

Avec les compliments de **IBM**

Edition limitée IBM

La virtualisation des services

POUR
LES NULS[®]

Vous apprendrez :

- Ce qu'est la virtualisation des services
- Comment choisir les services les plus avantageux à virtualiser
- Comment améliorer votre approche des tests
- Comment produire plus rapidement des logiciels de meilleure qualité

**Marcia Kaufman
Judith Hurwitz**



Pour en savoir encore plus sur la virtualisation des services et sur la manière dont IBM peut vous aider à produire plus rapidement des logiciels de meilleure qualité, consultez le site <http://ibm.co/servicevirtualization>

La virtualisation des services

POUR
LES NULS[®]

EDITION LIMITÉE IBM

**par Marcia Kaufman
et Judith Hurwitz**

WILEY

La virtualisation des services pour les Nuls®, édition limitée IBM

Publié par
John Wiley & Sons, Inc.
111 River St.
Hoboken, NJ 07030-5774
www.wiley.com

Copyright © 2013 par John Wiley & Sons, Inc., Hoboken, New Jersey

Publié par John Wiley & Sons, Inc., Hoboken, New Jersey

Aucun extrait de cette publication ne peut être reproduit, stocké dans une base de données ni transmis, sous quelque forme ou par quelque moyen que ce soit (électronique, mécanique, photocopie, enregistrement, numérisation ou autre), sauf aux conditions autorisées aux alinéas 107 et 108 du United States Copyright Act de 1976, en l'absence d'autorisation écrite préalable de l'Éditeur. Les demandes d'autorisation doivent être adressées par courrier à Permissions Department, John Wiley & Sons, Inc., 111 River Street, Hoboken, NJ 07030, par téléphone au (201) 748-6011, par télécopie au (201) 748-6008 ou en ligne sur <http://www.wiley.com/go/permissions>.

Marques : Wiley, le logo Wiley, pour les Nuls, le logo Dummies Man, A Reference for the Rest of Us!, The Dummies Way, Dummies.com, Making Everything Easier et les appellations commerciales afférentes sont des marques commerciales ou déposées de John Wiley & Sons, Inc. et/ou de ses sociétés affiliées aux États-Unis et dans d'autres pays, dont l'utilisation est interdite en l'absence d'autorisation écrite. IBM et le logo IBM sont des marques déposées d'IBM. Toutes les autres marques citées sont la propriété de leurs détenteurs respectifs. John Wiley & Sons Publishing, Inc. n'est lié à aucun des produits ou fournisseurs cités dans cet ouvrage.

LIMITATION DE RESPONSABILITE/DENI DE GARANTIE : L'ÉDITEUR ET L'AUTEUR S'ABSTIENNENT DE TOUTE DÉCLARATION OU GARANTIE S'AGISSANT DE L'EXACTITUDE OU DE L'EXHAUSTIVITÉ DU CONTENU DE CET OUVRAGE, ET REJETENT EN PARTICULIER TOUTE GARANTIE, Y COMPRIS, NON LIMITATIVEMENT, TOUTE GARANTIE D'APTITUDE À UN USAGE PARTICULIER. AUCUNE GARANTIE NE PEUT ÊTRE CONSENTIE OU ÉTENDUE AU TITRE D'UN DOCUMENT COMMERCIAL OU PROMOTIONNEL. LES CONSEILS ET STRATÉGIES PRÉSENTES ICI RISQUENT DE NE PAS CONVENIR À TOUTES LES SITUATIONS. CET OUVRAGE EST COMMERCIALISÉ, SACHANT QUE L'ÉDITEUR NE DISPENSE AUCUN SERVICE JURIDIQUE, COMPTABLE OU AUTRES SERVICES PROFESSIONNELS. SI UNE ASSISTANCE PROFESSIONNELLE EST REQUISE, LES SERVICES D'UN PROFESSIONNEL COMPÉTENT DEVRONT ÊTRE SOLLICITÉS. NI L'ÉDITEUR, NI L'AUTEUR NE POURRONT ÊTRE TENUS RESPONSABLES DES DOMMAGES DECOULANT DES PRÉSENTES. SI UN ÉTABLISSEMENT OU SITE WEB EST RÉFÉRENCÉ DANS UNE CITATION ET/OU COMME SOURCE POTENTIELLE D'INFORMATIONS COMPLÉMENTAIRES DANS CET OUVRAGE, CELA NE SIGNIFIE AUCUNEMENT QUE L'AUTEUR OU L'ÉDITEUR AVAISE LES INFORMATIONS SUSCEPTIBLES D'ÊTRE COMMUNIQUÉES PAR CET ÉTABLISSEMENT OU CE SITE WEB OU SES RECOMMANDATIONS. PAR AILLEURS, LE LECTEUR DOIT AVOIR CONSCIENCE QUE LES SITES WEB CITÉS DANS CET OUVRAGE PEUVENT AVOIR ÉVOLUÉ OU DISPARU ENTRE LE MOMENT OU CE LIVRE A ÉTÉ ÉCRIT ET CELUI OU IL EST LU.

Pour toute information d'ordre général sur nos autres produits et services, ou pour obtenir tous les détails utiles sur la création d'un ouvrage pour les Nuls personnalisé adapté à votre entreprise, veuillez contacter notre service de développement commercial aux États-Unis par téléphone au 877-409-4177, envoyer un e-mail à info@dummies.biz ou consulter le site www.wiley.com/go/custompub. Pour de plus amples informations sur l'exploitation sous licence de la marque pour les Nuls avec des produits ou services, contactez BrandedRights&Licenses@Wiley.com.

ISBN 978-1-118-87240-6 (pbk); ISBN 978-1-118-87323-6 (ebk)

Fabriqué aux États-Unis

10987654321

Sommaire

Introduction 1

A propos de cet ouvrage.....	1
Les icônes utilisées dans ce livre.....	2

Chapitre 1 : Qu'est-ce que la virtualisation des services ? . . . 3

Définition de la virtualisation des services.....	4
La virtualisation des services en action.....	5
Exemple : Whiz Bang International.....	5
Exemple : grande société de services financiers.....	7
Différences entre la virtualisation des services et d'autres types de virtualisation.....	8
Valeur ajoutée de la virtualisation des services.....	9
Réalisation des tests.....	10
Développement.....	11
Utilisation hors production.....	11
Avantages de la virtualisation des services.....	12
Réduction des coûts.....	13
Accroissement de la productivité.....	13
Réduction des risques.....	14
Amélioration de la qualité.....	14

Chapitre 2 : Les moteurs du changement 15

Répondre aux attentes croissantes en matière d'applications d'entreprise.....	16
Adoption d'architectures orientées services.....	17
Essor des applications pour mobile.....	18
Poursuite d'une transformation agile.....	19

Chapitre 3 : Evolution vers l'avenir 21

Améliorer la qualité au cours du cycle de vie applicative.....	21
Repenser l'automatisation des tests.....	23
Faire face aux défis d'environnements de test complexes.....	25
Virtualisation des services et environnements de test complexes.....	26

Chapitre 4 : Comment évoluer vers la virtualisation des services. 27

Identification des services à virtualiser.....	27
Analyse coûts-avantages.....	29

Volatilité des services	29
Impact de l'indisponibilité	30
Coût de déploiement ou d'utilisation	30
Complexité de la technologie.....	31
Calculs.....	31
Stratégies d'automatisation des tests.....	33
Mise en œuvre de la virtualisation des services	
par objectif et phase de test	34
Phases de test.....	34
Tests de performance.....	36
Tests négatifs	38

Chapitre 5 : Mise en œuvre de la virtualisation des services 39

Comprendre votre architecture	39
Communication entre les composants.....	41
Normes relatives aux messages	43
Recherche des points finaux (endpoints).....	43
Définition de composants virtuels.....	44
Synchronisation avec des sources externes	44
Enregistrement de services existants	44
Rééchantillonnage du comportement des composants virtuels	45
Comprendre le mécanisme d'enregistrement	46
Comportement des composants virtuels	46
Provisionnement des services virtuels	49

Chapitre 6 : Mesure du retour sur investissement. 51

Elaboration d'une étude de rentabilité.....	51
Pourquoi choisir la virtualisation des services ?.....	52
Estimation des coûts de mise en œuvre de la virtualisation des services	53
Estimation des avantages de la mise en œuvre de la virtualisation des services	53
Quantification des avantages	53
Elimination ou réduction des coûts liés aux environnements de test classiques	54
Durée de provisionnement des environnements de test...55	
Recherche et résolution des défauts en amont du processus de développement.....	55
Accélération de la mise sur le marché.....	56
Amélioration des processus.....	57
Choix d'une solution.....	57



Chapitre 7 : Dix points clés pour réussir avec la virtualisation des services	59
Repenser votre approche des tests	59
Planifier pour la flexibilité.....	60
Pratiquer l'intégration contrôlée	60
Tester en permanence, du développement à la production.....	61
Externaliser les données de test	61
Explorer des scénarios de test évolués.....	62
Eviter de réinventer la roue.....	62
La virtualisation des services	
ne s'adresse pas qu'aux testeurs.....	63
Partager les composants virtuels au sein de l'entreprise	63
Accroissement de la productivité	
des équipes en développant les compétences	64

Remerciements de l'éditeur

Nous sommes fiers de cet ouvrage et des personnes qui ont participé à sa réalisation. Pour obtenir des détails utiles sur la création d'un ouvrage « pour les Nuls » personnalisé adapté à votre entreprise, contactez info@dummies.biz ou consultez le site www.wiley.com/go/custompub. Pour de plus amples informations sur l'exploitation sous licence de la marque pour les Nuls avec des produits ou services, contactez BrandedRights&Licenses@Wiley.com.

Voici une liste non exhaustive des personnes qui ont contribué à la publication de cet ouvrage :

Acquisitions, édition et développement des médias

Rédactrice en chef du projet :

Carrie A. Burchfield

Responsable de l'édition : Rev Mengle

Chargée du développement commercial :

Sue Blessing

Chargé du projet d'édition personnalisée :

Michael Sullivan

Services de rédaction

Coordonnatrice principale du projet :

Kristie Rees

Publication et éditorial pour les débutants en technologies

Richard Swadley, vice-président et éditeur en chef du groupe

Andy Cummings, vice-président et éditeur

Mary Bednarek, directrice générale, acquisitions

Mary C. Corder, rédactrice en chef

Publication et éditorial pour les débutants en produits de consommation

Kathleen Nebenhaus, vice-présidente et éditrice en chef

Services de rédaction

Debbie Stailey, directrice des services de rédaction

Développement commercial

Lisa Coleman, directrice, développement de la marque et de nouveaux marchés

Remerciements des auteurs

Sincères remerciements à nos amis des équipes IBM, Cheri Bergeron, Carole Gibbins, Peter Klenk et Al Wagner, sans lesquels cet ouvrage n'aurait pas vu le jour.

Introduction

Bienvenue dans *La virtualisation des services pour les Nuls*, une édition limitée IBM. Grâce à la virtualisation des services, les entreprises peuvent créer des environnements de test plus efficaces en éliminant un grand nombre de goulets d'étranglement qui sclérosent généralement ce type d'environnement. Freinées par la complexité croissante des environnements logiciels, les équipes de test ont toutes les difficultés du monde à effectuer des essais à un stade précoce du processus de développement d'applications. Afin de limiter les risques liés aux projets et d'obtenir de meilleurs résultats, votre entreprise a besoin d'une nouvelle approche proactive des tests. Vous avez besoin d'une approche qui améliore le niveau global de test et augmente l'efficacité de résolution des défauts.

La virtualisation des services s'adresse aux entreprises dont les équipes développent et produisent des applications complexes basées sur plusieurs composants interdépendants qui doivent être testés. Grâce à la virtualisation des services, nul besoin d'attendre que chaque service soit prêt à être testé. Il suffit aux équipes d'émuler les éléments manquants. En basant les environnements de test sur des services virtuels, et non sur des services de production, la virtualisation des services permet d'accroître la fréquence des tests d'intégration. En conséquence, le déploiement de la virtualisation des services peut vous aider à réduire les coûts des tests, accroît la productivité des équipes et enfin, améliore la qualité des logiciels.

A propos de cet ouvrage

Cet ouvrage présente l'exploitation de la virtualisation des services au sein d'environnements de test. Grâce à la simulation des composants de service, les entreprises ont la possibilité de valider rapidement le comportement et les performances des composants d'une application, et de déterminer la manière dont ils interagissent. Cet ouvrage présente les défis majeurs liés au développement d'applications complexes associées à plusieurs dépendances, et explique comment rendre les équipes de test plus

2 La virtualisation des services pour les Nuls, Edition Limitée IBM

efficaces grâce à la virtualisation des services, afin de mener des tests plus sophistiqués et précis à un stade plus précoce du cycle de vie.

Les icônes utilisées dans ce livre



Ce livre utilise les icônes suivantes pour attirer votre attention sur les informations que vous pouvez trouver utiles.

Cette icône signale les points nécessitant une attention particulière.



Les paragraphes signalés par cette icône attirent votre attention sur les pièges les plus courants dans lesquels vous risquez de tomber.



Cette icône vous signale les concepts importants à retenir.



Cette icône signale les endroits où des sujets techniques sont évoqués en détail.

Chapitre 1

Qu'est-ce que la virtualisation des services ?

Dans ce chapitre

- ▶ Virtualisation des services : présentation
- ▶ La virtualisation des services en action
- ▶ Différences entre la virtualisation des services et d'autres types de virtualisation
- ▶ Valeur ajoutée de la virtualisation des services
- ▶ Avantages de la virtualisation des services

Imaginez un monde dans lequel les équipes de développement logiciel livreraient de nouvelles applications systématiquement dans les délais, sans dépasser le budget et avec une qualité et des performances exceptionnelles. Face aux exigences des applications complexes d'aujourd'hui en termes de test, de nombreuses équipes opérationnelles et de développement ne parviennent pas à concrétiser ces objectifs au sein de leur propre environnement.

Ce chapitre présente aux communautés de développement et de test la nouvelle technologie de *virtualisation des services*. Nous verrons comment certaines entreprises intègrent cette technologie parmi les composants clés de leur stratégie de test pour réduire les risques, limiter les coûts et produire des logiciels de meilleure qualité. Face aux systèmes actuels à la fois complexes et interdépendants, la virtualisation des services résout de nombreuses problématiques en matière de test. Le terme *virtualisation* étant communément employé au sein de différents cercles, nous verrons la différence entre la virtualisation des services et d'autres types de virtualisation. Nous examinerons également dans quels cas la virtualisation des services génère le plus de valeur ajoutée, et examinerons ses divers usages et principaux avantages.

Définition de la virtualisation des services

La *virtualisation des services* simule le comportement de certains composants d'une application, afin que celle-ci puisse être testée dans son ensemble, de bout en bout. En basant les environnements de test sur des services virtuels au lieu de services de production, les équipes ont la possibilité de réaliser des tests d'intégration à un stade plus précoce du processus de développement. La virtualisation des services s'adresse à tout personnel impliqué dans le développement et la production d'applications logicielles. Il n'est pas rare que l'indisponibilité de certains composants associés à une application testée retarde les tests d'intégration. En émulant les dépendances des composants indisponibles, la virtualisation des services permet d'effectuer des tests d'intégration plus fréquents et plus précoces.



Les solutions de virtualisation des services présentent les caractéristiques suivantes :

- **Emulation applicative** : des composants virtuels simulent le comportement d'une application ou de l'un de ses composants.
- **Environnements de test multiples** : les développeurs et les professionnels de la qualité ont la possibilité de créer des environnements de test à partir de composants virtuels configurés selon leurs besoins.
- **Utilisation des outils de test habituels** : les développeurs et les professionnels de la qualité peuvent utiliser leurs outils de test habituels. Ceux-ci ne font pas la différence entre un système réel et un service virtuel.



La création de composants virtuels vise à simuler un environnement réel selon deux points d'entrée fondamentaux :

- **Observation du système en action** : élaboration d'un composant virtuel à partir de l'écoute du trafic réseau du service à émuler.
- **Lecture des descriptions du système** : élaboration d'un composant virtuel à partir d'autres sources d'information (spécifications de service, par exemple). Exemple : fichier WSDL (Web Services Description Language) décrivant les opérations proposées par un service, avec les paramètres attendus et les données renvoyées.

