans le cadre d'exercices réguliers**

- Introduction : Partie V
- Introduction : Chap 19
- Culture d’apprentissage juste
- Réunions post-mortem
- **Publication des post-mortem**
- Réduire la tolérance aux incidents
- Redéfinir l'échec et encourager la prise de risques calculée
- Injection d’échec de production
- Utilisation de "Game Days"
- Conclusion

Publication des post-mortem

- Après avoir tenu une réunion post-mortem sans reproche, nous devrions annoncer largement la disponibilité des notes de réunion et de tout artefact associé (par exemple, chronologie, journaux de discussion IRC, communications externes)
  - Ces informations devraient (idéalement) être placées dans un emplacement centralisé où toute l'organisation peut y accéder et tirer les leçons de l'incident
  - La post-mortem est tellement importante que nous pouvons même interdire la clôture des incidents de production jusqu'à la fin de la réunion post-mortem
- Cela nous **aide à traduire les acquis et les améliorations locales en acquis et améliorations globales**
  - Randy Shoup (ancien directeur technique de Google App Engine), explique comment la documentation de réunions post-mortem peut avoir une valeur inestimable pour les autres membres de l'organisation:

    «Comme vous pouvez l'imaginer **chez Google, tout est consultable**.

    Tous les documents post-mortem se trouvent à des endroits où d'autres personnes de Google peuvent les voir.

    Et croyez-moi, quand un groupe a un incident qui ressemble à quelque chose qui s'est passé auparavant, ces documents post-mortem font partie des premiers documents lus et étudiés. »

Publication des post-mortem

- **Publier de nombreux post-mortem et encourager les autres membres de l'organisation à les lire augmentent l'apprentissage organisationnel**
  - Il devient également de plus en plus courant que les sociétés de services en ligne publient des post-mortem pour les pannes touchant les clients
  - Cela augmente souvent considérablement la transparence que nous avons avec nos clients internes et externes, ce qui accroît leur confiance en nous
- Cette volonté de tenir autant de réunions post-mortem sans reproche que nécessaire chez Etsy a posé quelques problèmes:
  - au cours des quatre dernières années, Etsy a accumulé un grand nombre de notes de réunion post-mortem dans des pages de wiki, qui sont devenues de plus en plus difficiles à rechercher, sauvegarder et utiliser
- Pour résoudre ce problème, ils ont développé un outil appelé **Morgue** permettant
  - Enregistrer facilement les aspects de chaque accident, tels que le MTTR et la gravité de l'incident
  - Mieux gérer les fuseaux horaires (ce qui est devenu pertinent dès lors que davantage d'employés d'Etsy travaillaient à distance)
  - Enregistrer d'autres données, tels que le texte enrichi au format Markdown, les images incorporées, les balises et l'historique

Publication des post-mortem

- La **Morgue** a été conçue pour que l'équipe puisse **enregistrer facilement**:
  - Si le problème était dû à un incident programmé ou non programmé
  - Le propriétaire du post mortem
  - Journaux de discussion IRC (Internet Relay Chat) pertinents (particulièrement importants pour les problèmes survenant à 3 heures du matin lorsque la prise de notes précise risque de ne pas se produire)
  - Tickets JIRA pertinents pour les actions correctives et leurs dates d'échéance (informations particulièrement importantes pour la direction)
  - Liens vers les messages des forums clients (où les clients se plaignent de problèmes)
- **Après avoir développé et utilisé la Morgue, le nombre d'enquêtes post-mortem enregistrées chez Etsy a augmenté de manière significative par rapport au moment où elles utilisaient des pages wiki**, en particulier pour les incidents P2, P3 et P4 (c'est-à-dire des problèmes de gravité moindre)
  - Ce résultat renforce l'hypothèse selon laquelle s'ils facilitaient la documentation des post-mortem à l'aide d'outils tels que Morgue, davantage de personnes enregistreraient et détailleraient les résultats de leurs réunions post-mortem, permettant ainsi un apprentissage organisationnel plus approfondi

Publication des post-mortem

- Dr. Amy C. Edmondson (Professeure Novartis de Leadership and Management à la Harvard Business School et co-auteur de "Building the Future: Big Teaming for Audacious Innovation"), écrit:

  « Encore une fois, **le remède** - qui ne nécessite pas forcément beaucoup de temps et d'argent - **consiste à réduire les stigmates de l'échec.**

  Eli Lilly le fait depuis le début des années 90 en organisant des "**party d'échecs**" afin d'**honorer des expériences scientifiques intelligentes et de grande qualité qui ne donnent pas les résultats souhaités**.

