

LOG680

Introduction à l'approche DevOps

First Way: Les principes de Flux

DevOps Handbook
Part I, Chap 2



Francis Bordeleau, 2021



Three Ways of DevOps

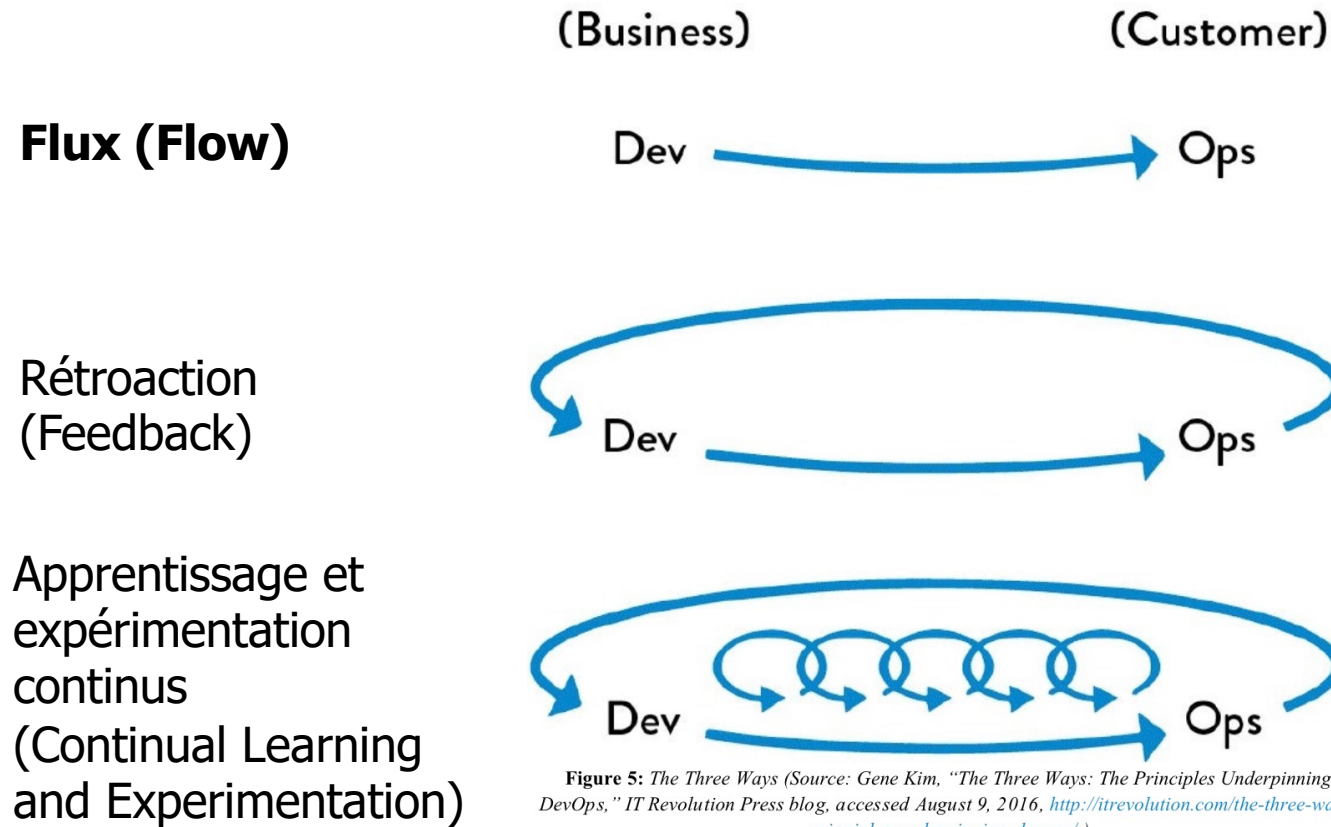


Figure 5: The Three Ways (Source: Gene Kim, "The Three Ways: The Principles Underpinning DevOps," IT Revolution Press blog, accessed August 9, 2016, <http://itrevolution.com/the-three-ways-principles-underpinning-devops/>.)

Objectifs d'apprentissage

- Expliquer en quoi consiste la première voie (First Way)
- Expliquer comment on peut rendre le travail visible
- Expliquer le rôle des tableaux Kanban dans le contexte de DevOps
- Définir le concept de Work in Process (WIP)
- Expliquer comment on peut limiter les travaux en cours (WIP)
- Expliquer l'impact de la taille des lots sur la fluidité et la rapidité du flux
- Expliquer en quoi la réduction du nombre de transferts permet d'augmenter le flux
- Identifier et expliquer cinq (5) types de pertes/gaspillages (waste) dans le développement logiciel.