La virtualisation des services en action

Pour comprendre les avantages de la virtualisation des services, rien de mieux que de prendre des exemples. Cette section présente un exemple fictif et reprend un scénario de client réel.

Exemple : Whiz Bang International

Afin de tester une nouvelle application, l'entreprise Whiz Bang International doit déployer le service de vérification de la solvabilité URGoodForIt (service fictif), issu d'un fournisseur tiers. Or, ce service n'est pas disponible dans l'environnement de test, et l'équipe ne peut pas débiter les tests sans ce composant dépendant. L'équipe est donc contrainte de faire un choix : soit réduire la portée des tests, soit retarder le calendrier de livraison.

La figure 1-1 illustre un exemple d'application de commande en ligne associée au service de vérification de la solvabilité URGoodForIt. Whiz Bang International utilise une architecture orientée services (SOA), et la mise en œuvre de l'application s'appuie sur plusieurs services : service de gestion des commandes, service tiers de vérification de la solvabilité, service tiers de paiement, service personnalisé pour le provisionnement d'un nouveau dispositif et base de données. Cette représentation complète du système montre l'environnement de production sans la virtualisation des services.

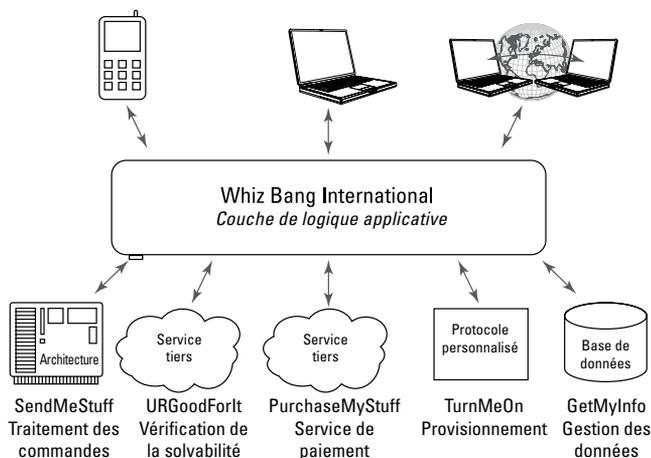


Figure 1-1: application commerciale en production sans la virtualisation des services.

Le service URGoodForIt se prête parfaitement à la virtualisation pour plusieurs raisons :

- La disponibilité de l'environnement de test est retardée, et l'équipe doit attendre que le service soit disponible pour pouvoir débiter les tests.
- Chaque exécution du service URGoodForIt induit un coût.
- L'équipe doit tester les performances pour 100 000 utilisateurs. Le service URGoodForIt étant fourni par un tiers, l'entreprise doit s'acquitter de frais à chaque appel du service dans un test. Le coût lié aux tests de performance pour 100 000 utilisateurs s'enflamme donc très rapidement.

L'indisponibilité de ce service dépendant lors des tests engendre un goulet d'étranglement qui pénalise l'ensemble de l'équipe. Les tests de bout en bout (fonctionnement ou performances) ne pourront débiter que lorsque tous les composants auront été réunis. Ici, la virtualisation du service indisponible débloque la situation. Dans la figure 1-2, on a remplacé un composant de production de la figure 1-1 par un composant virtuel.

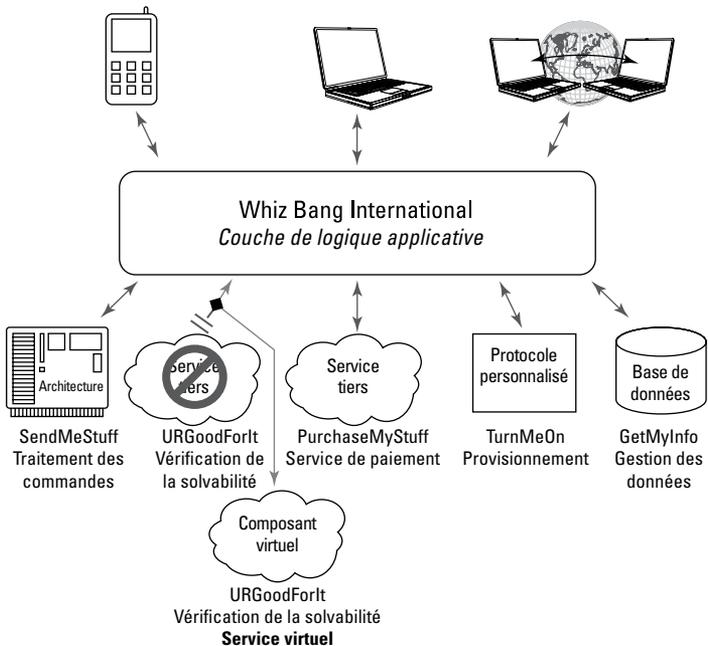


Figure 1-2: a virtualisation des services rend disponibles les composants qui ne le sont pas.



Pour être efficace, toute solution de virtualisation des services doit permettre de créer facilement des composants virtuels qui :

- ✓ Imitent le comportement des véritables composants à l'origine du service
- ✓ Répondent avec des données réalistes
- ✓ Traitent les requêtes dans des plages de débit configurables
- ✓ Peuvent être activés et désactivés lorsque le véritable service devient disponible, sans nécessiter aucune reconfiguration de l'application déployée

Bien sûr, vous devrez à un moment ou un autre tester votre système avec une application de production réelle. La virtualisation des services ne remplace pas le test du code source de l'application composite. Vous ne pouvez pas mettre votre logiciel sur le marché sans avoir réalisé de véritables tests de bout en bout. La virtualisation des services vise à repérer la plupart des défauts à un stade beaucoup plus précoce du processus, lorsque ceux-ci sont plus faciles et moins coûteux à résoudre. Il se peut que les tests de bout en bout rapportent tout de même des erreurs. Si c'est le cas, celles-ci sont en principe peu nombreuses, et sans doute indétectables hors du test de l'application dans son ensemble à l'aide des composants réels.

Exemple : grande société de services financiers

Cette section présente un exemple de terrain (le nom de la société a été supprimé pour des raisons de confidentialité). Une grande société de services financiers souhaitait tester sa nouvelle application de facturation, dont chaque intégration avec les applications internes et chaque dépendance externe tierce. Les intégrations entre les différents composants de l'application exigeaient des tests permanents au cours du processus de développement.

Par exemple, les interfaces entre l'application de facturation et l'application de commande nécessitaient d'être testées indépendamment l'une de l'autre. La configuration de chaque environnement de test occasionnait de longs délais, en raison de l'indisponibilité d'au moins l'un des composants de l'application. L'équipe risquait donc de subir une situation de test critique en fin de cycle. Le département informatique sollicitait des ressources supplémentaires pour exécuter les tests. Comme certaines suites

de tests contenaient des milliers de tests individuels, les délais mettaient en péril l'ensemble du projet, en raison du risque de non-respect d'échéances importantes.

En adoptant la virtualisation des services, l'équipe a réalisé une volte-face majeure dans l'économie de test de sa nouvelle application de facturation. Les professionnels de la qualité ont créé des composants virtuels pour émuler certains aspects de l'environnement de terrain, en enregistrant les messages et les réponses au sein de celui-ci. Cet environnement virtualisé se comportait comme l'application réelle, moyennant un effort de création d'environnements de test considérablement moindre. Par conséquent, inutile pour les développeurs et les professionnels de la qualité de tester leurs modifications à partir des véritables applications dépendantes. A la place, les équipes se sont servi de composants virtuels pour tester l'intégration et les performances des composants inchangés avec les composants modifiés ou les nouveaux composants.

Résultat : l'entreprise a réduit de plusieurs semaines à seulement quelques minutes la durée de mise en place de ses environnements de test. Les gains de temps ont permis de limiter considérablement les coûts liés aux tests, et aux professionnels de travailler plus en profondeur les tests et la validation de la qualité des logiciels.

Différences entre la virtualisation des services et d'autres types de virtualisation

On associe souvent le terme *virtualisation* aux « machines virtuelles » ou à la « virtualisation matérielle ». En réalité, le terme *virtualisation* s'applique à de nombreux aspects de l'informatique : serveurs, applications, réseau, stockage, etc. Généralement, la virtualisation signifie « utiliser des ressources informatiques pour imiter d'autres ressources informatiques ».

En virtualisation matérielle, par exemple, un même serveur physique est partitionné en plusieurs serveurs virtuels. Grâce aux logiciels de virtualisation, chaque machine virtuelle présente l'apparence d'un matériel dédié, ce qui contribue à réduire les coûts matériels. Toutefois, le déploiement de la virtualisation matérielle et de la création d'images de machines virtuelles engendre d'autres coûts.

L'utilisation de machines virtuelles comme environnement de test a beaucoup fait parler d'elle récemment. En effet, ce type d'environnement offre une bonne représentation des éléments envoyés en production, dans un espace d'infrastructure inférieur à celui d'environnements de préproduction physiques intégraux. Les services informatiques ou de test peuvent créer des images virtuelles de l'environnement de production, afin de travailler sur des machines virtuelles sans avoir à procéder à certaines tâches manuelles, ni à assurer les coûts inhérents au provisionnement et à l'élaboration d'un serveur de préproduction physique. L'image virtuelle étant théoriquement très proche de la réalité, cette démarche renforce la confiance du groupe de test. Il existe cependant un inconvénient : la création et la gestion permanente de ces images (suivi de l'utilisation des licences, installations et mises à jour du système d'exploitation, par exemple) peuvent accroître considérablement le coût du projet, sans compter l'attente et le déploiement de chaque composant nécessaire à l'application.

En virtualisation de services, cependant, les *composants logiciels* sont virtualisés grâce à l'émulation de leur interface et à l'imitation de leurs comportements. La virtualisation des services n'émule que ce dont l'environnement de test a besoin, et, par rapport à la virtualisation matérielle, élimine les tâches supplémentaires liées à la gestion des licences, à la configuration et à l'exécution de tous les autres éléments nécessaires sur une machine virtuelle (système d'exploitation, par exemple). Certains investissements sont bien sûr indispensables pour créer les composants virtuels des services. En même temps, les composants virtuels sont disponibles tout au long du cycle de vie de l'application et occupent très peu d'espace (considérablement moins qu'une machine virtuelle). De plus, les composants virtuels sont plus faciles à partager et plus rapides à déployer, car ils sont hébergés sur un serveur spécialement optimisé à cet effet.

Valeur ajoutée de la virtualisation des services

La virtualisation des services peut changer radicalement l'économie et le déroulement du processus de développement d'applications. La qualité des applications repose sur la responsabilité de chacun, du développement, aux tests et au déploiement. La virtualisation des services s'adresse donc à toute l'équipe au cours de l'ensemble du cycle de vie applicatif.

L'un des avantages majeurs de la virtualisation des services est qu'elle permet de réaliser des tests à un stade beaucoup plus précoce du processus de développement, en réduisant les goulets d'étranglement de test. Toutefois, pour déployer tous ses avantages, la technologie de virtualisation des services ne doit pas rester confinée aux processus de test formels.

Réalisation des tests

Les équipes de test efficaces s'accordent sur un principe fondamental : il faut tester tôt et souvent au cours du processus de développement. Pourquoi ? Parce que plus les défauts sont détectés et isolés tôt, plus ils sont simples à résoudre. Facile à dire, direz-vous. Plus les tests débutent tard ou plus la mise en place des environnements de test prend de temps, moins il est probable de pouvoir tester au bon moment, et donc d'obtenir de bons résultats. La virtualisation des services contribue à limiter les coûts des tests et accélère les délais de commencement et d'exécution des tests.

La virtualisation des services offre des avantages quel que soit le type de test, y compris les tests fonctionnels (manuels et automatisés), d'intégration et de performance. Prenons l'exemple des tests de performance. Les performances des applications composites actuelles sont littéralement la somme de la capacité de chaque composant applicatif à répondre dans les temps. La dégradation du temps de réponse d'un quelconque composant peut ralentir les performances de l'application, et donc nuire à l'expérience utilisateur. En reportant les tests de performance à la fin du déploiement de l'application (approche classique), il est beaucoup plus difficile d'identifier la cause du ralentissement et de la mauvaise qualité de l'expérience utilisateur.

La réalisation de tests de performance à l'aide d'un composant virtuel (ou d'un ensemble de composants virtuels) permet de résoudre plus tôt les problèmes les plus évidents. Les tests menés selon la méthode classique, où chacun crée un composant qui est ensuite testé lorsque tout est prêt, permettent rarement d'atteindre les objectifs de performance. Cette méthode ne permet pas de savoir avant la fin du cycle de développement et de test qu'un composant présente un temps de réponse trop long et nuit à l'expérience utilisateur, par exemple. Ce processus est présenté plus en détail au chapitre 4.



Il est essentiel de comprendre les attentes des clients ou des utilisateurs finaux dès le début du processus de développement, notamment en matière de fonctionnalités, de simplicité d'utilisation,

de fiabilité, d'assistance et de performances, ainsi que la logique applicative impliquée. Vous devez par ailleurs calculer et déterminer les spécifications de performances nécessaires pour chaque composant. Avec ces données en main, vous pourrez simuler dès le premier jour les conditions exactes nécessaires à l'équipe de développement au sein de votre environnement de test virtuel.

Développement

Pourquoi les développeurs ont-ils besoin de la virtualisation des services ? En plus du processus de test formel géré par l'équipe de test, les développeurs doivent tester leur propre code en permanence. La virtualisation des services permet de simuler les environnements de tests, quels qu'ils soient, pendant que les développeurs écrivent le code, au-delà des tests de compilation et unitaires.

Grâce à la virtualisation des services, les développeurs n'ont plus besoin d'écrire manuellement leurs propres simulations ou éléments de remplacement (objets fictifs visant à imiter des objets réels lors des tests, par exemple). Les éléments de remplacement nécessitent parfois une maintenance ultérieure, qui monopolise le temps dont les développeurs ont besoin pour créer de nouvelles fonctionnalités. Dans les environnements de développement agiles ou itératifs, où les équipes doivent tester de nouvelles fonctionnalités en permanence au cours du processus de développement, l'absence d'obligation d'écrire manuellement les éléments de remplacement peut être particulièrement utile.

Utilisation hors production

Grâce à la virtualisation des services, les entreprises ont la possibilité d'élaborer des environnements de formation réalistes qui ne sont pas liés aux ressources de l'environnement de production réel. Le scénario ci-après montre comment la virtualisation des services permet l'élaboration de programmes de formation plus efficaces.

Ne pas évaluer son propre travail

Vos développeurs écrivent leurs propres simulations ou éléments de remplacement pour valider leur code ou leurs modifications ? Mauvaise idée. C'est comme si l'on demandait à un étudiant de noter son propre travail. Les testeurs, qui savent comment les

fonctionnalités doivent être testées et quelles sont les données nécessaires pour tester correctement divers scénarios, sont en mesure de créer facilement des tests et composants virtuels, afin de les proposer à l'équipe de développement.

Une entreprise souhaite former ses nouveaux personnels de service client à l'utilisation de son système de gestion de la relation client (CRM). Ces personnels ont accès à une version de production du système de CRM pour simuler des situations réelles. Or, le système de CRM est relié à plusieurs systèmes dorsaux. Pour répondre à une série de questions d'évaluation, le personnel en formation doit créer des requêtes sur des informations de produits et de tarification gérées par plusieurs systèmes dorsaux. Dans ce cas, la virtualisation des services permet d'émuler les systèmes dorsaux, afin d'exécuter les opérations CRUD (créer, lire, mettre à jour et supprimer) sans interférer avec les systèmes de production réels.



La virtualisation des services n'isole pas simplement le personnel en formation des systèmes de production. Elle facilite également la réinitialisation de l'environnement de formation vers un état et des données connus. Par rapport au déploiement d'un système complet dans un environnement de formation isolé, cette approche est considérablement moins coûteuse.