  Les parties ne coûtent pas cher, et le redéploiement plus tôt que tard de précieuses ressources, notamment de scientifiques, vers de nouveaux projets peut permettre d'économiser des centaines de milliers de dollars, sans oublier de lancer de nouvelles découvertes potentielles. »

- Introduction : Partie V
- Introduction : Chap 19
- Culture d’apprentissage juste
- Réunions post-mortem
- Publication des post-mortem
- **Réduire la tolérance aux incidents**
- Redéfinir l'échec et encourager la prise de risques calculée
- Injection d’échec de production
- Utilisation de "Game Days"
- Conclusion

Réduire la tolérance aux incidents

- Lorsque nous travaillons au sein de **systèmes complexes, le besoin d'amplifier les signaux de défaillance faibles est essentiel pour éviter les défaillances catastrophiques**
- La manière dont la NASA a géré les signaux de panne à l'ère de la navette spatiale en est un exemple
  - En 2003, après seize jours de mission dans la mission de la navette spatiale Columbia, la navette a explosé lorsqu'elle est revenue dans l'atmosphère terrestre
  - Nous savons maintenant qu'un morceau de mousse isolante s'était détaché du réservoir de carburant externe lors du décollage
  - Cependant, avant le retour de Columbia, une poignée d'ingénieurs de niveau intermédiaire de la NASA avaient signalé l'incident, mais leurs voix n'avaient pas été entendues
  - Ils ont observé la frappe de mousse sur des moniteurs vidéo au cours d'une séance d'examen post-lancement et ont immédiatement averti les responsables de la NASA, mais on leur a dit que le problème de la mousse n'était pas nouveau
  - Le délogement de la mousse avait endommagé les navettes lors des lancements précédents, mais n'avait jamais provoqué d'accident
  - C'était considéré comme un problème de maintenance et il n'a pas été traité avant qu'il ne soit trop tard

- Introduction : Partie V
- Introduction : Chap 19
- Culture d’apprentissage juste
- Réunions post-mortem
- Publication des post-mortem
- Réduire la tolérance aux incidents
- **Redéfinir l'échec et encourager la prise de risques calculée**
- Injection d’échec de production
- Utilisation de "Game Days "
- Conclusion

Redéfinir l'échec et encourager la prise de risques calculée

- Les dirigeants d'une organisation, délibérément ou par inadvertance, renforcent la culture et les valeurs de l'organisation par leurs actions
  - Les experts en audit, comptabilité et éthique observent depuis longtemps que le «ton au sommet» prédit la probabilité de fraude et d'autres pratiques contraires à l'éthique
  - Pour renforcer notre culture d'apprentissage et de prise de risque calculée, nous avons besoin de leaders qui insistent sur le fait que tout le monde doit se sentir à l'aise et responsable de soulever et d'apprendre des échecs
- Au sujet des échecs, Roy Rapoport (Netflix) a déclaré:

  « Ce que le 2014 State of DevOps Report m'a prouvé, c'est que **les organisations très performantes ("high performers") de DevOps échoueront et commettront plus souvent des erreurs.**

  Non seulement c'est correct, c'est ce dont les organisations ont besoin!

  Vous pouvez même le voir dans les données: si les "high performers" exécutent trente fois plus fréquemment, même avec seulement la moitié du taux d'échec, elles ont évidemment plus d'échecs. »

- Introduction : Partie V
- Introduction : Chap 19
- Culture d’apprentissage juste
- Réunions post-mortem
- Publication des post-mortem
- Réduire la tolérance aux incidents
- Redéfinir l'échec et encourager la prise de risques calculée
- **Injection d’échec de production**
- Utilisation de "Game Days"
- Conclusion

Injection d'échec de production

- Comme nous l'avons vu dans l'introduction du chapitre, **l'injection de défauts dans l'environnement de production (tel que Chaos Monkey) est l'un des moyens d'accroître notre résilience**
- Dans cette section, nous décrivons les processus de répétition et d'injection des défaillances dans notre système pour confirmer que nous avons correctement conçu et architecturé nos systèmes, de sorte que les défaillances se produisent de manière spécifique et contrôlée
  - Nous le faisons en effectuant des tests régulièrement (voire en continu) pour nous assurer que nos systèmes échouent normalement
- Michael Nygard (auteur de "Release It! Design and Deploy Production-Ready Software »), commente:

  «Tout comme la création de zones de déformation dans les voitures qui permettent d'absorber les chocs et protéger les passagers,

  vous pouvez choisir les fonctionnalités du système qui sont indispensables et créer des modes de défaillance qui préservent les fissures.