Sujets

- Introduction
- Rendre le travail visible
- Limite des travaux en cours (WIP)
- Réduire la taille des lots (batch sizes)
- Réduire le nombre de transferts
- Identifier et éliminer les contraintes
- Éliminer les difficultés et les pertes
- Conclusion

- **Introduction**
- Rendre le travail visible
- Limite des travaux en cours (WIP)
- Réduire la taille des lots (batch sizes)
- Réduire le nombre de transferts
- Identifier et éliminer les contraintes
- Éliminer les difficultés et les pertes
- Conclusion

First Way: Les principes de Flux



Introduction

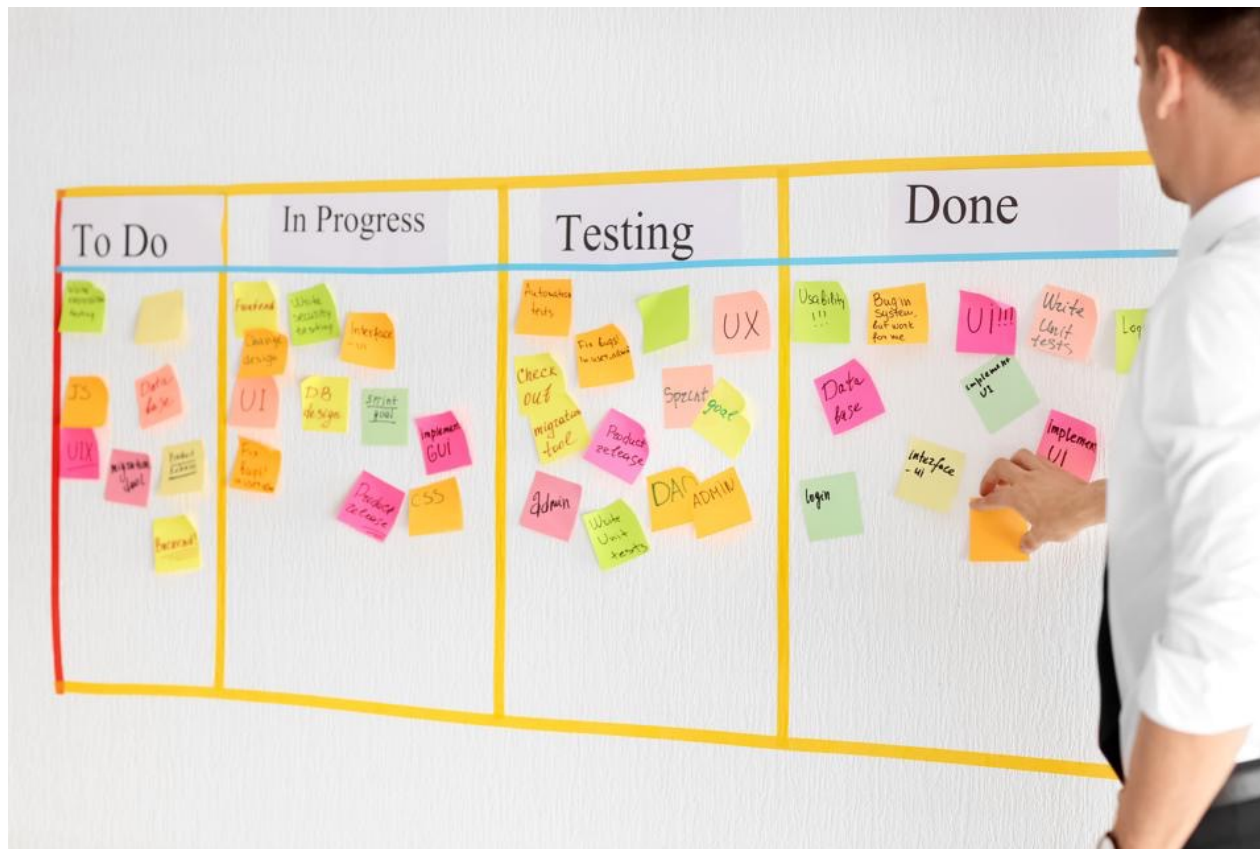
- Flux de valeur technologique : Les travaux vont du développement aux opérations
- First Way : Établir un flux de travail rapide et fluide, du développement aux opérations, pour apporter rapidement de la valeur aux clients.
 - Se concentrer sur un **objectif global** plutôt que pour des objectifs locaux
- Comment augmenter le débit
 - **Rendre le travail visible**
 - **Réduire la taille des lots** et **les intervalles** de travail
 - **Intégrer la qualité**, empêchant le transfert des défauts aux centres de travail en aval
- Pourquoi?
 - **Réduire le délai requis pour répondre aux demandes des clients** internes et externes
 - **Augmenter la qualité de notre travail**
 - **Rendre les organisations plus agiles** et **capables de surpasser la concurrence.**
- Objectif:
 - **Réduire le temps nécessaire au déploiement des modifications en production**
 - **Accroître la fiabilité et la qualité de ces services.**