Avantages de la virtualisation des services

La virtualisation des services permet de tester de manière permanente, plus précoce et parallèle des applications complexes tout au long du cycle de vie de développement. Cette approche est particulièrement utile pour les applications composées de services interconnectés au sein d'un environnement SOA, où les tests sont souvent retardés dans l'attente que tous les services soient prêts et déployés. La virtualisation des services permet donc aux entreprises de bénéficier des avantages présentés dans cette section.

Réduction des coûts

Une infrastructure de test physique peut engendrer des coûts considérables. Contrairement au provisionnement de serveurs ou d'infrastructures de grande envergure, les environnements de test virtuels peuvent être exécutés sur des matériels standard peu coûteux. Par ailleurs, ces environnements sont facilement reconfigurables selon les projets ou les besoins en matière de test.

La virtualisation des services limite également les coûteux tests de bout en bout susceptibles de créer des situations critiques au niveau des délais. En effet, les tests fonctionnels, d'intégration et applicatifs ayant déjà été exécutés de nombreuses fois, la majorité des défauts a en principe été détectée. La réalisation de ces tests tout au long du cycle de vie de développement accélère les tests de bout en bout, et permet d'éliminer les besoins supplémentaires en personnel de test.

Un grand nombre d'applications composites actuelles s'appuie sur des services fournis par des tiers. Or, certains de ces fournisseurs facturent des frais à chaque exécution de leurs services. En virtualisant ces services à des fins de test, l'on peut réduire les frais d'accès aux services tiers sans toutefois restreindre les activités de test. Par ailleurs, la virtualisation des services résout un problème fréquent concernant les services tiers : ceux-ci ne sont pas toujours disponibles lorsque les équipes de test en ont besoin. Avec la virtualisation des services, en revanche, les services sont toujours disponibles au bon moment.

Accroissement de la productivité

Les contraintes qui pèsent sur les développeurs et les ingénieurs qualité peuvent être un frein à la productivité. Les environnements de test physiques présentent un certain nombre de contraintes :

- ✓ Délai de provisionnement des environnements
- ✓ Périodes d'indisponibilité des environnements auprès des développeurs et des testeurs
- ✓ Limite du nombre de scénarios de test pouvant être traités par un système

La virtualisation des services n'induit aucune contrainte dans la manière de procéder aux tests ou au développement. Les composants virtuels sont disponibles 24 h/24, 7 j/7. Ceci accroît considérablement la productivité et libère des ressources, qui

peuvent alors être consacrées à d'autres activités génératrices de valeur ajoutée ou à l'amélioration du processus de test (intégration de tests exploratoires, par exemple).

Réduction des risques

La virtualisation des services contribue également à réduire les risques. Elle permet de tester les logiciels à un stade plus précoce du processus. Les défauts peuvent ainsi être résolus plus tôt, ce qui engendre moins de surprises en fin de programme. Le produit final peut être mis en production plus tôt et avec moins d'erreurs. Par ailleurs, les équipes de grande envergure peuvent travailler efficacement en parallèle, en collaborant sur différentes parties d'une application en virtualisant les composants de l'ensemble du système, et mettre en place un plan qui permettra de tester le système correctement et de vérifier que les différents composants fonctionnent ensemble.

Amélioration de la qualité

En améliorant l'efficacité des tests, la virtualisation des services accroît la qualité globale des applications. Les équipes peuvent ainsi tester les applications plus en profondeur, et mettre sur le marché plus rapidement des logiciels de meilleure qualité.

Chapitre 2

Les moteurs du changement

.....

Dans ce chapitre

- ▶ Répondre aux attentes en matière d'applications d'entreprise
 - ▶ Appréhender les architectures orientées services
 - ▶ Exploiter les capacités mobiles
 - ▶ Répondre aux attentes des clients en perpétuelle évolution
-

Depuis toujours, les services informatiques s'efforcent de trouver le juste équilibre entre le temps et les ressources qu'ils consacrent à la gestion de la qualité (souvent dominée par les tests) et le risque de produire des applications de mauvaise qualité. Auparavant, les décisions en matière de processus de qualité étaient l'affaire des services informatiques, et la sphère commerciale s'impliquait peu dans ces décisions. Aujourd'hui, les logiciels sont indissociables de la stratégie d'entreprise et jouent un rôle majeur sur le marché. C'est pourquoi les tests ne peuvent plus être considérés comme une activité autonome. Les clients, partenaires et fournisseurs associent souvent la valeur et la réussite d'une entreprise à la qualité de ses logiciels. Il n'est donc pas surprenant que la gestion de la qualité soit désormais fondamentale dans le cycle de vie applicatif.

Ce chapitre présente l'évolution du lien entre les services informatiques et l'entreprise. Aujourd'hui, l'informatique n'assiste plus simplement l'entreprise, mais elle joue un rôle fondamental dans la création de valeur. De nouvelles architectures répondent désormais aux pressions subies par les entreprises, ainsi qu'aux problématiques des environnements dynamiques, aux opportunités et défis liés à l'émergence de nouveaux canaux de diffusion d'applications et à l'évolution de l'organisation informatique elle-même (des processus de développement à l'exploitation). Face à la nécessité sous-jacente permanente d'améliorer la qualité des logiciels, les entreprises doivent

repenser leur approche classique des tests pour réussir dans un environnement en rapide évolution tant au niveau technique que commercial.

Répondre aux attentes croissantes en matière d'applications d'entreprise

En tant que dirigeant d'entreprise, vous devez rester centré sur votre stratégie, quel que soit votre objectif : dépasser vos concurrents, vous développer ou contrôler les coûts afin d'atteindre une rentabilité maximale. Pour atteindre ces objectifs dans l'environnement dynamique actuel ciblé sur le consommateur, vous devez déployer un niveau de collaboration élevé entre les sphères informatique et commerciale. Auparavant, le rôle du service informatique se limitait essentiellement à développer des applications visant à répondre aux besoins des services internes. Or, ces dernières années, l'informatique a connu deux évolutions majeures :

- ✓ Davantage expérimentés en matière de technologie, les utilisateurs ont des attentes plus importantes.
- ✓ Le rôle des services informatiques a évolué, et implique désormais le développement et le déploiement d'applications innovantes dont l'interface est orientée vers l'extérieur.

Aujourd'hui, ces applications sont indispensables pour développer de nouveaux canaux de partenariat et enrichir l'interaction des clients avec l'entreprise. En d'autres termes, l'informatique a évolué d'un rôle de support vers un rôle fondamental dans la stratégie de l'entreprise. Les dirigeants attendent des services informatiques qu'ils répondent à des besoins en constante évolution et qu'ils satisfassent des objectifs métier extrêmement rapidement. Les dirigeants considèrent que la technologie ne doit plus être un frein au changement.

Ce lien étroit entre stratégie d'entreprise et innovation informatique nécessite de comprendre certaines pressions majeures auxquelles les entreprises font face. Les applications adaptées aux nouveaux modèles d'entreprise doivent être capables de traiter des volumes élevés de données variées et de s'intégrer avec un large éventail de systèmes nouveaux et existants. Par ailleurs, les entreprises doivent simplifier et automatiser leurs processus manuels, afin de pouvoir développer et déployer rapidement des applications de haute qualité.

Pour supporter ce nouvel environnement collaboratif entre les sphères informatique et commerciale, de nombreuses entreprises intègrent de nouvelles technologies afin d'accroître leur réactivité et la valeur ajoutée côté utilisateurs :

- ✓ Modernisation des systèmes back-office traditionnels afin de mettre de nouvelles fonctionnalités sur le marché
- ✓ Usage élargi de services partagés modulaires permettant d'assembler rapidement de nouvelles applications composites
- ✓ Support plus vaste de nouveaux canaux de diffusion d'applications, en particulier sur appareils mobiles
- ✓ Utilisation du cloud computing pour accroître la flexibilité et l'évolutivité de l'entreprise

Adoption d'architectures orientées services

Les architectures ont suivi l'évolution rapide de l'innovation et de la croissance des entreprises. Pour de nombreuses entreprises, l'approche orientée services permet de mieux répondre aux besoins d'évolutivité et de flexibilité.



L'approche orientée services offre d'autres avantages :

- ✓ Meilleure capacité à créer des applications plus sophistiquées en associant des services métier modulaires réutilisables
- ✓ Croissement de la réactivité et des performances informatiques
- ✓ Capacité à échanger des données avec des entreprises externes (fournisseurs, partenaires, etc.)
- ✓ Flexibilité de consommation de services tiers
- ✓ Support de diverses topologies de déploiement, dont le cloud privé et public
- ✓ Extension de la normalisation de l'environnement informatique

L'architecture orientée services s'appuie sur la mise en œuvre de processus métier sous forme de services logiciels. Ces services sont constitués de composants librement associés, conçus pour gérer les dépendances et favoriser la réutilisation, et assemblés

de sorte à mettre en œuvre des tâches métier précises. La conception à partir de services métier modulaires permet d'obtenir des systèmes informatiques plus efficaces et plus flexibles. L'orientation services offre par ailleurs une approche et une méthodologie *métier* qui aide les entreprises à évoluer et à s'adapter en fonction des forces du marché.

Les caractéristiques fondamentales de l'orientation services sont la modularité, la capacité de réutilisation et la flexibilité :

- ✔ **Modularité** : évolution d'applications étendues, monolithiques, complexes et difficiles à gérer vers des services métier réutilisables basés sur plusieurs composants.
- ✔ **Capacité de réutilisation** : les règles et la logique des composants applicatifs sont communes aux processus métier fondamentaux, et intégrées pour créer des services métier réutilisables.

En utilisant un composant testé et fiable, l'entreprise accélère le développement, accroît le niveau de sécurité et de confiance, réduit les risques et réalise des économies.
- ✔ **Flexibilité** : caractéristique de modularité et de capacité de réutilisation des services métier.

Les architectures orientées métier ont donné le jour à tout un secteur, qui fournit aux entreprises des services métier parfaitement adaptés à de nombreux domaines, dont les services de paiement, la vérification de solvabilité et la disponibilité des stocks. Ce nouvel environnement crée de nouvelles alternatives au développement et au déploiement de logiciels sous-jacents en interne. Mais il génère aussi de nouveaux défis en termes de gestion des versions et de garantie de la qualité.

Essor des applications pour mobile

Le cycle de vie de développement et de déploiement d'applications connaît une véritable mutation liée aux attentes des consommateurs et aux exigences des plateformes. Ce tournant majeur, tant au niveau informatique que commercial, repose sur plusieurs facteurs :

- ✔ Demande d'applications pour mobile
- ✔ Nécessité d'intégrer les nouveaux « systèmes d'engagement » avec les systèmes back-office existants
- ✔ Opportunité d'innovation liée à de formidables capacités mobiles (géolocalisation, par exemple)

Aujourd'hui, les consommateurs veulent pouvoir interagir de plusieurs façons avec les entreprises et leurs services, y compris à partir d'ordinateurs portables, de tablettes ou de smartphones. Chacun de ces appareils intègre un système d'exploitation spécifique et se présente sous différents formats. De plus en plus, les utilisateurs finaux peuvent choisir la plateforme avec laquelle ils interagissent avec votre entreprise. Le développement et les tests nécessitent donc de nouvelles approches adaptées au large éventail d'appareils utilisateur.

Ces applications émergentes ne s'exécutent pas de manière isolée, mais se connectent à des systèmes back-office. Les applications orientées vers le client doivent être mises en œuvre selon le bon niveau de performance et de qualité, en fonction du processus métier adapté. Or, ceci peut faire peser de nouvelles exigences sur les systèmes sous-jacents, qui doivent être testés.

Les appareils mobiles offrent de nouvelles capacités exceptionnelles, mais engendrent aussi des contraintes. Certaines fonctionnalités, telles que la géolocalisation, les interfaces voix et la communication en champ proche ouvrent de nouvelles avenues d'innovation. Toutefois, comparés aux PC modernes, les petits écrans, la bande passante limitée et la latence élevée du réseau rappellent les limites actuelles de ces technologies. Sans parler de l'importante variabilité liée à l'appareil, à l'opérateur et au lieu. Face au rythme de l'innovation dans ce domaine, les entreprises ne peuvent pas anticiper l'évolution des exigences futures de leurs clients. Les approches de développement, de test et de déploiement doivent être capables de suivre l'évolution des modèles d'interaction privilégiés.

Poursuite d'une transformation agile

De nombreuses équipes de développement logiciel emploient des approches agiles pour accélérer le processus de développement. Par conséquent, le processus de test doit lui aussi s'accélérer, afin que l'entreprise puisse mettre sur le marché plus rapidement de nouvelles applications de meilleure qualité. Les processus de développement agiles s'appuient sur des itérations de développement courtes qui intègrent la planification, les tests et l'intégration en continu. L'objectif consiste à faire avancer les projets rapidement à partir d'un environnement hautement collaboratif.

Cette approche n'est pas réalisable si les tests prennent du retard sur le développement. Malheureusement, ce retard n'est pas

rare. Il se peut, par exemple, que l'équipe de test soit contrainte de passer plusieurs heures la nuit à réinitialiser manuellement l'environnement de test. Il se peut aussi que les tests applicatifs soient retardés dans l'attente des logiciels dépendants. D'autres retards peuvent se produire lorsque la durée du cycle d'exécution des tests manuels dépasse celle du sprint de développement. Si l'équipe de test n'est pas en mesure de suivre le rythme de l'approche de développement agile, le service informatique peut accuser de nombreux retards et manquer ses échéances de livraison.

Il y a seulement quelques années, une application n'était modifiée que quelques fois par an. Aujourd'hui, les environnements de développement, de déploiement et d'exploitation logiciels sont soumis à des évolutions permanentes. Il n'est pas rare qu'une application soit modifiée chaque semaine, voire quotidiennement. Les équipes de développement et d'exploitation (production) logiciels ne peuvent plus fonctionner de manière indépendante, avec leurs propres tâches, procédures de déploiement et calendriers.

Chapitre 3

Evolution vers l'avenir

.....

Dans ce chapitre

- ▶ Amélioration de la qualité au cours du cycle de vie des applications
 - ▶ Une autre perspective de l'automatisation des tests
 - ▶ Défis des environnements de test complexes
 - ▶ Utilité de la virtualisation des services
-

Face à des applications de plus en plus complexes impliquant davantage d'interfaces et d'options de diffusion, il est plus important que jamais d'améliorer en permanence les pratiques de gestion de la qualité logicielle. Or, tandis que les entreprises s'efforcent de développer leur agilité et leur réactivité pour répondre aux exigences du marché, les équipes de test ont souvent du mal à suivre, ce qui crée un goulet d'étranglement dans le processus de production logicielle.

Ce chapitre présente plusieurs approches d'amélioration des processus de gestion de la qualité. Nous allons aborder certaines bonnes pratiques de mise en place de procédures de test plus sophistiquées et centrées sur la qualité. Face à des applications et des environnements de test de plus en plus complexes, la virtualisation des services répond à des défis fondamentaux et améliore l'efficacité des équipes grâce à la réalisation de tests plus sophistiqués et plus précis à un stade plus précoce du cycle de vie.

Améliorer la qualité au cours du cycle de vie applicative

Dans presque tous les secteurs, la qualité des logiciels représente une problématique coûteuse. La résolution des problèmes logiciels coûte en effet des milliards d'euros chaque année. De nombreuses entreprises n'ont semble-t-il pas encore réalisé que l'activité

de test constitue un maillon dans la production de logiciels de qualité. Pour produire des logiciels fiables, simples d'utilisation, disponibles, faciles à gérer et évolutifs qui répondent aux objectifs métier, le cycle de vie applicatif doit être plus évolué en termes de planification, de collaboration, de traçabilité et de précision des informations, de la phase d'étude des besoins à la phase de déploiement.

Les entreprises optent trop souvent pour une approche réactive de l'amélioration de la qualité, au lieu de mettre en œuvre un processus de qualité proactif optimisé, basé sur la compréhension de l'évolution des besoins et des risques. La collaboration et la traçabilité sont la clé des équipes proactives. Elles savent quels composants de l'application ont été modifiés au cours du développement, et peuvent cibler les tests sur ces aspects précis. Elles minimisent en même temps les risques pour l'entreprise.