  Si vous ne concevez pas vos modes de défaillance, vous obtiendrez ce qui est imprévisible - et généralement dangereux - qui se présente. »

Injection d’échec de production

- **La résilience exige que nous définissions d'abord nos modes de défaillance, puis que nous effectuions des tests pour nous assurer que ces modes de défaillance fonctionnent comme prévu**
  - Pour ce faire, nous pouvons notamment injecter des erreurs dans notre environnement de production et répéter des erreurs de grande envergure, de sorte que nous puissions être certains que nous pouvons récupérer des accidents quand ils se produisent, idéalement sans que cela affecte nos clients
- L'histoire de 2012 sur Netflix et la panne Amazon AWS-EAST présentée dans l'introduction n'est qu'un exemple

- Introduction : Partie V
- Introduction : Chap 19
- Culture d’apprentissage juste
- Réunions post-mortem
- Publication des post-mortem
- Réduire la tolérance aux incidents
- Redéfinir l'échec et encourager la prise de risques calculée
- Injection d’échec de production
- **Utilisation de "Game Days"**
- Conclusion

Utilisation de "Game Days"

- Dans cette section, nous décrivons un mécanisme de répétition spécifique pour le recouvrement en cas de désastre, appelées **Game Days**
  - Terme popularisé par Jesse Robbins pour le travail qu'il a effectué chez **Amazon**
  - Jesse Robbins
    - Co-fondateurs de la communauté Velocity Conference et co-fondateur de Chef
    - Était responsable des activités suivantes: programmes connus pour assurer la disponibilité du site et était largement connu en interne comme le «maître des catastrophes».
  - Le concept des Game Days est issu de la discipline de l'ingénierie de la résilience
  - Robbins définit l'**ingénierie de la résilience** comme « un **exercice conçu pour augmenter la résilience grâce à une injection de fautes à grande échelle sur des systèmes critiques** »

- Robbins observe que

  « **Chaque fois que vous envisagez de concevoir un système à grande échelle, le mieux que vous puissiez espérer est de créer une plate-forme logicielle fiable au-dessus de composants totalement non fiables**.

  Cela vous place dans un environnement où **les défaillances complexes sont à la fois inévitables et imprévisibles.** »

Utilisation de "Game Days"

- Par conséquent, nous devons veiller à ce que **les services continuent de fonctionner en cas de défaillance**, potentiellement dans l'ensemble de notre système, **idéalement sans crise ni même intervention manuelle**
  - Comme le dit Robbins, "un service n'est pas vraiment testé jusqu'à ce que nous le cassions en production"
- Notre **objectif pour Game Day** est d'**aider les équipes à simuler et à répéter les accidents pour leur permettre de s'exercer**
  - Tout d'abord, nous planifions qu'un événement catastrophique, tel que la destruction simulée d'un centre de données entier, se produise à un moment donné dans l'avenir
  - Nous donnons ensuite aux équipes le temps de se préparer, d'éliminer tous les points uniques de défaillance et de créer les procédures de surveillance, les procédures de basculement, etc. nécessaires

Utilisation de "Game Days"

- Notre équipe Game Day définit et exécute des exercices, tels que l'exécution de défaillance de base de données (simulation d'une défaillance de la base de données et vérification du fonctionnement de la base de données secondaire) ou la désactivation d'une connexion réseau importante pour exposer les problèmes rencontrés dans les processus définis
  - Tous les problèmes ou difficultés rencontrés sont identifiés, résolus et testés à nouveau

- À l'heure programmée, nous exécutons ensuite la panne
  - Comme le décrit Robbins, sur Amazon, ils « éteindraient littéralement une installation - sans préavis», puis laisseraient les systèmes en échec de façon naturelle et [permettraient] aux personnes de suivre leurs processus, où qu'elles se trouvent. »

Utilisation de "Game Days"