- Introduction
- **Rendre le travail visible**
- Limite des travaux en cours (WIP)
- Réduire la taille des lots (batch sizes)
- Réduire le nombre de transferts
- Identifier et éliminer les contraintes
- Éliminer les difficultés et les pertes
- Conclusion

Rendre le travail visible

- Différence entre flux de valeur technologique et flux de valeur de fabrication: **visibilité du travail**
 - Domaine de la fabrication: le transfert entre les postes de travail est généralement très visible et lent, car les stocks doivent être déplacés physiquement.
 - Domaine technologique, le déplacement peut être effectué en un clic, par exemple en réaffectant un ticket de travail à une autre équipe
- Problématique
 - Contrairement aux processus physiques, nous ne pouvons pas facilement voir où le flux est entravé ou lorsque le travail s'accumule devant des stations de travail soumis à des contraintes.
 - Le travail peut basculer d'une équipe à l'autre en tout en contenant des **informations incomplètes** ou des **problèmes invisibles** qui **résultent dans des retards de livraison ou des problèmes de déploiement** (sur des environnements de production)
- Solution:
 - **Utiliser des tableaux de travail visuels**, e.g. tableau kanban ou tableau de planification de sprint

Kanban



Questions

- Quels sont les bénéfices des tableaux Kanban?
- Comment peut-on utiliser les tableaux Kanban pour mesurer notre de flux de valeur?
- Peut-on utiliser les tableaux Kanban pour mesurer le délai d'exécution ("lead time") et le temps de traitement ("processing time")?
- Quelles sont les limites de tableaux Kanban?

Bénéfices de tableau Kanban

- Permet de
 - **Gérer notre travail** afin de permettre la transition de gauche à droite le plus rapidement possible
 - **Mesurer le délai** entre le moment où une carte est placée sur le tableau et son déplacement dans la colonne "Terminé".
- Idéalement, notre tableau kanban couvrira l'ensemble de la chaîne de valeur
 - Définit le travail comme terminé ("Done") uniquement lorsqu'il atteint le côté droit du tableau
 - Le travail ne se termine pas avec la fin du développement d'une fonctionnalité
 - Il se termine uniquement lorsque notre application est déployée (en production) et apporte de la valeur au client.
- Permet à toutes les parties prenantes de plus facilement visualiser le travail dans le contexte d'objectifs globaux.
- Permet à chaque poste de travail de se concentrer sur les tâches les plus prioritaires jusqu'à la fin, augmentant ainsi le débit.

- Introduction
- Rendre le travail visible
- **Limite des travaux en cours (WIP)**
- Réduire la taille des lots (batch sizes)
- Réduire le nombre de transferts
- Identifier et éliminer les contraintes
- Éliminer les difficultés et les pertes
- Conclusion

Limite des travaux en cours (WIP)

- Fabrication,
 - Le travail quotidien est généralement **dicté par un programme de production**
 - Généré régulièrement, par exemple, chaque jour, chaque semaine
 - Détermine les travaux à exécuter en fonction des commandes des clients, des dates d'échéance des commandes, des pièces disponibles, etc.
- Technologie,
 - **Le travail est généralement beaucoup plus dynamique**
 - Par exemple, dans les services partagés, les équipes doivent satisfaire les demandes de nombreux clients
 - **Le travail quotidien est dominé par la priorité du jour**
 - Souvent des demandes de travail urgent
 - E.g. "tickets", appels en cas de panne, courriels, appels téléphoniques, salles de discussion et escalades de gestion