Même en employant les meilleures pratiques qui soient en matière de qualité, les logiciels (et les matériels) peuvent s'accompagner de défauts lors de leur mise sur le marché, dont un grand nombre n'a pas été détecté au préalable. Dans la course éfrénée à la production logicielle, il est impossible de tester chaque chemin de code et état logiciel. De plus, les utilisateurs ont changé. Ils sont relativement tolérants face à certains défauts initiaux. Mais ils attendent que ces défauts soient résolus rapidement par la mise à disposition fréquente de nouvelles versions.

Pour satisfaire les exigences actuelles des utilisateurs finaux, les entreprises doivent mettre en place des processus capables de détecter, d'isoler et de résoudre plus tôt les défauts. Par ailleurs, pour traiter les problèmes qui seraient passés entre les mailles du filet avant la production, les entreprises doivent impérativement appliquer une stratégie de résolution rapide des défauts et de mise sur le marché accélérée de correctifs.



La virtualisation des services contribue à améliorer les processus de gestion de la qualité grâce à plusieurs atouts :

- ✓ **Tests d'intégration plus précoces** : les composants virtuels simulent les interfaces de service dont le système testé a besoin. Inutile d'attendre un stade avancé du cycle de vie de développement pour tester les interfaces. Les composants virtuels permettent de tester à un stade plus précoce du processus. En réalité, la virtualisation des services fait partie intégrante du processus de versionnage standard.
- ✓ **Disponibilité plus précoce de l'environnement de test et réduction des coûts** : il est possible de mettre en

place un environnement de test prêt à l'emploi dans un délai relativement court. En émulant les environnements dépendants à l'aide de composants virtuels, les tests peuvent débiter sans délai. Par ailleurs, le coût des environnements de test (matériel, logiciels et main-d'œuvre) est moins important.

- ✓ **Possibilité de tester plus tôt au cours du développement :** les développeurs et les professionnels de l'assurance qualité peuvent tester les intégrations à un stade plus précoce du processus en utilisant un ensemble partagé de composants virtuels, qui permet également de développer en parallèle et de produire des logiciels de meilleure qualité.

Repenser l'automatisation des tests

Les processus d'automatisation des tests sont les mêmes depuis de nombreuses années. Ils se basent généralement sur des interfaces utilisateur :

1. **Attendre la stabilisation de l'interface utilisateur.**
2. **Mettre en place l'environnement de test en déployant chaque composant de l'application à mesure de sa mise à disposition.**
3. **Enregistrer les interactions utilisateur par l'intermédiaire de l'interface.**
4. **Ajuster les enregistrements si nécessaire pour améliorer les scénarios de test.**
5. **Exécuter les tests.**
6. **Réinitialiser l'environnement et réexécuter les tests (en espérant ne pas avoir à effectuer trop d'ajustements ou de réenregistrements).**
7. **Gérer une bibliothèque de scripts de test à mesure des modifications de l'application à chaque itération, souvent en réenregistrant des scénarios entiers.**



Par rapport aux méthodes manuelles, l'automatisation des tests d'interface permet de tester plus rapidement, et donc plus souvent. Cependant, les tests d'interface sont relativement fragiles. Il n'est pas rare qu'ils soient perturbés par les modifications du code, même lorsque celles-ci sont invisibles. Ce problème est d'autant plus sévère lorsque les scripts sont créés en enregistrant les interactions utilisateur. Le compromis pour limiter les coûts de maintenance permanente des tests consiste à attendre que les modifications de l'interface utilisateur soient finalisées. Bien sûr,

cette démarche introduit le risque de ne pas détecter certains problèmes suffisamment tôt pour qu'ils puissent être résolus dans les délais du calendrier d'origine du projet.

L'interface utilisateur est la face émergée de l'iceberg : la plupart du code et de la complexité se trouve en dessous. Le moyen le plus direct d'identifier la cause d'un défaut consiste à le rechercher près de l'endroit où il s'est produit, sans que d'autres couches applicatives puissent potentiellement masquer l'origine du problème (par exemple, des exceptions introduites par du code). Ceci nécessite une approche élargie de l'automatisation des scénarios d'essai, par exemple, tester chaque couche de l'architecture indépendamment.

Le fait d'effectuer les tests au niveau de la couche API (interface de programmation d'applications), où les composants et les applications communiquent entre eux, peut améliorer l'efficacité des tests et réduire les risques pour plusieurs raisons :

- C'est souvent à cet endroit que les applications présentent des problèmes, aux interconnexions entre les sous-systèmes.
- Ces frontières correspondent souvent à des frontières organisationnelles et de planification. Par conséquent, le fait de résoudre les problèmes à cet endroit peut s'avérer coûteux ou difficile, en particulier lorsque les problématiques sont détectées tard.
- Les interfaces de service sont par nature plus stables que les interfaces utilisateur, car les mêmes spécifications de service supportent de nombreuses applications. Les modifications sont généralement bien gérées entre les parties prenantes pour prévenir les ruptures au niveau de la production. L'exécution de tests automatisés sur cette couche nécessite donc moins de maintenance.

Il existe une synergie naturelle entre la virtualisation des services et l'automatisation des tests d'interface de service. Les tests sollicitent une interface de service spécifique en générant des requêtes et en validant les réponses. Les composants virtuels reçoivent ces requêtes, émulent le comportement du service de terrain et envoient les réponses appropriées sur un nombre indéfini de protocoles compatibles et dans divers formats de message. Les outils bénéficient de cette synergie grâce au partage des définitions de protocole et des ensembles de données entre les services virtuels et les tests de service automatisés. Parmi les bonnes pratiques, il est bon de créer une suite de tests sur les interfaces de service, qui servira à valider à la fois le composant de production et le composant virtuel.

Faire face aux défis d'environnements de test complexes

La création d'environnements de test hébergeant les applications complexes d'aujourd'hui peut présenter des difficultés, surtout lorsque les applications de test s'interfacent avec d'autres systèmes internes, voire externes. Ceci implique d'autres défis :

- ✔ La duplication d'un environnement de production complet en vue de réaliser des tests peut s'avérer à la fois coûteuse et chronophage. Par exemple, un serveur de production classique sous Windows Server 2012 peut coûter des dizaines de milliers d'euros, voire davantage selon la configuration. Les coûts d'exploitation peuvent donc s'enflammer très vite lorsque plusieurs serveurs sont nécessaires.
- ✔ La création de ces environnements requiert de nombreuses connaissances et compétences techniques. De plus, la configuration d'un environnement de test peut nécessiter de l'expertise en matière d'applications, mais aussi d'administration de systèmes. Prenons l'exemple d'une application qui s'interface avec un système d'ERP (SAP, Oracle, Siebel, etc.). Dans ce cas, l'équipe de test ne possède pas nécessairement les compétences pour déployer une instance du système d'ERP afin de tester l'application.
- ✔ La planification de tests impliquant des services tiers est souvent coûteuse et compliquée.

Privés d'un environnement de test complet, certains développeurs expérimentés peuvent tenter de créer leurs propres simulations ou éléments de remplacement. Cette approche peut servir aux développeurs pour réaliser des tests unitaires. Mais elle est rarement adaptée à l'ensemble d'une l'équipe, pour plusieurs raisons :

- ✔ Le développement d'une simulation réaliste capable de supporter tous les scénarios et objectifs de test est une opération compliquée qui peut se transformer rapidement en initiative majeure de développement et de maintenance, et consommer le temps dont l'équipe de développement a besoin pour travailler sur l'application.
- ✔ Les développeurs doivent souvent modifier l'application sous-jacente afin de substituer les éléments de remplacement aux composants réels, ce qui induit des différences entre

l'application testée et l'application préparée pour la production.

- ✓ Aucune infrastructure ne permet de partager des éléments de remplacement au sein d'équipes.
- ✓ Des problèmes peuvent facilement passer inaperçus lorsque les tests sont réalisés par les développeurs qui ont créé le code.

Virtualisation des services et environnements de test complexes

Face à une complexité croissante, la technologie de virtualisation des services permet aux entreprises de se centrer davantage sur la qualité et l'efficacité. Elle permet aux développeurs et aux testeurs de créer des composants virtuels qui peuvent être partagés, afin de procéder au développement parallèle au sein d'équipes. De plus, en émulant les services, les applications ou les systèmes de terrain, les composants virtuels peuvent contribuer à éliminer les retards occasionnés par le processus de test. Par ailleurs, ces composants fonctionnent sur du matériel standard et réduisent le coût lié au support de plusieurs environnements de test, ce qui allège les préoccupations de l'équipe d'exploitation en matière de capacité, d'évolutivité et de sécurité.

Chapitre 4

Comment évoluer vers la virtualisation des services

.....

Dans ce chapitre

- ▶ Identification des services à virtualiser
 - ▶ Automatisation des tests
 - ▶ Virtualisation par objectif et phase de test
-

Si vous avez lu les chapitres précédents, vous pouvez maintenant comprendre les avantages de la virtualisation de service et son rôle dans le processus de qualité logicielle. A présent, vous vous demandez probablement par où commencer. Au premier abord, la tâche peut sembler phénoménale. Vous pouvez être tenté de choisir un élément à virtualiser au hasard, juste pour observer comment fonctionne la virtualisation des services. Mais vous obtiendrez sans aucun doute un meilleur retour sur investissement en évaluant de manière mesurée vos défis en matière de test, afin de hiérarchiser vos décisions concernant les éléments à virtualiser en premier.

Ce chapitre propose une méthode pour analyser quels services vous devez virtualiser. Nous verrons comment la virtualisation des services et l'automatisation des tests se complètent, et l'utilité de la virtualisation des services à chaque phase de test, des tests unitaires au test d'acceptation client, et même aux tests de performance.

Identification des services à virtualiser

Pour optimiser votre réussite avec la virtualisation des services, vous devez d'abord identifier les services à virtualiser. Comment procéder ? Dans un premier temps, rassemblez les principales parties prenantes du processus de développement impliquées dans

le cycle de vie applicatif, et commencez à réfléchir aux domaines de test dans lesquels votre entreprise rencontre le plus de problèmes. Puis posez-vous les questions suivantes :

- ✔ Possédez-vous tous les environnements nécessaires pour réaliser vos tests d'intégration ?
- ✔ Ces environnements de test sont-ils à la disposition des équipes tout au long du cycle de développement ?
- ✔ Subissez-vous des interruptions provoquées par l'indisponibilité des environnements de test ?
- ✔ A quelle fréquence les interruptions se produisent-elles ? Combien de temps vos équipes doivent-elles généralement patienter ?
- ✔ Quel est l'impact des interruptions de test sur les délais et les coûts ?
- ✔ Votre application s'interface-t-elle avec des services tiers ?
- ✔ Devez-vous payer et planifier l'accès aux interfaces tierces avant de planifier vos tests ? Combien cela vous coûte-t-il ?
- ✔ Qui contrôle les informations nécessaires à la création d'environnements de test ?
- ✔ Le partage d'environnements de test (ou de parties d'un environnement de test) crée-t-il des conflits au niveau du personnel ou des équipes ?

En répondant à ces questions, vous pouvez hiérarchiser les domaines dans lesquels la virtualisation des services pourrait vous être utile. Pour commencer, nous vous conseillons l'approche suivante :

- ✔ **Commencez par les domaines les plus facilement exploitables qui apportent le plus d'avantages moyennant un minimum de travail** : il peut par exemple s'agir d'un service Web défini dans un fichier WSDL qui renvoie des données, mais qui n'autorise pas les effets secondaires. Ce type de service est facile à virtualiser. Il n'y a pas de modification d'état complexe à modéliser. Il suffit de mapper les arguments d'entrée sur les données renvoyées dans la réponse. Facile à créer, ce service peut également éliminer un problème de planification d'environnement de test potentiellement source de retard au niveau du projet.
- ✔ **Centrez-vous sur les aspects qui contribuent au coût global des tests** : les services d'intégration sont souvent de bons candidats à la virtualisation. Si le point final (endpoint) est un

système d'ERP, par exemple, il faut beaucoup de matériel et de main-d'œuvre, et un ensemble de données raisonnable pour déployer une copie du système au sein de l'environnement de test. Toutefois, vous avez également la possibilité de simuler ce point final.

- ✓ **Traiter les dépendances sur les services tiers** : si vous devez vous acquitter de frais à chaque fois que vous accédez à certains services tiers, le principe est le même. L'émulation peut être coûteuse de par sa complexité, mais elle peut s'avérer indispensable. C'est là que la réalisation d'une analyse coûts-avantage prend tout son sens.

Analyse coûts-avantages



Pour décider quels éléments virtualiser, il suffit souvent de réaliser une analyse coûts-avantages. Les coûts sont engendrés par plusieurs facteurs :

- ✓ **Impact de l'indisponibilité** : perte de productivité des équipes et retards au cours des projets dus à l'indisponibilité de certains services/logiciels dépendants lors des tests
- ✓ **Coût du personnel qualifié** : acquisition et conservation de personnel possédant l'expertise nécessaire pour configurer et gérer les environnements de test
- ✓ **Sous-utilisation des environnements de test** : utilisation inefficace de matériels coûteux qui engendre des coûts d'infrastructure élevés inutiles
- ✓ **Coût des licences** : logiciels (systèmes d'exploitation, systèmes de gestion de bases de données, etc.) déployés dans l'environnement de test physique ou sur une machine virtuelle
- ✓ **Frais d'accès aux services tiers** : frais induits par chaque exécution d'un service externe

En adoptant la virtualisation des services, vous limitez ces coûts et bénéficiez de nombreux avantages (économies d'infrastructure, accroissement de la productivité, accélération de la mise sur le marché, etc.), dont chacun contribue au retour sur investissement global. Vous devez également envisager une dynamique de système plus évoluée pour identifier les éléments à virtualiser.

Volatilité des services

La *volatilité des services* désigne la fréquence à laquelle l'interface ou le comportement d'une application, d'un système ou d'un composant change. Les composants amont, l'environnement de

test et les tests eux-mêmes doivent réagir à ces changements. Les changements doivent être gérés de sorte à ne nuire ni à la qualité, ni aux délais. Posez-vous les questions suivantes :

- ✔ Le service dépendant est-il toujours en cours de développement et soumis à des modifications de spécifications permanentes ? Vous avez la possibilité de créer un composant virtuel dont la modification nécessite moins de tâches, tout en n'empêchant pas les tests d'intégration de se poursuivre.
- ✔ La dépendance concerne-t-elle une application patrimoniale rarement modifiée ? Si c'est le cas, vous pouvez créer un composant virtuel pour simuler l'ensemble de l'application/du système à la frontière du service.
- ✔ Vos équipes ont-elles besoin de mises en œuvre légèrement différentes d'un même service ? Les composants virtuels permettent de modéliser et de simuler des comportements spécifiques, afin de répondre aux besoins de plusieurs équipes et de soutenir les tests lors du développement parallèle.

Impact de l'indisponibilité

Quel est l'effet global sur les processus de test de l'indisponibilité d'un système, d'une application ou d'un composant dans l'environnement de test ? L'absence de certains composants peut n'impacter que quelques scénarios de test. Cependant, l'indisponibilité de certains autres composants peut faire planter l'ensemble de l'application et bloquer tous les tests.

Lors de l'évaluation de l'impact d'un composant indisponible, posez-vous les questions suivantes :

- ✔ Combien de testeurs seront inactifs et pendant combien de temps ?
- ✔ Quels scénarios de test faut-il réduire (ou retarder) si aucun composant n'est disponible ?

Chacun de ces facteurs est associé à un coût. Pour réduire ce coût, vous devez virtualiser les éléments permettant de débloquer les tests stratégiques.

Coût de déploiement ou d'utilisation

Calculez le coût de déploiement du système, de l'application ou du composant. Déterminez le coût de création physique d'un environnement de test. Certaines technologies sont plus coûteuses à déployer que d'autres, même en environnement de test. Peut-être utilisez-vous des services tiers facturés à la demande. En évaluant

le coût de création d'environnements de test ou d'exécution de services facturés à la demande, vous pouvez décider très facilement quels éléments virtualiser.