- Ce faisant, nous commençons à exposer les défauts latents de notre système, qui sont les problèmes qui apparaissent uniquement du fait de l'injection de défauts dans le système
  - Robbins explique: «Vous découvrirez peut-être que certains systèmes de surveillance ou de gestion essentiels au processus de récupération finissent par être désactivés à la suite de l'échec orchestré. [Ou] vous trouveriez des points d'échec uniques que vous ne connaissiez pas de cette façon. »
  - Ces exercices sont ensuite menés de manière de plus en plus intense et complexe dans le but de leur donner l'impression de faire partie d'une journée ordinaire.
- **En exécutant Game Days**, nous **créons progressivement un service plus résilient** et un **degré de certitude supérieur que nous pouvons reprendre les opérations lorsque des événements inopportuns se produisent**, **tout en créant plus d'apprentissage et une organisation plus résiliente**.

Utilisation de "Game Days"

- Le **programme de recouvrement après sinistre (DiRT)** de Google est un excellent **exemple de simulation de catastrophe**
  - Kripa Krishnan étaitt directeur technique de programmes chez Google et, au moment où le livre a été écrit, dirigeait le programme depuis plus de sept ans
  - Pendant ce temps, ils ont **simulé un tremblement de terre dans la Silicon Valley**, entraînant la déconnexion de l'ensemble du campus de Mountain View de Google; perte totale de puissance pour des centres de données principaux, et même des extraterrestres attaquant des villes où résidaient des ingénieurs
- Comme l'a écrit Krishnan,

  « les processus et les communications sont un domaine de test souvent négligé.

  Les systèmes et les processus sont étroitement liés, et séparer les tests des systèmes des tests des processus d'entreprise n'est pas réaliste:

  une défaillance d'un système d'affaires affectera les processus, et inversement, un système opérationnel n'est pas très utile sans le personnel approprié. »

Utilisation de "Game Days"

- Parmi les enseignements tirés de ces catastrophes, citons:
  - Lorsque la connectivité a été perdue, le basculement vers les postes de travail d'ingénieur n'a pas fonctionné
  - Les ingénieurs ne savaient pas comment accéder à un système de téléconférence (call bridge) ou celui-ci ne pouvait accueillir que 50 personnes, ou ils avaient besoin d'un nouveau fournisseur de téléconférence leur permettant d'éjecter des ingénieurs qui avaient utilisé toute la conférence pour de la musique
  - Lorsque les centres de données ont manqué de diesel pour les groupes électrogènes de secours, personne ne connaissait les procédures d'achat d'urgence auprès du fournisseur, ce qui a poussé quelqu'un à utiliser une carte de crédit personnelle pour acheter 50 000 $ de diesel
- En créant un échec dans une situation contrôlée, nous pouvons pratiquer et créer les documents (playbooks) dont nous avons besoin
- Un des autres résultats de Game Days est que les gens savent réellement qui appeler et à qui parler
  - En créant cela, ils développent des relations avec des personnes d'autres départements afin de pouvoir travailler ensemble lors d'un incident, transformant des actions conscientes en actions inconscientes qui peuvent devenir une routine

- Introduction : Partie V
- Introduction : Chap 19
- Culture d’apprentissage juste
- Réunions post-mortem
- Publication des post-mortem
- Réduire la tolérance aux incidents
- Redéfinir l'échec et encourager la prise de risques calculée
- Injection d’échec de production
- Utilisation de "Game Days"
- **Conclusion**

Conclusion

- **Pour créer une culture juste** qui permette l'apprentissage organisationnel, nous devons **re-contextualiser ce que l'on appelle les échecs**
  - Traitées correctement, les erreurs inhérentes à des systèmes complexes peuvent créer un environnement d'apprentissage dynamique dans lequel tous les actionnaires se sentent suffisamment en sécurité pour émettre des idées et des observations, et où les groupes se remettent plus facilement de projets qui ne se déroulent pas comme prévu
- Les **post-mortem sans reproche** et l' **injection d'échecs de production** renforcent une culture dans laquelle chacun devrait se sentir à l'aise et responsable de faire surface et d'apprendre des échecs
  - En fait, lorsque nous réduisons suffisamment le nombre d'accidents, nous réduisons notre tolérance afin de pouvoir continuer à apprendre
- Comme le dit Peter Senge,

  «Le seul avantage concurrentiel durable est la capacité d'une organisation à apprendre plus vite que ses concurrents. »