Interruptions des travaux en cours

- Dans la fabrication,
 - Les perturbations sont très visibles et coûteuses.
 - Elles nécessitent souvent d'interrompre le travail en cours et de supprimer tout travail incomplet (en cours) pour commencer une nouvelle tâche.
 - Ce niveau d'effort élevé décourage les perturbations fréquentes.
- En technologie,
 - Facile d'interrompre les travailleurs
 - Les conséquences sont invisibles pour presque tout le monde, même si l'impact négatif sur la productivité peut être beaucoup plus important que dans l'industrie manufacturière
 - E.g. un ingénieur affecté à plusieurs projets doit basculer d'une tâche à l'autre, ce qui entraîne tous les coûts du rétablissement du contexte, ainsi que des règles et objectifs cognitifs.
- Plusieurs études ont montré que le temps nécessaire pour effectuer des tâches même simples, telles que le tri de formes géométriques, se dégrade de manière significative en mode multitâche.

Limite des travaux en cours (WIP) – Solution

- Nous pouvons limiter le multitâche lorsque nous utilisons un tableau kanban pour gérer notre travail
 - Par exemple en codifiant et en appliquant des limites WIP (travaux en cours) pour chaque colonne ou centre de travail qui limite le nombre de cartes pouvant être contenues dans une colonne, e.g. limite WIP de trois cartes pour les tests.
 - Rien ne peut être fait tant qu'il n'a pas été représenté en premier sur une carte de travail, ce qui renforce le fait que tout travail doit être rendu visible.
- Dominica DeGrandis (expert d'utilisation de tableaux kanban dans les flux de valeur DevOps,
"le contrôle de la taille de la file d'attente [WIP] est un outil de gestion extrêmement puissant, car il s'agit de l'un des rares indicateurs avancés du délai d'exécution - avec la plupart des tâches, nous ne savons pas combien de temps cela prendra avant que ce soit réellement terminé."

Limite des travaux en cours (WIP)



<https://medium.com/hiredscore-engineering/team-initiatives-stop-starting-and-start-finishing-261b7d64e859>

Arrêtez de commencer. Commencez à finir!

- Introduction
- Rendre le travail visible
- Limite des travaux en cours (WIP)
- **Réduire la taille des lots (batch sizes)**
- Réduire le nombre de transferts
- Identifier et éliminer les contraintes
- Éliminer les difficultés et les pertes
- Conclusion

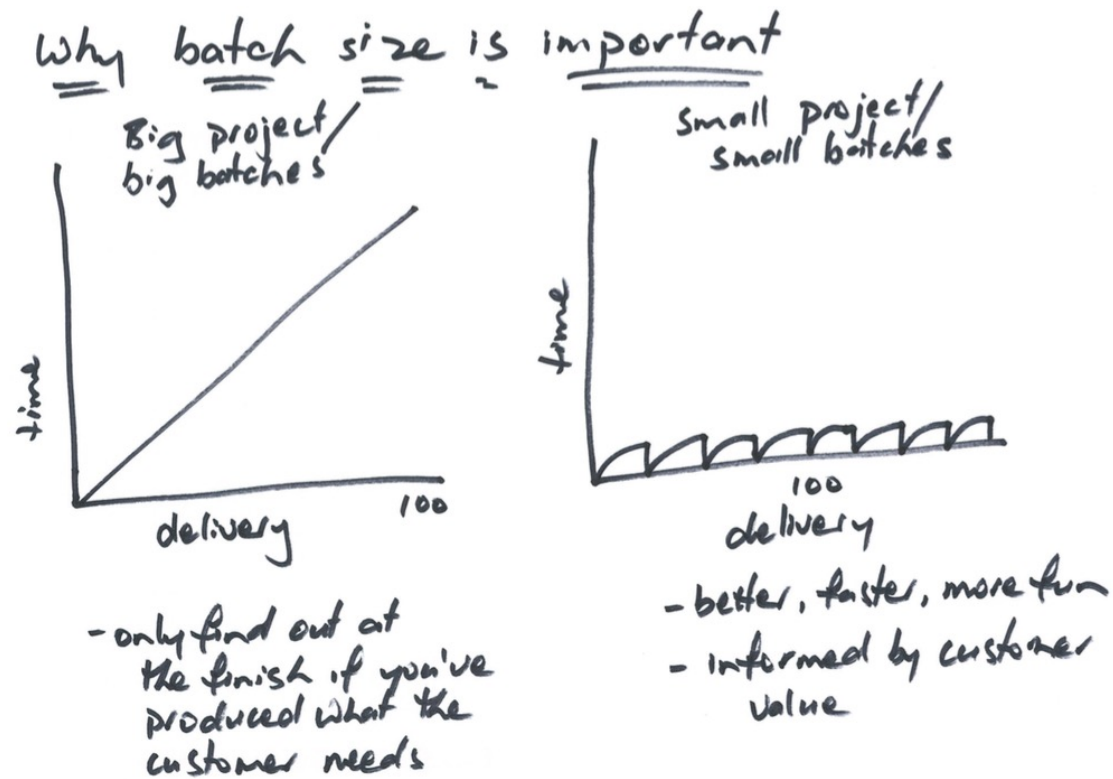
Questions

- Pourquoi est-il important de réduire la taille des lots de travail?