Complexité de la technologie

Bien qu'il soit possible de virtualiser de nombreuses technologies, certaines peuvent être plus simples à virtualiser que d'autres. Dans certains cas, la technologie est facile à virtualiser, mais l'émulation du comportement métier du service peut s'avérer plus compliquée. Évaluez les diverses technologies au sein de l'environnement à l'aide d'une échelle de notation. Le processus d'évaluation vous aidera à comparer la complexité et l'effort nécessaire pour simuler chaque dépendance.

Calculs

La planification est le maître mot de la virtualisation des services. Vous devez impérativement planifier votre analyse. Vous pouvez regrouper les données sous forme de tableau. Le tableau 4-1 contient l'ensemble de composants de l'exemple d'application développée par Whiz Bang International (voir le chapitre 1). La technologie employée pour chaque composant est indiquée dans la colonne « Technologie utilisée ». Par exemple, le composant PurchaseStuff utilise la technologie SOAP via HTTPS. La colonne « Impact » décrit l'impact d'une contrainte d'indisponibilité ou d'un autre coût (service tiers, par exemple) sur le test du composant. Par exemple, la mise en place d'un environnement de test physique pour le composant de provisionnement TurnMeOn s'élèverait à 100 000 euros. Ou peut-être que l'environnement existe déjà, mais n'est pas disponible en raison de conflits de planification. Le coût des retards liés à l'incapacité de procéder aux tests s'élève à 100 000 euros. La dernière colonne indique le niveau de complexité de la virtualisation des services, que vous évaluez sur une échelle de 1 à 10 ou à l'aide des critères « élevée », « moyenne » ou « faible », selon le système d'évaluation qui vous convient le mieux. Ici, nous utilisons une échelle de 1 à 10, où 1 indique une faible complexité, et 10 une complexité très élevée.

Tableau 4-1 Composants de l'application de Whiz Bang International

<i>Technologie utilisée</i>	<i>Composant/Fonctionnalité</i>	<i>Impact de l'indisponibilité + Coût de déploiement ou d'utilisation</i>	<i>Niveau de complexité de la virtualisation des services (1-10)</i>
SOAP via HTTPS	Service de vérification de la solvabilité URGoodFort	226 000 € (300 000 \$)	1
SOAP via HTTPS	Service de panier d'achats international PurchaseStuff	226 000 € (300 000 \$)	4
COBOL Copybook via MQ, quatre copybooks différents	Service de gestion des commandes SendMeStuff	380 000 € (500 000 \$)	3
Protocole personnalisé basé sur TCP/IP	Provisionnement TurnMeOn	75 345 € (100 000 \$)	7

Ici, le service URGoodFort semble être un bon candidat à la virtualisation. Mais pourquoi celui-ci et pas le composant PurchaseStuff ? Tous deux utilisent la même technologie, à savoir SOAP via HTTPS. Tous deux ont un impact d'indisponibilité équivalent à 300 000 euros. Mais le service PurchaseStuff est quatre fois plus compliqué et sa création prendra plus de temps. En virtualisant le service URGoodFort, on obtient la même valeur (300 000 euros) moyennant un effort moindre. Vos propres chiffres et mises en œuvre seront bien sûr différents de ceux du tableau 4-1. Mais cette approche et ces exemples de données peuvent vous aider à cerner le processus de prise de décision concernant les éléments à virtualiser.

Généralement, plus une technologie ou un protocole de communication est courant, plus il est susceptible d'être directement compatible avec les outils de virtualisation. Certaines technologies, comme SOAP via XML, sont quasiment universelles (bien que des nuances existent). En revanche, certaines normes sectorielles, telles que SWIFT et FIX, sont moins communes.

Dans tous les cas, reprenez ceci : pour chaque outil, le retour sur investissement diffère selon que le protocole est ou non directement compatible, qu'il nécessite de la configuration ou qu'il requiert une extension pour être écrit dans l'outil à l'aide d'une API. Certaines entreprises, comme IBM, investissent lourdement dans la technologie dans le but de pouvoir modéliser n'importe quel protocole sans codage personnalisé.

Stratégies d'automatisation des tests

De nombreux projets d'automatisation des tests sont centrés sur l'automatisation de l'interface utilisateur. Toutefois, cette méthode peut s'avérer problématique (voir le chapitre 3).

Afin d'éliminer les retards de test liés à l'enregistrement et à l'exécution d'interfaces utilisateur fragiles, envisagez d'automatiser les tests au niveau de la couche API (interface de programmation d'applications) ou de service, au lieu de n'effectuer les tests que sur la couche interface. Nous avons déjà abordé cette couche dans le contexte de la virtualisation, et ce n'est pas un hasard. La virtualisation des services est compatible avec tous les types de test (manuels, automatisés, interfaces utilisateur, API, fonctionnels, performances). Cette approche présente également des avantages uniques et des synergies avec les tests de service :

- ✔ Les points d'intégration stratégiques sont testés tôt, avant même que l'interface utilisateur soit disponible.
- ✔ Les tests moins fragiles, comme les tests d'interface de service, sont plus stables que les tests d'interface utilisateur, en particulier après leur déploiement en production.
- ✔ Les tests peuvent jouer un rôle dans le contrat de service, en clarifiant les frontières au sein des équipes ou des organisations.
- ✔ Les mêmes tests permettent de vérifier les composants virtuels et la mise en œuvre de terrain.
- ✔ Les tests qui génèrent des requêtes et valident les réponses s'appuient sur les mêmes protocoles que les composants virtuels. Par conséquent, certaines tâches de création de composants virtuels et de test de service sont identiques, et ces tâches peuvent être réutilisées à l'aide d'outils extrêmement utiles.
- ✔ Il est plus facile d'isoler la cause des problèmes détectés sur la couche de services, car ceux-ci sont découverts plus près de la source du problème.

Mise en œuvre de la virtualisation des services par objectif et phase de test

Dans tous les cas, vos tests vont révéler des défauts. Toutefois, plus vous testez à un stade avancé du cycle de vie applicatif, plus le coût de résolution des défauts risque d'être élevé. Grâce à la virtualisation des services, vous pouvez repérer les erreurs et problèmes à chaque phase de test.

Phases de test

Le cycle de vie de développement applicatif comprend plusieurs types de test. Le processus de test est généralement proche du processus illustré à la figure 4-1.

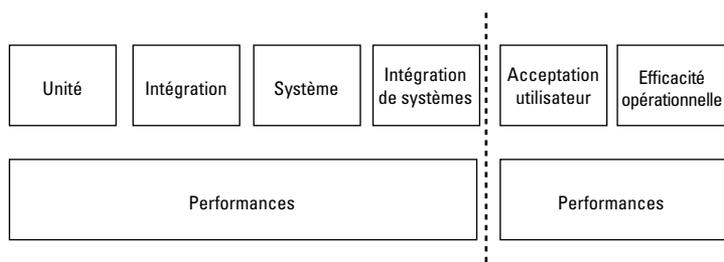


Figure 4-1: progression des phases de test au cours d'un projet.

L'approche classique consiste à réaliser les tests de manière séquentielle. Plusieurs types de test sont déployés successivement, dont :

- ✓ **Tests unitaires** : effectués par les développeurs pour tester de petites parties de code
- ✓ **Tests d'intégration** : modules de code testés ensemble
- ✓ **Tests système** : test de l'ensemble d'un logiciel par besoin système
- ✓ **Tests d'intégration système** : test des interactions entre systèmes
- ✓ **Tests d'acceptation utilisateur** : test du système par client selon des besoins spécifiques

✔ **Tests d'efficacité opérationnelle** : tests de la disponibilité des composants qui permettent à l'application de s'exécuter

Il n'est pas rare que les professionnels de la qualité attendent la fin des phases de test d'intégration système et d'acceptation client pour tester le système dans son ensemble. Dans ce cas, les défauts sont découverts à un stade avancé du processus de développement, là où ils sont coûteux à résoudre. Et les coûts de recherche de défauts lors des phases de test ultérieures peuvent faire boule de neige entre chaque phase. La figure 4-2 illustre ce concept.

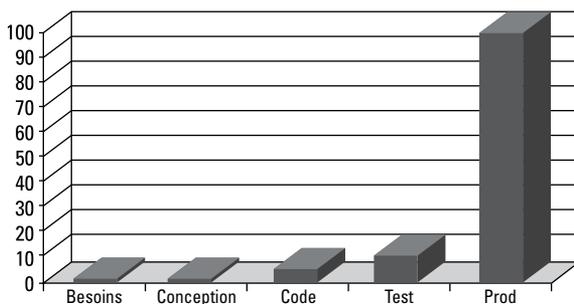


Figure 4-2: coût de résolution des défauts élevé en cours de production.



La virtualisation des services débloque le processus de test de bout en bout en supprimant les dépendances et en permettant de trouver les défauts plus tôt grâce aux avantages suivants :

- ✔ Les développeurs peuvent commencer à valider les intégrations à un stade beaucoup plus précoce du cycle de vie applicatif, au-delà des tests unitaires et en accroissant le niveau des tests réalisés lors du développement.
- ✔ Les testeurs peuvent débiter les tests d'intégration à un stade plus précoce et isoler les défauts dans des zones spécifiques de l'application, ce qui allège l'effort de résolution et prévient les problèmes d'intégration « explosifs » qui frappent souvent les projets.
- ✔ L'ensemble de l'équipe de développement peut bénéficier des avantages de la virtualisation des services en incluant celle-ci et les tests d'intégration dans le processus de production, et en profitant de retours immédiats sur la qualité des versions automatisées.



Si vous découvrez un défaut qui crée un blocage et ne peut pas être résolu immédiatement, vous pouvez utiliser la virtualisation des services pour simuler la fonctionnalité, afin que l'équipe puisse poursuivre les tests. Une fois le défaut résolu, vous pouvez revenir facilement au nouveau code source et poursuivre les tests avec la mise en œuvre réelle.

Le test des applications composites actuelles implique la validation d'un processus plus global constitué de plusieurs parties de fonctionnalités, qui s'appuie sur l'interconnectivité et l'interdépendance des applications. Tous les composants ne sont donc pas toujours prêts lorsque les équipes en ont besoin. Ils sont généralement rassemblés en vue des tests d'intégration système ou d'acceptation client peu avant la fin du développement, au début des tests de bout en bout.

Tests de performance

Les *tests de performance* sont un élément clé d'un processus de test global. Il n'est pas rare que les équipes reportent les tests de performance à des phases de test ultérieures, ou qu'elles les ignorent purement et simplement. Cette pratique peut causer des problèmes au niveau de la production et nuire à la réputation de l'entreprise. De nombreux problèmes de performance sont dus à des défauts d'architecture ou de conception applicative. La virtualisation des services permet de repérer ces défauts plus tôt, lorsqu'ils sont moins coûteux à résoudre par rapport à la phase de test d'acceptation client. Pourquoi ? Car il est possible de réaliser des tests de performance beaucoup plus tôt et de simuler des conditions très difficiles à rassembler dans des déploiements normaux.

Par exemple, le développement et le déploiement d'applications dans le cloud laissent peu de contrôle sur la latence réseau. Imaginez un instant la visibilité dont vous pourriez bénéficier si vous pouviez tester des conditions nécessitant une réponse immédiate ou retardée. La virtualisation des services vous offre cette possibilité.

Le chapitre 1 présente l'application fictive de l'entreprise Whiz Bang International. Lors des tests de sa nouvelle application, Whiz Bang International visait la réception de 95 % des réponses en quatre secondes maximum. La figure 4-3 illustre ce concept.

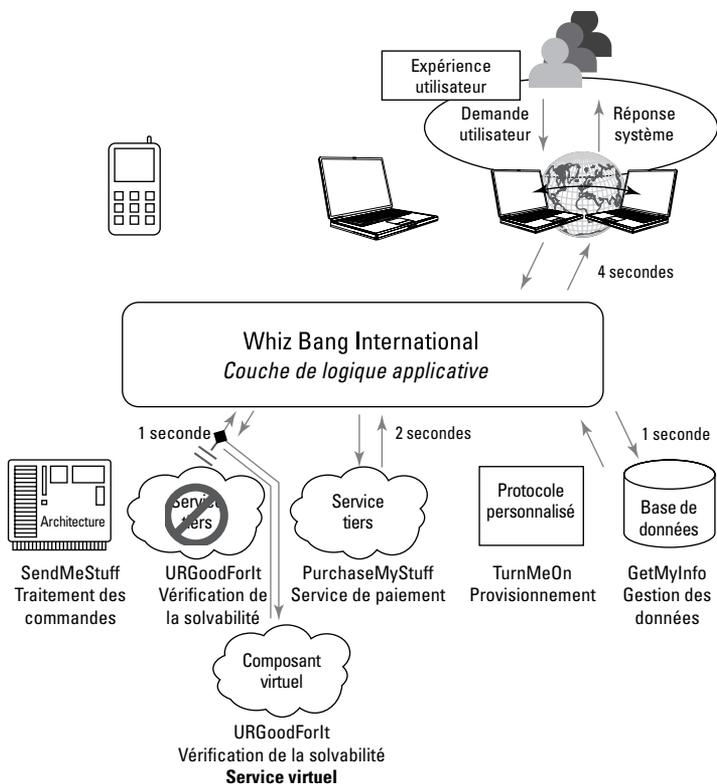


Figure 4-3: mesure du temps de réponse cumulé lors de la simulation d'un service tiers.

L'équipe avait déjà décidé de virtualiser le service de vérification de la solvabilité URGoodForIt alors que celui-ci était encore en phase de développement. Le temps de réponse de ce composant virtuel était fixé à une seconde, conformément au contrat de niveau de service conclu avec le fournisseur. La virtualisation des services a permis, dans ce cas, de réaliser des tests de performance plus précocement, car les deux autres composants étaient déjà prêts pour les tests, et le troisième pouvait être simulé. Lors de ces tests, l'équipe a découvert qu'un autre service (PurchaseMyStuff) présentait un problème. L'équipe chargée de PurchaseMyStuff a eu le temps d'améliorer les performances du service même avant que l'équipe du service URGoodForIt ne livre sa mise en œuvre.



Il existe de nombreuses façons intéressantes d'exploiter la virtualisation des services dans le cadre des tests de performance. Les voici plus en détail :

- ✓ **Modéliser le pire scénario.** Vous pouvez, par exemple, virtualiser certains services dorsaux et limiter leur débit, afin d'observer l'impact sur l'expérience utilisateur de l'application lorsqu'elle est soumise à une charge importante.
- ✓ **Rechercher le goulet d'étranglement suivant.** Dans ce scénario, vous pouvez, par exemple, virtualiser certains services avec un débit quasi illimité, afin d'observer à quels composants la prochaine contrainte sera liée en termes de performances globales du système. Ceci vous permet d'évaluer l'impact de l'amélioration de la mise en œuvre d'un service avant d'investir dans le développement.

Pour vous assurer que les systèmes réels fonctionnent correctement, les véritables composants doivent être validés. Vous pouvez désactiver tous les composants virtuels à un stade avancé des phases de test d'intégration système et d'acceptation client. La virtualisation des services ayant permis de réaliser des tests d'intégration précoces, vous devriez avoir peu de surprises.

Tests négatifs

Les tests négatifs visent à déterminer la manière dont une application ou un composant gère des conditions d'erreur et inattendues. Il peut par exemple s'agir d'entrées incorrectes (par exemple, vous tentez de déposer une somme négative sur un compte bancaire) ou de l'échec inattendu d'un service (un serveur plante). Dans tous les cas, l'application doit gérer correctement l'erreur. Toutefois, il peut être très difficile de reproduire certaines de ces conditions lorsque l'on teste des applications complexes. Il faut parfois une séquence compliquée d'événements et un ensemble complexe de données de test pour produire une erreur précise. Vous pouvez également ne pas être en mesure de placer le système dans un état particulier (en désactivant un service essentiel, par exemple), car cet état impacterait l'activité d'autres personnels partageant le même environnement de test. Il se peut également que vous ne sachiez pas comment reproduire une erreur (bug intermittent d'un composant spécifique, par exemple), tout en souhaitant vérifier que votre application peut la contourner.

La virtualisation des services facilite les tests négatifs, car elle permet de modifier le comportement d'un composant virtuel afin de produire une condition d'erreur spécifique. Différents testeurs (ou scénarios de test) peuvent observer des comportements différents sans impacter les autres personnels, en reconfiguration leur environnement de test avec différentes associations de composants virtuels.