Réduire la taille des lots (batch sizes)

- Un autre élément clé pour créer un flux fluide et rapide consiste à **effectuer des travaux dans des lots de petite taille**.
 - Avant la révolution de la fabrication "Lean", il était courant de fabriquer des lots de grande taille, en particulier pour les opérations où la configuration ou le changement de travail nécessitait beaucoup de temps ou d'argent.
 - Lorsque le coût du changement est si élevé, nous estamperions autant de panneaux à la fois que possible, créant ainsi des lots volumineux afin de réduire le nombre de changements.
- Cependant, **les lots de grande taille entraînent une montée en flèche du nombre de travaux en cours** et une **grande variabilité des flux** qui se répercutent dans l'ensemble de l'usine de fabrication.
 - Il en résulte des délais longs et une qualité médiocre: si un problème est détecté dans un panneau de carrosserie, le lot entier doit être mis au rebut.
- L'une des principales leçons de Lean est que, **pour réduire les délais et accroître la qualité**, nous devons nous efforcer de **réduire en permanence la taille des lots**.
 - La limite inférieure théorique de la taille du lot est le flux monobloc, où chaque opération est effectuée unité par unité (batch size one).

Exemple de réduction de la taille des lots



<https://www.boost.co.nz/blog/2016/06/how-reducing-your-batch-size-radically-reduces-your-costs>

Exemple de réduction de la taille des lots

Large Batches



Single-Piece Flow



Figure 7: Simulation of “envelope game” (fold, insert, seal, and stamp the envelope)

(Source: Stefan Luyten, “Single Piece Flow: Why mass production isn’t the most efficient way of doing ‘stuff’,” Medium.com, August 8, 2014, <https://medium.com/@stefanluyten/single-piece-flow-5d2c2bec845b#.9o7sn74ns>.)

Taille des lots (batch sizes)

- Eric Ries – article sur les leçons apprises au démarrage, déclare:
«La taille du lot est l'unité dans laquelle les produits de travail se déplacent entre les étapes d'un processus de développement [ou de DevOps].
Pour les logiciels, le lot le plus facile à voir est le code.
Chaque fois qu'un ingénieur enregistre le code, il effectue une certaine quantité de travail.
Il existe de nombreuses techniques pour contrôler ces lots, allant des très petits lots requis pour un déploiement continu au développement plus traditionnel basé sur les branches, où tout le code provenant de plusieurs développeurs travaillant pendant des semaines ou des mois est mis en lots et intégré ensemble. ”
- L'équivalent d'un "single-piece flow" de la chaîne de valeur technologique est réalisé avec un déploiement continu, où chaque modification engagée pour le contrôle de version est intégrée, testée et déployée en production.
- Les pratiques qui permettent cela sont décrites dans la partie IV.

- Introduction
- Rendre le travail visible
- Limite des travaux en cours (WIP)
- Réduire la taille des lots (batch sizes)
- **Réduire le nombre de transferts**
- Identifier et éliminer les contraintes
- Éliminer les difficultés et les pertes
- Conclusion

Réduire le nombre de transferts

CS312821



"Failure is not an option. But inaction due to foot-dragging and red tape is a definite possibility."

https://www.cartoonstock.com/directory/p/project_delay.asp

Réduire le nombre de transferts – Contexte

- Chaîne de valeur technologique, chaque fois que nous avons de longs délais de déploiement, mesurés en mois, c'est souvent parce que des centaines (voire des milliers) d'opérations sont nécessaires pour transférer notre code du contrôle de version à l'environnement de production.
- Pour transmettre du code via le flux de valeur, plusieurs services doivent travailler sur diverses tâches, notamment les tests fonctionnels, les tests d'intégration, la création d'environnement, l'administration de serveurs, l'administration de stockage, la mise en réseau, l'équilibrage de charge et la sécurité des informations.

Réduire le nombre de transferts – Problème

- Chaque fois que le travail passe d'une équipe à une autre, nous avons besoin de toutes sortes de communications: demander, spécifier, signaler, coordonner et souvent hiérarchiser, planifier, résoudre les conflits, tester et vérifier.
 - Cela peut nécessiter l'utilisation de différents systèmes de "tickets" ou de gestion de projet; rédiger des documents de spécifications techniques; communiquer via des réunions, des courriels ou des appels téléphoniques; et en utilisant des partages de système de fichiers, des serveurs FTP et des pages Wiki.
- Même dans les meilleures circonstances, **certaines connaissances sont inévitablement perdues à chaque transfert.**
- Avec suffisamment de transferts, le travail peut complètement perdre le contexte du problème à résoudre ou de l'objectif organisationnel pris en charge.
 - Par exemple, un administrateur de serveur peut voir un ticket nouvellement créé demandant la création de comptes d'utilisateur, sans savoir pour quelle application ou service, il doit être créé, quelles en sont les dépendances ou s'il s'agit d'un travail récurrent.

Réduire le nombre de transferts – Exemple

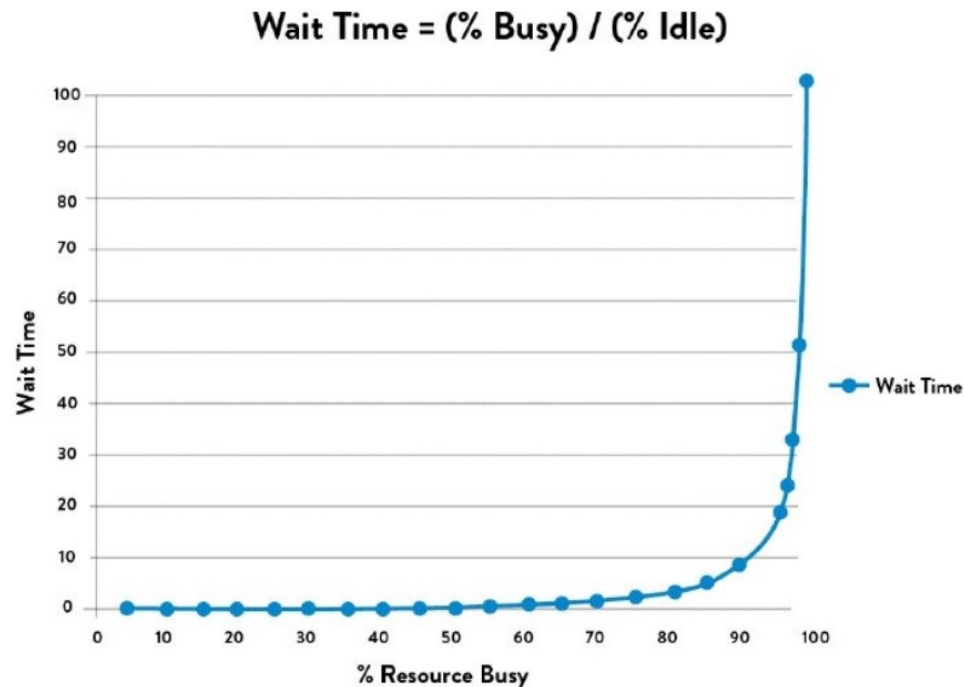


Figure 47: Queue size and wait times as function of percent utilization (Source: Kim, Behr, and Spafford, *The Phoenix Project*, ePub edition, 557.)