Chapitre 5

Mise en œuvre de la virtualisation des services

.....

Dans ce chapitre

- ▶ Prendre le contrôle de votre architecture
 - ▶ Comprendre les différents services à virtualiser
 - ▶ Provisionner les composants virtuels
-

La virtualisation des services améliore et supporte la stratégie de gestion de la qualité de votre entreprise, quels que soient les types de test réalisés. Bien que vos processus de test soient susceptibles d'évoluer dans le temps, vous pouvez adopter la virtualisation des services progressivement, afin d'en retirer des avantages immédiats. Ce chapitre présente les différentes étapes qui vous permettront de créer vos premiers composants virtuels. La complexité des services à émuler (et des tests à effectuer) étant variable, nous vous proposerons un éventail de modèles comportementaux spécifiques aux composants virtuels. Nous donnerons également quelques critères auxquels une solution de virtualisation des services doit répondre. Enfin, nous expliquerons comment mener à bien le déploiement et le provisionnement de composants virtuels, et présenterons les bonnes pratiques de test. A la fin du chapitre, vous aurez en principe de bonnes connaissances de la création et de l'exécution de composants virtuels.

Comprendre votre architecture

Habituellement, lorsqu'ils testent les applications complexes actuelles, les professionnels de la qualité traitent tous les éléments qui se trouvent derrière l'interface utilisateur avec des tests fonctionnels. Cette approche leur permet de se centrer sur l'expérience de l'utilisateur final, élément important de la

stratégie de gestion de la qualité de l'entreprise. Toutefois, les tests d'interface utilisateur classiques présentent certaines limites face à la complexité croissante des environnements applicatifs. Pour mieux comprendre les problèmes susceptibles d'avoir un impact sur l'expérience utilisateur, les professionnels de la qualité doivent adopter une approche basée sur des tests structurels. Ils doivent connaître l'architecture sous-jacente afin de pouvoir virtualiser des services ou automatiser des tests d'intégration.

Différences entre la méthode de tests fonctionnels et la méthode de tests structurels :

- ✔ **Tests fonctionnels** : en quoi consistent les tests fonctionnels ?
Les testeurs d'interfaces utilisateur traitent le code d'application avec des tests fonctionnels, car ils n'ont besoin de connaître que les entrées et les sorties de l'application. Les transformations de données ou les calculs analytiques qui génèrent les sorties vues par les utilisateurs ne sont pas pertinents.
- ✔ **Tests structurels** : en quoi consistent les tests structurels ?
Les testeurs effectuent des tests structurels pour examiner et tester la structure interne ou le fonctionnement d'une application.

Pour effectuer des tests sur la couche service, les testeurs doivent réaliser des tests structurels. Pour qu'une application s'exécute correctement, les éléments interconnectés et interdépendants doivent fonctionner ensemble parfaitement. En d'autres termes, il faut acquérir une bonne compréhension des composants du système et des relations entre chaque composant. Pour virtualiser des services ou automatiser des tests d'intégration, il est indispensable de comprendre certains aspects de l'architecture du système qui s'exécutent sous l'interface utilisateur. Posez-vous les types de questions suivants :

- ✔ Quels sont les composants ?
- ✔ Comment les composants communiquent-ils entre eux ?
- ✔ A l'aide de quelle technologie les messages parviennent-ils d'un point à un autre ?
- ✔ Quels sont les protocoles utilisés (détails des messages, par exemple) ?
- ✔ Quels sont les points finaux (endpoints) ?

Le chapitre 1 présente un exemple associé au secteur du commerce électronique. Nous vous invitons à consulter la figure 1-1. Cet

exemple comprend une couche de logique applicative, une couche de présentation (appareils mobiles, ordinateurs portables) et des services dorsaux (gestion des commandes, vérification de la solvabilité, etc.). Certains services dorsaux sont développés en interne, tandis que d'autres sont fournis par des tiers. La couche intermédiaire s'appuie sur JMS pour la publication de messages. Les messages sont mis en forme en XML selon un schéma spécifique (fichiers .xsd).



Votre système est susceptible de contenir d'autres composants. Il ne s'agit ici que de quelques exemples. Il existe des centaines de protocoles et de méthodes d'acheminement. Toutefois, la virtualisation des composants est plus facile si votre logiciel de virtualisation des services est directement compatible avec les technologies que vous utilisez.

Communication entre les composants

Pour virtualiser un composant, il est indispensable de comprendre comment celui-ci communique avec d'autres composants au sein de l'application. Il peut s'agir de services Web, d'intergiciels (middleware), de bases de données, etc. Les détails de la communication sont définis dans un modèle de votre environnement applicatif. Des messages (unités d'information) sont envoyés et reçus par les composants selon plusieurs schémas de communication :

- ✓ **Requête/réponse** : ce type de communication est défini par un schéma de comportement où chaque requête génère une réponse. Le principe requête/réponse est le fondement de la communication de données sur Internet. Les services Web et le protocole HTTP s'appuient sur ce type de communication. Par exemple, lorsque vous saisissez une URL dans un navigateur, celui-ci envoie une requête GET HTTP à un serveur Web associé à l'URL. Le serveur répond en renvoyant le contenu de la page (souvent au format HTML) dans le navigateur. Chaque requête génère une réponse synchrone.
- ✓ **Publication/souscription** : ce modèle est légèrement différent. Ici, un composant publie une requête afin qu'elle soit traitée dans une file d'attente de messages. Un autre composant s'abonne à la file d'attente de messages, surveille les messages entrants et traite les requêtes de la file d'attente. Lorsqu'une réponse au demandeur d'origine est requise, la même technique est employée dans une autre file d'attente de messages. Dans ce cas, le composant de traitement publie

un message de réponse dans une file d'attente à laquelle le composant d'origine est abonné. Les requêtes et les réponses sont asynchrones. Par exemple, l'intergiciel orienté messages IBM WebSphere MQ utilise ce type de communication.

Prenons l'exemple d'un processus métier dont la fonction consiste à valider une déclaration de sinistre dans le cadre d'une assurance. Le processus de validation peut nécessiter le rassemblement de données supplémentaires pour traiter la déclaration. Or, le délai d'obtention de ces informations peut être plus long que la durée pendant laquelle le service demandeur peut raisonnablement attendre. Dans ce cas, la requête est publiée et le demandeur est libre d'effectuer d'autres activités pendant le temps nécessaire au traitement. Une fois la validation traitée, le résultat est publié auprès du demandeur.

- ✓ **Requête/résultat** : ce schéma de communication se caractérise par un schéma de comportement dans lequel une demande de requête génère une réponse synchrone sous forme d'ensemble de résultats. Les modèles de programmation de bases de données relationnelles, par exemple JDBC, s'appuient sur ce modèle pour exécuter des requêtes. Il est important de faire la différence avec le modèle requête/réponse car, contrairement au protocole HTTP, ce modèle écarte toutes les communications réseau, nécessaires ou non.

Acheminement des messages

Les protocoles de communication s'appuient sur diverses méthodes d'acheminement pour l'envoi de messages et la réception de réponses. Ces méthodes de communication entre composants définissent la manière dont les messages passent d'un composant à l'autre. Voici quelques exemples :

- ✓ **Hypertext Transfer Protocol (HTTP/https)** : ce protocole est fondamental pour la communication de données sur le Web. Il s'agit d'un protocole requête/réponse basé sur TCP/IP.
- ✓ **Simple Object Access Protocol (SOAP)** : SOAP est un protocole simple basé sur le langage XML, qui permet aux applications d'échanger des informations via HTTP.
- ✓ **Enterprise Service Bus (ESB)** : ESB est un composant d'architecture qui supervise et contrôle la communication entre services métier. Exemples d'ESB : IBM WebSphere MQ, Software AG WebMethods Integration Server et l'application

open source Mule. Ces systèmes sont souvent utilisés via une API standard telle que JMS (Java Message Service), par exemple.

- ✓ **Java Database Connectivity (JDBC)** : JDBC est une API qui utilise le langage SQL pour associer des bases de données relationnelles et d'autres sources de données.

Normes relatives aux messages

De nombreux secteurs ont élaboré des normes concernant les détails des messages spécifiques à leur domaine. Ces normes définissent les spécifications du *schéma* des messages, c'est-à-dire les règles qui régissent leur format (structure des champs, types, valeurs, etc.). Les normes permettent d'assurer l'acheminement rapide et précis des messages entre les composants. Les normes de message se prêtent parfaitement à la virtualisation des services. Voici quelques exemples :

- ✓ **Service Web défini à l'aide d'un fichier WSDL (Web Services Description Language)** : les services Web sont définis à l'aide de descriptions WSDL. La description WSDL permet de décrire les points finaux (endpoints) et les messages associés, quel que soit le schéma de message déployé.
- ✓ **SWIFT (Society for the Worldwide Interbank Financial Telecommunication) et FIX (Financial Information eXchange)** : les protocoles SWIFT et FIX sont spécifiques au secteur des services financiers. Ils offrent une méthode normalisée et sûre pour transférer et communiquer des informations financières.
- ✓ **Health Level 7 (HL 7)** : la norme HL 7 est spécifique aux messages des environnements du secteur de la santé.

Recherche des points finaux (endpoints)

Les applications doivent savoir comment accéder aux services dont elles dépendent (points finaux). Exemples de points finaux : URL de service Web, point final JMS, chaîne de connexion JDBC. Certaines technologies de virtualisation des services surveillent et gèrent la communication entre composants, sans que l'équipe ait à modifier les points finaux au sein de l'application. Cette méthode simplifie considérablement le processus de configuration d'environnements de test pour tirer parti des composants virtuels.

Définition de composants virtuels

La première étape de la définition de composants virtuels consiste à modéliser l'architecture avec suffisamment de précision pour mettre en évidence les frontières où les composants virtuels peuvent être introduits. Il est bon d'utiliser un outil qui représente visuellement les services et les composants du système testé, et les dépendances entre chacun de ces éléments. Si l'outil distingue la vue logique (composants, services et protocoles) de la vue physique (URL des points finaux, noms d'hôte, adresses IP, etc.), c'est encore mieux. Il peut ainsi supporter plusieurs environnements exécutant les mêmes composants. Ceux-ci partagent une vue logique, mais chaque environnement présente des caractéristiques physiques spécifiques. Par exemple, les adresses IP des serveurs de production sont différentes de chaque environnement de test/préproduction.

Synchronisation avec des sources externes

Il se peut qu'un grand nombre des informations nécessaires pour dresser ce panorama se trouve déjà parmi vos actifs d'environnement et de développement. Vous pouvez éventuellement vous synchroniser avec ces sources externes pour renseigner les vues de l'architecture. Exemples : fichiers WSDL, environnements intergiciels (middleware).

Enregistrement de services existants

Vous devez définir le comportement de vos composants virtuels. L'enregistrement de services existants est un excellent moyen de capturer un grand nombre d'informations (comportement, données, etc.) nécessaires à la création très rapide de scénarios de test et de composants virtuels. Cette méthode contribue également à décider quels éléments simuler.

Certains protocoles, par exemple, sont vastes et contiennent des dizaines de messages. En pratique, il se peut que votre application n'utilise pas tous ces messages. Pour comprendre ce qu'il se passe réellement en arrière-plan, vous pouvez enregistrer une session dans un environnement de production ou de préproduction, puis utiliser ces nouvelles informations pour définir l'étendue de votre projet de virtualisation des services. Il en va de même pour les données.

L'enregistrement présente de nombreux avantages :

- Il permet de repérer les messages partagés et leur format. Supposons que le service de vérification de la solvabilité propose une méthode de demande de contestation, que votre application n'utilisera probablement jamais. Dans ce cas, il n'est pas nécessaire que le composant virtuel modélise le comportement correspondant.
- Il permet de déterminer la plage de données utilisée, par exemple si le service de vérification de la solvabilité renvoie une note de solvabilité, et que votre service virtuel a besoin de connaître la plage de réponses raisonnable à associer à cette valeur.
- L'enregistrement permet de comprendre les schémas d'échange de messages. Il s'agit de séquences de messages. Situation fréquente : vous exécutez une requête, vous obtenez une réponse et un élément de cette réponse est intégré dans la requête suivante. Exemple : vous utilisez un service Web pour placer un article dans un panier d'achats électronique et vous recevez en retour l'ID du panier. Cet ID permettra ensuite d'exécuter une autre requête pour afficher le contenu du panier.

Quels sont les points importants à prendre en compte pendant l'enregistrement ? Cette section fournit les éléments de réponse.

Que faire si vous ne pouvez pas effectuer d'enregistrement ? Pour commencer à utiliser la virtualisation des services, vous n'avez pas besoin de réaliser d'enregistrements. Vous pouvez créer des composants virtuels à partir des spécifications de conception préalablement au développement complet du service.



Rééchantillonnage du comportement des composants virtuels

Les outils de virtualisation des services prennent en compte les entrées et les sorties des composants. L'exemple ci-après illustre comment rééchantillonner un composant virtuel.

Prenons l'exemple d'un service Web qui permet de rechercher le code postal de lieux aux Etats-Unis. La requête contient deux paramètres : la ville et l'état. La réponse est une chaîne de cinq chiffres. La figure 5-1 illustre un exemple d'enregistrement du trafic vers ce service.

Entrées		Sorties
Ville	Etat	Code postal
Port Jefferson	NY	11777
Austin	TX	78701
Littleton	MA	01460

Figure 5-1: exemple d'enregistrement du trafic pour un service de recherche de code postal.

L'enregistrement des entrées (ville et état) et des sorties (code postal) fournit les informations nécessaires pour le rééchantillonnage ou la capture rapide du comportement des composants virtuels.

Comprendre le mécanisme d'enregistrement

Vous devez vous assurer que la technologie d'enregistrement n'interfère pas avec le déploiement de votre application et que les composants sont faciles à introduire dans votre environnement. Par exemple, le processus de test peut subir des ralentissements causés par des ajustements manuels visant à assurer la communication entre le composant virtuel et le reste du système. Vous ne pouvez pas vous permettre de passer du temps à reconfigurer manuellement le déploiement de votre application avant d'utiliser le composant virtuel. Par ailleurs, si un enregistrement nécessite la modification du code de l'application, cette modification peut nuire à votre productivité et à votre réussite avec la virtualisation des services.

Comportement des composants virtuels

Les composants virtuels présentent un éventail de comportements, dont :

- ✓ **Simple** : les comportements simples sont déterministes. Le composant virtuel émule un service Web en renvoyant la même réponse à chaque fois pour une entrée donnée. Prenons l'exemple d'une application du secteur des voyages professionnels, Hotel Finder, qui communique avec plusieurs services pour rechercher des hôtels. L'application exécute une requête afin de rechercher un hôtel. Lors du test de cette application, à chaque fois que vous recherchez un hôtel, vous recevez la même réponse : ABC Hotel. C'est suffisant pour certains scénarios de test.
- ✓ **Non déterministe** : ici, la variabilité est plus large. Exemple : le service reçoit la cote boursière de l'entreprise Big Company. Vous souhaitez que le composant virtuel génère un chiffre différent à chaque fois que vous sollicitez le service. Dans le cadre des tests, peu importe le chiffre, tant qu'il se trouve dans une plage raisonnable. Dans ce cas, il n'y a pas de logique applicative, mais la variation permet d'effectuer des tests plus réalistes.
- ✓ **Basé sur des données** : ce comportement étend la richesse des données issues du composant virtuel. Les données

d'entrée et de sortie sont spécifiées dans une source de données externe, par exemple une feuille de calcul ou une base de données. Les entrées du service servent à rechercher les sorties correspondantes. Prenons l'exemple d'un service de recherche d'hôtel dans lequel vous devez indiquer douze villes et trois à quatre hôtels dans chaque ville. Il faudra environ 42 (12 x 3½) lignes de données pour réaliser un tableau de réponses. Cette diversité peut être nécessaire pour s'assurer que le service communique correctement avec d'autres services.