Réduire le nombre de transferts – Solution

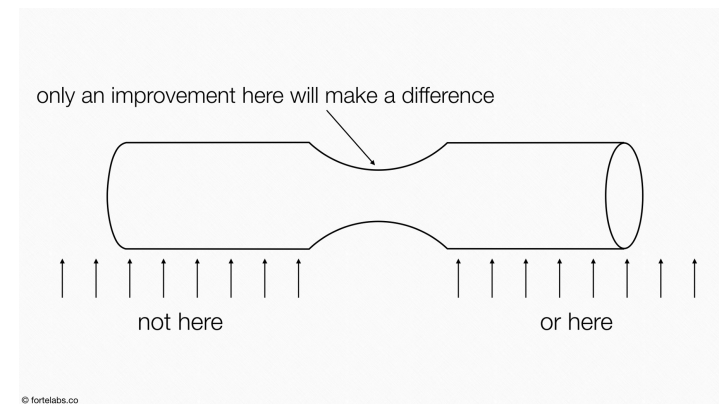
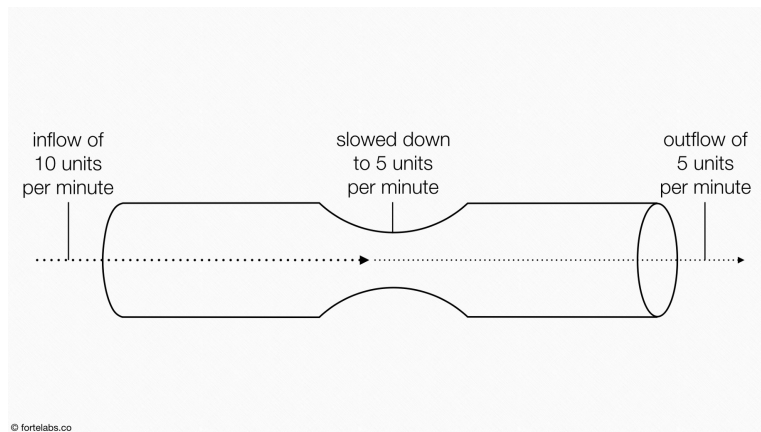
- Pour atténuer ce type de problèmes, nous nous efforçons de réduire le nombre de transferts, en automatisant des parties importantes du travail ou en réorganisant les équipes afin qu'elles puissent offrir de la valeur au client, au lieu d'être constamment dépendantes des autres.
- En conséquence, nous augmentons le flux en réduisant le temps passé en attente par notre travail, ainsi que le temps sans valeur ajoutée.

- Introduction
- Rendre le travail visible
- Limite des travaux en cours (WIP)
- Réduire la taille des lots (batch sizes)
- Réduire le nombre de transferts
- **Identifier et éliminer les contraintes**
- Éliminer les difficultés et les pertes
- Conclusion

Identifier et éliminer les contraintes

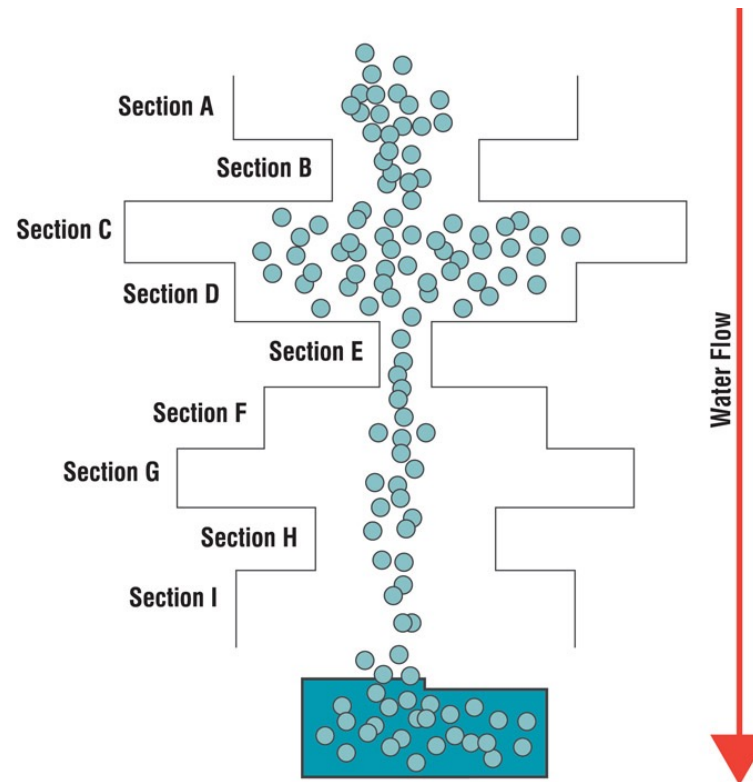
«**Dans tout flux de valeur**, il existe toujours une direction de flux et **il y a toujours une seule et unique contrainte**; toute amélioration non réalisée avec cette contrainte est une illusion. »

Dr. Goldratt. Beyond the Goal.