- ✓ **Basé sur un modèle/contextuel (stateful)** : dans un service *contextuel (stateful)*, l'état d'une requête peut changer sur le serveur. Cet état doit être géré et pris en compte dans les réponses aux requêtes suivantes. Ces changements peuvent être modélisés avec un diagramme de transition d'état. Supposons, par exemple, que vous souhaitez ajouter un article au panier d'achats d'une application de commerce électronique. Vous exécutez une requête auprès d'un service pour ajouter l'article au panier, et le composant virtuel doit mémoriser le contenu du panier pour répondre correctement aux requêtes qui examinent le panier, le paiement, etc. Il vous faut un outil dans lequel ce comportement est facile à spécifier.
- ✓ **Comportemental** : cette catégorie concerne les composants qui ne suivent pas les autres schémas de comportement plus classiques. Il vous faut un moyen simple d'ajouter un comportement arbitraire à vos composants virtuels par l'intermédiaire de scripts ou de code.



Voici une checklist des points importants pour gérer les données de test des composants virtuels :

- ✓ Les données extraites des environnements de production doivent être présentées de façon à être simples à utiliser.
- ✓ Les données doivent être capturées pendant l'enregistrement selon une méthode qui permette de créer facilement un service virtualisé.
- ✓ Les données doivent pouvoir être anonymisées ou masquées si nécessaire. Exemple : vous virtualisez un service qui extrait des dossiers médicaux. Vous avez besoin de données réelles pour votre test. Vous devez anonymiser/adapter ces données pour garantir la sécurité et la conformité.
- ✓ Vous devez tester plusieurs types de conditions de frontière. Par exemple, est-ce qu'un panier d'achats vide sera suffisant

pour votre environnement de test, ou avez-vous besoin d'y ajouter des articles ?

- Externalisez les données des composants virtuels afin de pouvoir procéder facilement à des mises à jour. L'auteur d'un nouveau scénario de test pourra ajouter facilement des données pertinentes dans une feuille de calcul.

Définir l'outil adapté

Lors du choix d'une solution de virtualisation des services, il vous faut déterminer les besoins des différents rôles au sein de votre entreprise. Tous les personnels n'ont pas besoin de définir des composants virtuels, mais de nombreux testeurs doivent pouvoir y accéder dans leurs environnements de test. Pour garantir le déploiement simple et efficace de la virtualisation des services, votre solution de virtualisation doit supporter plusieurs éléments clés ou fournir les fonctionnalités associées à ces éléments :

- Méthode d'observation et d'enregistrement des conversations de messages
- Outil de création et de gestion de composants virtuels
- Environnement d'hébergement de composants virtuels
- Méthode simple permettant aux testeurs de configurer leur environnement avec des composants virtuels

Vous devez disposer d'une solution de services virtuels suffisamment souple pour basculer entre le composant réel et les composants virtuels lors des tests. Exemple : 90

% de vos tests s'appuient sur un composant virtuel (Hotel Finder, par exemple). Si les tests réussissent, les 10 % restants nécessitent de basculer vers le véritable service (Expedia, par exemple). Les deux ensembles doivent utiliser la même version, afin de prévenir le risque de modifications entre différentes versions. D'autres solutions, par exemple les éléments de remplacement ad hoc écrits par des développeurs, nécessitent la modification, la reconstitution et le redéploiement de l'application. Grâce à des outils tels qu'IBM Rational Test Workbench et IBM Rational Test Virtualization Server, il est inutile de modifier le code de l'application pour basculer les appels de service entre le composant réel et un composant virtuel. Il suffit de cliquer sur un bouton dans le panneau de configuration. Vous pouvez même définir des règles dans votre composant virtuel pour répondre de manière sélective à certaines requêtes entrantes, tout en autorisant d'autres à pénétrer dans un système réel. Ceci est particulièrement utile pour limiter l'accès au système réel pour des raisons de coûts ou de performance. Mais certains comportements sont difficiles à simuler.

Provisionnement des services virtuels

Le provisionnement de services virtuels nécessite deux capacités fondamentales :

- ✓ Un moyen de développer des services virtuels dans un environnement personnel sur votre bureau
- ✓ Un environnement dédié à la virtualisation des services partagé au sein de l'équipe de développement

Vos développeurs et vos testeurs ont besoin d'une infrastructure partagée pour l'hébergement des composants virtuels. Par ailleurs, il vous faut gérer plusieurs environnements en parallèle avec différentes associations de composants réels et virtuels. Chaque environnement associe un ensemble de variables, issues de la vue logique du système, à des ressources physiques et virtuelles spécifiques (identifiées par des URL, des adresses d'hôte, des ports ou d'autres paramètres de connexion). En créant plusieurs environnements (environnements développeur privés, environnements de test d'intégration système et environnements d'acceptation client, par exemple), vous pouvez tester plusieurs configurations à chaque phase du cycle de vie du produit. La figure 5-2 illustre l'association des services de Whiz Bang (voir le chapitre 1) à trois environnements différents.

Services	Environnements		
	Dév. privé	Intégration système (SIT)	Acceptation client (UAT)
SendMeStuff	Virtual-SMS	192.168.12.15	192.168.1.10
UR GoodForIt	Virtual-GFI	Virtual-GFI	http://sv4d.co/goodforit
PurchaseMyStuff	Virtual-PMS	http://sv4d/test/purchaSe	http://sv4d.co/purchase
TurnMeOn	Virtual-TMO	Virtual-TMO-2	Virtual-TMO-2
GetMyInfo	Virtual-DB	jdbc://192.168.12.27/infodb	jdbc://192.168.1.20/infodb

Figure 5-2: provisionnement de services (réels et virtuels) dans plusieurs environnements de test.

A ce stade, vous devez être en mesure de définir le trafic acheminé vers un service réel, et le trafic acheminé vers des services virtuels au sein de vos environnements. Une fois votre environnement établi, vous devez être capable d'exécuter tous vos tests sans faire la différence entre les services réels et virtuels. Votre outil de virtualisation des services doit supporter le changement d'acheminement sans que l'équipe ait à modifier ni le code, ni la configuration de l'application. Vous pouvez ajuster les composants virtuels tout au long de ce processus.

Exécution des tests et évaluation des résultats

La virtualisation des services permet d'effectuer des tests facilement. Grâce à cette technologie, tous vos tests (manuels, automatisés, d'interface utilisateur et d'intégration) peuvent être exécutés de la même manière que dans un environnement non virtualisé. Idéalement, l'outil servant à créer vos services virtuels doit intégrer une fonction d'exécution de tests et de création de rapports sur

l'exécution des tests. L'outil sera d'autant plus utile s'il est intégré dans une solution de gestion de la qualité, afin que vous puissiez gérer et créer des rapports sur les résultats d'exécution dans un référentiel centralisé accessible par tous. Les bonnes pratiques de gestion de la qualité des environnements de test non virtualisés ont également tout leur intérêt en environnement de test virtualisé.

Chapitre 6

Mesure du retour sur investissement

.....

Dans ce chapitre

- ▶ Elaboration d'une étude de rentabilité
 - ▶ Avantages financiers de la virtualisation des services
 - ▶ Réussir grâce à votre solution de virtualisation des services
-

Dans la première partie de cet ouvrage, nous avons vu comment la virtualisation des services peut contribuer à réduire les coûts, améliorer la qualité et limiter le risque lié à la production d'applications complexes de haute qualité. A présent, nous allons voir comment réaliser une étude de rentabilité sur la virtualisation des services.

Ce chapitre présente certains des avantages fondamentaux de la virtualisation des services et leur potentielle contribution au retour sur investissement. Nous aborderons également plusieurs points importants qui vous permettront de choisir soigneusement vos outils et d'investir intelligemment.

Elaboration d'une étude de rentabilité

L'objectif d'une étude de rentabilité consiste à amener les dirigeants à comprendre les coûts et avantages d'un projet avant d'investir. Dans le cas de la virtualisation des services, l'étude de rentabilité vise à déterminer si la mise en œuvre de cette technologie est économiquement viable pour votre entreprise, et à évaluer sa valeur par rapport à des alternatives d'investissement. Vous devez évaluer dans quelle mesure la virtualisation des services améliore votre processus de test et, de manière plus globale, comment cette technologie améliore vos processus de gestion du cycle de vie applicatif. Les résultats étant mesurables, rapides et tangibles, votre

étude de rentabilité doit en principe contribuer à convaincre les dirigeants de faire de la virtualisation des services une priorité en matière d'investissement.

Votre étude de rentabilité doit intégrer trois éléments fondamentaux :

- ✓ Argumentaire exposant les raisons pour lesquelles vous avez décidé de mettre en œuvre la virtualisation des services
- ✓ Estimation des coûts de la mise en œuvre
- ✓ Estimation des avantages de la mise en œuvre



Etablissez votre propre liste d'avantages escomptés pour mesurer votre retour sur investissement. Ces avantages pouvant être très nombreux, en voici quelques-uns pour commencer :

- ✓ Accroissement de la rapidité des équipes
- ✓ Réduction des coûts liés à la qualité
- ✓ Diminution des risques liés aux projets
- ✓ Niveau de test plus élevé
- ✓ Efficacité de résolution des défauts

Pourquoi choisir la virtualisation des services ?

Quelle est l'ampleur de l'inactivité liée à l'incapacité de votre équipe d'accéder aux systèmes ? Quel est le coût supporté pendant que les équipes de développement attendent de pouvoir commencer les tests ? La virtualisation des services est importante, car elle contribue à résoudre les points les plus problématiques du processus de test. Vous devez comprendre les coûts attribuables aux retards lors des tests. Reportez-vous à l'analyse interne de votre entreprise sur ce thème pour élaborer un argumentaire en faveur de la mise en œuvre de la virtualisation des services aux niveaux technique et commercial.

Par ailleurs, avant de planifier vos tests, déterminez la fréquence à laquelle vous avez besoin d'accéder à des interfaces tierces. Déterminez également le coût de l'accès aux environnements de test de fournisseurs de services tiers. Cette partie de l'étude de rentabilité commence à justifier l'intérêt de la virtualisation des services. Dressez un état des lieux de votre environnement de test et identifiez les problèmes majeurs à résoudre. Faites valoir de manière convaincante l'importance de simplifier le processus de

développement logiciel et la manière dont cet objectif peut être atteint.

Reportez-vous éventuellement au chapitre 4 pour élaborer la première partie de votre étude de rentabilité. Vous y trouverez une série de questions pour évaluer les besoins en matière de virtualisation des services.

Estimation des coûts de mise en œuvre de la virtualisation des services

Les coûts de mise en œuvre de la virtualisation des services sont relativement faciles à identifier. Ils comprennent le coût d'acquisition des licences logicielles, le coût du matériel nécessaire pour héberger la solution (si vous n'utilisez pas de matériel existant) et les coûts de mise en œuvre associés. Vous devez également prendre en compte le temps de formation de l'équipe de développement aux nouveaux outils. Les estimations de ces dépenses varient selon la solution logicielle choisie, et selon la simplicité d'utilisation de cette solution.

Estimation des avantages de la mise en œuvre de la virtualisation des services

Cette partie de l'étude de rentabilité expose les coûts que l'entreprise peut éviter grâce à la virtualisation des services. Elle explique également comment la virtualisation des services peut accroître l'efficacité et la productivité. La section suivante présente des exemples d'avantages potentiels sur le terrain.

Quantification des avantages

Les tests constituent une part importante du coût de développement d'applications. L'élimination de certains coûts grâce à la virtualisation des services permet d'obtenir un avantage financier conséquent. L'accélération des tests et l'amélioration de la qualité des applications sont également des sources d'avantages. Toutefois, bien que ces avantages puissent être considérables, certains sont difficiles à quantifier pour le calcul du retour sur investissement. Par exemple, comment mesurer l'avantage de la mise sur le marché d'un produit innovant avant vos concurrents ? Quelle est la valeur monétaire de la réduction de la durée d'un cycle de test ?

Quelles que soient la manière dont vous réalisez vos tests et la complexité de vos applications, vous avez la possibilité d'éliminer rapidement de nombreux coûts de test grâce à la mise en œuvre d'une solution de virtualisation des services efficace. Si vous faites partie des nombreuses entreprises qui rencontrent actuellement des difficultés pour tester des applications complexes ou si vous dépendez de services tiers, attendez-vous à bénéficier d'un retour sur investissement significatif !

Cette section présente certains facteurs majeurs à mesurer pour estimer le retour sur investissement.

Elimination ou réduction des coûts liés aux environnements de test classiques

La complexité des applications composites actuelles nécessite des environnements de test modernes capables d'exécuter plusieurs serveurs hébergeant diverses applications logicielles (serveurs HTTP, serveurs d'applications, intergiciels (middleware), bases de données, etc.) installées sur divers systèmes d'exploitation. Les coûts de configuration et de gestion de ce type d'environnement comprennent les coûts matériels et logiciels initiaux, mais aussi les coûts d'administration et de maintenance continue des serveurs.

Prenons l'exemple d'un environnement de test nécessitant cinq serveurs : serveur HTTP, serveur d'applications, serveur d'intergiciels, serveur de bases de données et système d'ERP. Quatre des serveurs induisent un coût de 5 000 euros par mois, et le système d'ERP coûte 20 000 euros par mois. L'équipe envisage de mettre en œuvre la virtualisation des services pour émuler les systèmes de base de données et d'ERP lors des premières phases de test, espérant ainsi réaliser 25 000 euros d'économies par mois.

Vous pouvez également éviter certains coûts élevés liés à l'achat et à la maintenance d'un environnement mainframe en optant pour un environnement hébergé. Dans ce cas, vous paierez des frais d'accès mensuels variables selon le nombre de MIPS (million d'instructions par seconde) consommé. Quelle que soit la manière dont vous payez votre mainframe, vous dépendez sans doute de logiciels hébergés dans cet environnement, et vous devez soutenir les coûts associés aux tests.

Et si vous pouviez réduire d'au moins de moitié les coûts de test mensuels liés au mainframe ? Supposons que le coût d'un environnement mainframe pour effectuer des tests s'élève à

15 000 euros par mois. Dans ce cas, la réduction de moitié de ce coût génèrerait une économie annuelle de 90 000 euros.

Examinons maintenant les frais d'accès à la demande facturés par les fournisseurs tiers, par exemple pour accéder à leurs services dans un environnement de test dédié. En optant pour la virtualisation des services, vous pouvez réduire considérablement ces frais d'accès. Supposons, par exemple, que les frais d'accès aux services tiers hébergés s'élèvent à 30 000 euros par mois. En virtualisant les services et en effectuant vos tests à partir de composants virtuels, vous pouvez réduire ce montant de 80 %. Cela équivaut à une économie de 24 000 euros par mois, soit potentiellement 300 000 euros par an.

Durée de provisionnement des environnements de test

L'emploi de ressources coûteuses pour configurer, désassembler et réinitialiser les environnements de test accroît considérablement le coût des tests. Avec des coûts moyens en équivalent temps complet de 104 000 euros pour les testeurs et de 135 000 euros pour les développeurs, il est possible de réaliser une économie de 80 000 à 110 000 euros par ressource et par an en réduisant de 80 % l'effort de gestion des environnements de test. Au lieu de gérer les environnements de test, les testeurs et les développeurs peuvent se recentrer sur la qualité des tests.

Cette nouvelle efficacité libère vos équipes hautement qualifiées, qui peuvent alors consacrer davantage de temps à des activités génératrices de revenus. En adoptant la virtualisation des services, vous pouvez gagner en efficacité et mettre plus rapidement sur le marché des logiciels d'excellente qualité.

Recherche et résolution des défauts en amont du processus de développement

L'un des facteurs clés du calcul du retour sur investissement repose sur la réduction du nombre de défauts logiciels. Les défauts sont coûteux à résoudre et peuvent exercer un impact négatif réel sur la satisfaction client. Retenez les points suivants :

- ✓ Les défauts logiciels sont le problème numéro un provoqué par l'homme dans les organisations de développement logiciel.

- ✓ La détection précoce des défauts au cours du processus de développement logiciel facilite leur résolution et réduit les coûts qui en découlent.
- ✓ Pour repérer les défauts plus précocement, il faut commencer à tester plus tôt.

Exemple de calcul de retour sur investissement : supposons que vous découvriez 100 défauts lors des tests d'acceptation client d'une version, pour un coût de résolution de 200 000 euros, soit 2 000 euros par défaut. En production, le coût par défaut est multiplié par dix, soit 20 000 euros par défaut. Quinze défauts sont découverts en production pour un coût de résolution de 300 000 euros, ce qui élève le coût total de résolution à 500 000 euros. Si le nombre de défauts trouvés lors des tests d'acceptation client et en production était réduit de 70 %, le potentiel d'économie serait d'au moins 350 000 euros.

Ce chiffre ne prend pas en compte l'impact financier potentiellement considérable du manque de satisfaction client ou de la perte de revenus. L'amélioration de la satisfaction client peut accroître encore davantage les économies réalisées. Grâce à la mise en œuvre d'une solution de virtualisation des services, les testeurs peuvent commencer à tester plus tôt, et isoler les défauts de sorte à les résoudre plus rapidement et moyennant un coût moindre.

Le calcul des chiffres à partir de ces trois facteurs quantifiables révèle des économies considérables, qui font boule de neige à chaque version logicielle testée avec la virtualisation des services. Deux autres facteurs doivent être pris en compte lors de l'élaboration de l'étude de rentabilité. Ces facteurs ne sont pas simples à quantifier, mais peuvent avoir un impact important. Il s'agit de l'accélération de la mise sur le marché et de l'amélioration des processus.

Accélération de la mise sur le marché

En effectuant des tests plus efficaces plus précocement, vous pouvez innover plus vite et mettre vos produits sur le marché avant vos concurrents. Réfléchissez au montant des revenus que vous pourriez générer en étant le premier à commercialiser une solution innovante.

Amélioration des processus

Les inefficacités de votre processus de test sont très coûteuses. Pensez aux dépenses de main-d'œuvre et aux coûts associés qui partent en fumée pendant que vos testeurs attendent le déploiement de logiciels dépendants ou la disponibilité d'un environnement de test partagé. Chaque minute d'inactivité évitée accroît le retour sur investissement de la virtualisation des services.



Dans certaines entreprises, l'inactivité des testeurs est si élevée qu'elle réduit considérablement la productivité des équipes de développement et de déploiement logiciels. Dans certains cas, ce temps d'attente représente jusqu'à 30 % du temps de travail des testeurs, soit une dépense de 22 500 euros consacrée à du temps non productif pour un salaire annuel de 75 000 euros. Grâce à la virtualisation des services, les testeurs testent, et n'attendent pas.

Choix d'une solution

Avant de choisir un logiciel de virtualisation des services, vous devez comprendre les clés de la réussite avec cette solution et comment celle-ci répond aux besoins spécifiques de votre entreprise. Avez-vous pensé aux coûts supplémentaires à supporter si ce logiciel ne fournissait pas les capacités dont vous avez besoin ? Si la solution choisie n'apporte pas les fonctionnalités nécessaires, vous n'atteindrez pas le retour sur investissement espéré.

Pour comparer plusieurs solutions, posez-vous les questions suivantes :

- ✔ La solution est-elle facile à utiliser ?
- ✔ Combien et quel niveau de formation faut-il envisager ?
- ✔ La solution permet-elle de créer des composants virtuels à partir d'enregistrements ou de spécifications de conception ?
- ✔ La solution permet-elle de réaliser des tests manuels, automatisés (tests d'intégration et fonctionnels) et de performance ?
- ✔ La solution permet-elle de créer des composants virtuels pour tester différents scénarios (« happy path », « alternative flow » et tests négatifs) ?
- ✔ La solution permet-elle de créer des composants virtuels pour tester des conditions de simulation (« what-if ») ?

- ✔ La solution permet-elle de déployer et de gérer rapidement des services virtualisés par l'intermédiaire d'une console d'administration ?
- ✔ La solution permet-elle de basculer entre les systèmes de terrain et les services virtuels sans avoir à reconfigurer le déploiement de l'application ?
- ✔ La solution est-elle capable de gérer la complexité de votre environnement applicatif (séquences de réponses corrélées et guidées par les données, émulation contextuelle de bases de données, etc.) ?
- ✔ La solution vous permet-elle de planifier et d'exécuter automatiquement des tests liés à des composants virtuels dès la mise à disposition d'une nouvelle version d'une application ?
- ✔ Pouvez-vous créer, modifier et déployer des composants virtuels sans que vos équipes aient à acquérir de nouvelles compétences de programmation ?
- ✔ La solution permet-elle de partager et de réutiliser facilement les composants virtuels parmi les équipes ?
- ✔ La solution permet-elle à vos équipes de développer en environnements parallèles ?
- ✔ La solution est-elle capable d'évoluer pour accueillir de larges équipes à mesure de la croissance de votre entreprise ?
- ✔ La solution requiert-elle un engagement de services professionnels sur le long terme ?

La technologie étant en perpétuelle évolution, veillez à choisir un fournisseur de logiciels qui s'engage à proposer régulièrement de nouvelles fonctionnalités pour éviter tout risque d'obsolescence. Pour accélérer le processus d'adoption et maintenir votre productivité, vérifiez que votre fournisseur propose un excellent support et la possibilité d'entrer en relation avec des techniciens compétents.

Chapitre 7

Dix points clés pour réussir avec la virtualisation des services

.....

Dans ce chapitre

- ▶ Partir du bon pied
 - ▶ Donner le contrôle aux testeurs
 - ▶ Accroître la productivité par l'acquisition de compétences
-

La virtualisation des services permet d'accroître considérablement l'efficacité des tests logiciels. Cette technologie vous invite à repenser votre approche des tests et à adopter des processus optimisés, afin de vous libérer des goulets d'étranglement des tests et de vous centrer sur la créativité et l'innovation. Vous gagnez du temps : plus besoin d'élaborer des environnements de test complexes, ni d'effectuer des tests en permanence tout au long du cycle de vie de développement. Vous explorez des scénarios de test évolués plus précocement, sans avoir à réinventer la roue. Le retour sur investissement de la virtualisation des services est rapide, et les avantages pour vos équipes de développement et votre entreprise sont réels.

Repenser votre approche des tests

La virtualisation des services est compatible avec toutes les méthodologies de test. Votre objectif doit être d'accélérer l'exécution des tests (manuels ou automatisés) de manière reproductible et plus efficace. Vous devez dans un premier temps

comprendre votre méthodologie de test et déterminer où la virtualisation des services peut accélérer le travail des équipes, tout en leur permettant de produire des logiciels de meilleure qualité. Vous pouvez ensuite observer de près l'échange de messages et le comportement entre les composants intégrés de vos applications composites. Pourquoi adopter des tests fonctionnels et tester à partir de l'interface utilisateur, alors que vous avez la possibilité de valider sur la couche d'intégration ? La virtualisation des services vous permet d'émuler plus précocement les logiciels et intégrations de test dépendants non encore disponibles, et d'isoler les défauts pour les résoudre plus rapidement.

Planifier pour la flexibilité

Commencez par identifier les problèmes majeurs. Quels sont les obstacles qui vous empêchent d'effectuer vos tests au bon moment et avec la précision nécessaire ? Pour réduire les goulets d'étranglement de test, les équipes peuvent commencer par virtualiser les composants les plus stables et les plus coûteux à établir en environnement de test. En cas de livraison tardive de certains composants, la virtualisation des services permet de simuler les fonctionnalités manquantes. Toutefois, les priorités évoluant avec le temps, les équipes doivent également être flexibles. Pour cette raison, il n'est pas toujours possible de définir en amont tous les besoins en matière de virtualisation des services.



Commencez simplement et enrichissez progressivement votre mise en œuvre de virtualisation des services à mesure que votre équipe gagne en aisance vis-à-vis du nouveau processus. Évaluez vos besoins en permanence et effectuez des ajustements pour accroître l'efficacité.

Pratiquer l'intégration contrôlée

Trop souvent, les équipes de développement reportent les tests d'intégration à la fin du développement, puis assemblent les composants en croisant les doigts. Malheureusement, les choses ne se déroulent pas toujours comme prévu. Découverts aux dernières étapes de développement, les défauts, problèmes de conception et d'architecture ou bugs massifs mettent en péril tout un projet. Grâce aux composants virtuels, qui simulent le comportement du logiciel réel, l'équipe de test bénéficie d'un contrôle sans précédent.

Les composants virtuels permettent d'isoler les systèmes et de simuler les sous-systèmes, afin de rendre disponibles les composants qui ne le sont pas. Les tests de performance peuvent être réalisés plus tôt pour valider les décisions en matière de conception et d'architecture. Et lorsque le nouveau logiciel ou le logiciel modifié est enfin prêt à être testé, les équipes de développement peuvent introduire le nouveau code source de manière contrôlée en désactivant le composant virtuel et en testant la mise en œuvre réelle.

Tester en permanence, du développement à la production

Vous avez besoin d'un moyen simple de tester en permanence votre code, afin d'éviter l'introduction d'erreurs plus difficiles à trouver et à résoudre à un stade plus avancé du processus. Les testeurs n'ont pas toujours la possibilité d'accéder à des éléments essentiels au bon moment pour réaliser une intégration continue. La virtualisation des services apporte une solution à cette problématique. Grâce à la création de composants virtuels qui fonctionnent de la même manière que la mise en œuvre réelle, les testeurs peuvent procéder plus facilement aux tests de manière synchronisée avec le développement. Au lieu de subir les goulets d'étranglement qui empêchent de mener des tests d'intégration continus, vos équipes pourront tester de manière plus régulière dans beaucoup plus d'environnements qu'auparavant.

Externaliser les données de test

Les tests et la virtualisation des services doivent se baser sur des données de test valides. Or, il n'est pas facile de s'assurer de l'exactitude et de la gérabilité des données dans le processus de test, tant en volume qu'en complexité. Vos données peuvent provenir de plusieurs systèmes et de plusieurs sources. Vous devez donc les externaliser, en les désolidarisant du composant virtuel lui-même. Les personnels qui ne connaissent pas le fonctionnement de la solution de virtualisation peuvent ainsi améliorer plus facilement les données. Vous pouvez donc créer un ensemble de données exact et cohérent avec l'application réelle, sans connaissances techniques de la solution de virtualisation des services elle-même. Lorsque vous extrayez des données d'un environnement de

production, veuillez à utiliser ces données de manière responsable en protégeant les informations personnelles. environment, you need to make sure you use this data in a responsible way that protects personal information.

Explorer des scénarios de test évolués

La virtualisation des services vous permet de réaliser les tests de système importants que vous avez toujours été contraint de laisser de côté pour des raisons de coût et de temps. Ou peut-être n'avez-vous pas accès à tous les composants nécessaires. Les applications logicielles complexes actuelles nécessitent des tests plus complets et plus évolués tout au long de leur cycle de vie.

En étant davantage centré sur la qualité, vous devez débiter les tests beaucoup plus tôt, même les tests négatifs et de performance. La virtualisation des services vous permet de créer des situations de simulation (« what-if ») et de surcharger votre système de sorte à tester la survenue d'éventuelles erreurs en dehors du déploiement de production. Par ailleurs, vous pouvez mettre en œuvre rapidement des chemins alternatifs dans un composant virtuel, afin de créer des situations fréquentes sur le terrain, mais difficiles à reproduire.

Eviter de réinventer la roue

Trop souvent, les développeurs et les testeurs partent de zéro pour tenter d'accroître l'efficacité des tests, d'améliorer l'identification des défauts ou de résoudre les problèmes plus rapidement. Vous pouvez, par exemple, tenter de résoudre des problèmes de goulet d'étranglement en créant manuellement des simulations. Grâce à la virtualisation des services, il est beaucoup plus productif et rentable de mettre en œuvre une solution commerciale basée sur des bonnes pratiques éprouvées pour créer des composants virtuels. L'utilisation d'une technologie basée sur un modèle hautement abstrait est beaucoup plus efficace que le codage manuel de simulations. Les entreprises doivent rechercher une solution directement compatible avec leurs intergiciels (middleware), protocoles et formats. Les développeurs et les testeurs pourront ainsi réellement faire leur travail, c'est-à-dire se consacrer à l'amélioration de la valeur ajoutée du produit et créer des logiciels de meilleure qualité.

La virtualisation des services ne s'adresse pas qu'aux testeurs

Dans l'entreprise, la qualité des applications ne repose pas que sur les testeurs. Chacun doit contribuer à l'amélioration de la qualité des logiciels produits. Aujourd'hui, les programmeurs exécutent des tests dans le cadre de leurs responsabilités de développement. Ils sont donc soumis aux mêmes contraintes que les testeurs. Dans certains cas, les programmeurs testent le code qu'ils ont eux-mêmes écrit sans valider les intégrations. Une grande partie des logiciels dépendants n'étant pas disponible pour les tests, les programmeurs laissent à l'équipe de tests la responsabilité des tests d'intégration. Or, les tests d'intégration sont trop importants pour être reportés à plus tard dans le cycle de développement.

La virtualisation des services permet aux développeurs de réaliser les tests d'intégration plus précocement, de simuler les dépendances manquantes et de bénéficier d'un aperçu de la qualité du livrable en intégrant des tests automatisés dans le cycle des versions.

Partager les composants virtuels au sein de l'entreprise

Votre solution de virtualisation des services doit fournir une méthode simple pour partager des composants virtuels au cours du développement, de l'exploitation et au sein des équipes de test. Les tests devant être permanents, il est important que les membres des équipes de développement logiciel et d'exploitation travaillent à l'unisson. La virtualisation des services implique les équipes pluridisciplinaires de telle sorte qu'elle modifie la dynamique des tests et favorise la conformité aux normes du secteur. Tous doivent travailler ensemble, y compris les analystes, développeurs, programmeurs et testeurs, afin d'obtenir une efficacité maximale, tout en partageant connaissances et expertise.

Accroissement de la productivité des équipes en développant les compétences

L'adoption d'une nouvelle technologie n'a d'intérêt que si les membres des équipes sont bien formés à son utilisation. Trop souvent, les équipes de développement, trop occupées, ne prennent pas le temps nécessaire pour se former, ce qui peut engendrer une baisse de productivité. En permettant aux équipes d'acquérir les bonnes compétences, l'entreprise s'offre un avantage considérable. Les développeurs et les testeurs doivent comprendre les technologies utilisées dans la mise en œuvre applicative et la manière dont la virtualisation des services sera intégrée. Si tous les membres de l'équipe travaillent ensemble pour résoudre les problèmes de test d'applications complexes, l'entreprise est mieux à même de tirer parti de la virtualisation des services.

L'adoption de la virtualisation des services ne se résume pas à l'installation d'un outil. Si vos équipes actualisent en permanence leurs connaissances de la virtualisation des services et acquièrent des compétences expertes, vous constaterez une réduction des goulets d'étranglement de test et des améliorations de la productivité et de la qualité des logiciels.

Découvrir la virtualisation des services et son rôle dans la qualité logicielle

Si vous souhaitez débiter vos tests plus tôt et produire plus rapidement des logiciels de meilleure qualité, La virtualisation des services pour les Nuls (édition limitée IBM) est faite pour vous. Découvrez le secret pour améliorer l'efficacité de vos processus de test tout en réduisant les interruptions et les coûts

- *Comprendre l'évolution de la relation entre les fonctions informatique et commerciale — Nécessité de livrer plus vite*
- *Examiner les applications complexes actuelles — Applications pour mobile, Web, intergiciels (middleware) de réseaux sociaux, applications prêtes à l'emploi, bases de données et mainframes*
- *Découvrir comment débiter avec la virtualisation des services — Elaborer votre étude de rentabilité*
- *Bénéficier des avantages de la virtualisation des services — Réaliser des tests plus tôt pour éliminer les surprises et réduire le risque*



Vous apprendrez :

- **Ce qu'est la virtualisation des services**
- **Comment tester plus efficacement les applications complexes actuelles**
- **Où commencer la virtualisation des services**
- **Les avantages de la virtualisation des services**
- **Les principales étapes pour réussir avec la virtualisation des services**

Allez sur [Dummies.com](https://www.dummies.com)

pour consulter des vidéos, des exemples étape par étape, des articles explicatifs ou pour faire des achats