<https://www.smallworldsocial.com/theory-of-constraints-101-introduction-to-the-series/>

Exemple de contrainte



<https://www.thefabricator.com/thefabricator/article/shopmanagement/scheduling-by-constraint-in-metal-fabrication>

Identifier et éliminer les contraintes – Solution

- Dr Goldratt a défini les «**cinq étapes principales**»:
 - **Identifiez** la contrainte du système
 - **Exploitez** la contrainte du système
 - **Subordonner** tout le reste aux décisions ci-dessus
 - **Élever** la contrainte du système
 - **Recommencer**



<https://www.smartsheet.com/all-about-theory-of-constraints>

Identifier et éliminer les contraintes – DevOps

- Dans les transformations DevOps typiques – lorsque vous passez de délais de déploiement mesurés en mois ou trimestres à des délais mesurés en minutes, la contrainte suit généralement la progression suivante :
 - Création d'environnement
 - Déploiement de code
 - Configuration et exécution des tests
 - Architecture trop serrée

Identifier et éliminer les contraintes – DevOps

- Après que toutes ces contraintes aient été éliminées, **notre contrainte principale sera probablement la gestion de produits.**
 - Parce que notre objectif est de permettre aux petites équipes de développeurs de développer, tester et déployer de la valeur ajoutée rapidement et de manière fiable pour les clients, c'est là que nous voulons que notre contrainte soit.
 - Dans les entreprises les plus performantes, qu'il s'agisse d'un ingénieur en développement, en assurance qualité, en opérations ou en infosec, tous déclarent que leur objectif est d'aider à optimiser la productivité des développeurs.
- Lorsque cet état est atteint, nous ne sommes **limités que par le nombre d'hypothèses de bonnes affaires que nous créons** et par notre **capacité à développer le code nécessaire pour tester ces hypothèses auprès de clients réels.**
- La progression des contraintes répertoriées ci-dessus sont des généralisations de transformations typiques.
- Des techniques permettant d'identifier la contrainte dans les flux de valeurs réelles, telles que la cartographie et les mesures de flux de valeurs (value stream mapping and measurements), seront discutées plus tard dans le cours.

- Introduction
- Rendre le travail visible
- Limite des travaux en cours (WIP)
- Réduire la taille des lots (batch sizes)
- Réduire le nombre de transferts
- Identifier et éliminer les contraintes
- **Éliminer les difficultés et les pertes**
- Conclusion

Questions

- Identifiez les pertes dans le développement logiciel?

Éliminer les difficultés et les pertes

- Shigeo Shingo (pionnier du système de production Toyota) a estimé que **les pertes ("waste") constituent la plus grande menace pour la viabilité de l'entreprise.**
- La définition couramment utilisée dans Lean est **"l'utilisation de tout matériau ou ressource au-delà de ce que le client exige et est prêt à payer."**
- Il a défini **sept grands types de pertes** de fabrication
 - **Inventaires/stocks, surproduction, traitement supplémentaire, transport, attente, mouvement, et défauts.**
- Interprétations plus modernes de Lean – l'élimination du gaspillage peut avoir un impact dégradant et déshumanisant; au lieu de cela, l'**objectif** est recadré pour **réduire les difficultés et les tâches fastidieuses de notre travail quotidien grâce à un apprentissage continu afin d'atteindre les objectifs de organisationnels.**
- Dans le livre, le terme pertes/gaspillage impliquera cette définition plus moderne, car elle correspond plus étroitement aux idéaux de DevOps et aux résultats souhaités.
- Mary et Tom Poppendieck ("Implementing Lean Software Development: From Concept to Cash ») décrivent le gaspillage et les difficultés du développement logiciel comme un **élément susceptible de causer des retards au client**, telles que des activités pouvant être contournées sans affecter le résultat.

Éliminer les difficultés et les pertes

- Pertes/gaspillages
 - Travail partiellement terminé
 - Processus supplémentaires
 - Fonctionnalités supplémentaires
 - Changement de tâche
 - En attente
 - Mouvement
 - Défauts
 - Travail non standard ou manuel
 - Héroïque

- Introduction
- Rendre le travail visible
- Limite des travaux en cours (WIP)
- Réduire la taille des lots (batch sizes)
- Réduire le nombre de transferts
- Identifier et éliminer les contraintes
- Éliminer les difficultés et les pertes
- **Conclusion**

Conclusion

- L'amélioration de la circulation dans le flux de valeur technologique est essentielle pour atteindre les résultats de DevOps.
- Nous le faisons en
 - **rendant le travail visible**
 - **limitant les travaux en cours**
 - **réduisant la taille des lots** et le **nombre de transferts**
 - **identifiant et en évaluant en permanence nos contraintes** et en **éliminant les difficultés de notre travail quotidien.**
- Les pratiques spécifiques permettant un flux rapide dans le flux de valeurs DevOps sont présentées dans la partie IV.