

Au cœur des modèles

Dans une série d'articles, Paul Gerrard, consultant et spécialiste des tests, aborde diverses questions sur ces derniers. Les modèles de test sont essentiels pour les tests et, dans cet article, Paul Gerrard évoque l'art de créer et d'utiliser ces modèles. Si les testeurs interviennent de plus en plus tôt, de pair avec les développeurs ou du moins plus étroitement, les testeurs (et les développeurs) doivent être capables de créer des modèles, de savoir comment les articuler et les partager, ainsi que d'encourager une meilleure collaboration.

Paul Gerrard
Gerrard Consulting

Sponsorisé par



Les modèles, base de la conception des tests

Dans cet article, je souhaite discuter du concept le plus important dans les tests : l'art (ou est-ce une science ?) de créer et d'utiliser les modèles. Boris Beizer a dit en 1990¹ :

« La phase de tests est un processus dans lequel nous créons des modèles mentaux de l'environnement, du programme, de la nature humaine et des tests eux-mêmes. Chaque modèle est utilisé tant que nous considérons que son comportement est correct ou jusqu'à ce que le modèle ne suffise plus à atteindre l'objectif. »

La conception de tests est le processus par lequel nous sélectionnons, parmi le nombre infini des possibles, les tests que nous estimons les plus pertinents pour nous et nos intervenants. Notre modèle de test nous aide à sélectionner les tests de manière systématique. Les modèles de test sont essentiels pour les tests. Ainsi, le reste de cet article traite de leur importance et de leur utilisation.

Fondements des processus de test

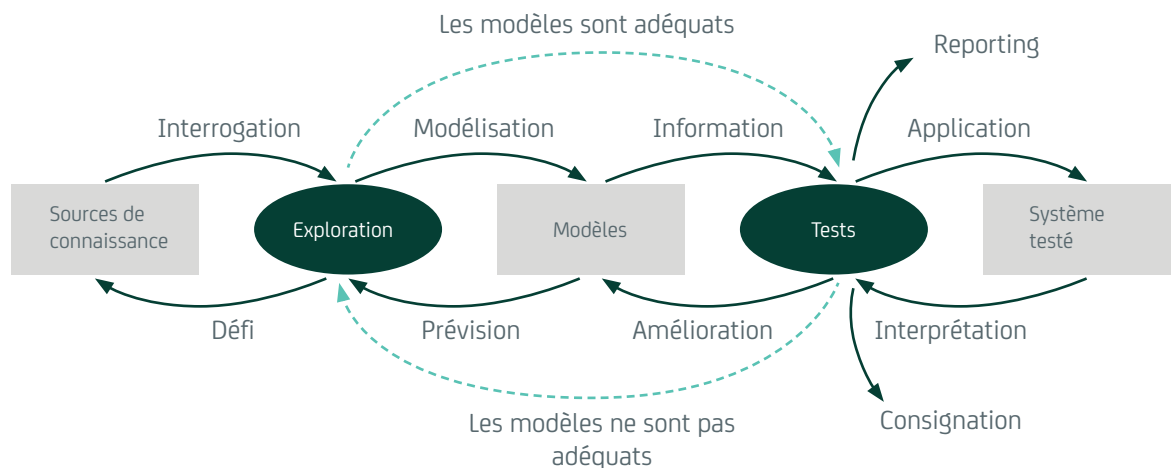
Dans « A New Model for Testing² », je suggère qu'au niveau le plus fondamental, tous les tests peuvent être décrits comme suit :

1. Nous identifions et explorons des sources de connaissance pour élaborer des modèles de test.
2. Nous utilisons ces modèles pour stimuler et valider les sources.
3. Nous utilisons ces modèles pour informer les équipes en charge (du développement et) des tests.

Je fais la distinction entre l'exploration et les tests. La différence principale par rapport à la position commune en matière de tests est que je vais utiliser le terme exploration pour évoquer l'obtention de connaissances sur le système à tester, à partir de sources de connaissance.

Illustration A.

Nouveau modèle pour les tests : les modèles au cœur du travail de réflexion des testeurs



Sources de connaissance

Nous concevons nos modèles à partir des informations que nous obtenons des sources de connaissance. Lors d'une mission de test, notre première tâche consiste à identifier ces sources. Ces sources de connaissance peuvent être les suivantes :

- **Documentation** : spécifications, conceptions, définition des besoins, normes, directives, etc.
- **Personnes** : intervenants, utilisateurs, analystes, concepteurs, développeurs ou autres
- **Expérience** : votre propre connaissance et expérience de systèmes similaires (ou différents), vos préférences, préjugés, suppositions, intuitions, croyances et penchants
- **Système** : le système testé, s'il existe, est disponible et accessible

Nous collectons des informations à partir de nos sources de connaissance pour élaborer les modèles que nous utilisons pour stimuler nos sources et concevoir et/ou tester nos systèmes.

Toutes nos sources de connaissance sont faillibles et incomplètes, comme le sont nos modèles. Les testeurs se servent de leur expérience, de leurs compétences et de leur jugement pour passer ces sources au crible, les comparer et les opposer, les interroger et arriver à un consensus. Ces aptitudes sont généralement associées aux analystes métier ou système.

Qu'est-ce qu'un modèle de test ?

Un modèle de test peut prendre la forme d'une liste de contrôle ou d'un ensemble de critères ; il peut s'agir d'un diagramme dérivé d'un document de conception ou de l'analyse d'un texte narratif. De nombreux modèles de test ne sont jamais transcrits sur papier, ils peuvent rester des modèles mentaux construits spécialement pour guider le testeur alors qu'il explore le système testé.

Nous utilisons des modèles de test aux fins suivantes :

- Simplifier le contexte du test. Les détails non pertinents ou négligeables sont ignorés dans le modèle.
- Focaliser sur un aspect précis du comportement du système. Il peut s'agir de fonctions critiques ou risquées, d'opérations utilisateur ou d'aspects techniques intéressants, ou encore d'aspects particuliers de la construction ou de l'architecture du système.
- Générer un jeu de tests uniques (adaptés au contexte de ce modèle) et divers (par rapport à ce modèle).
- Permettre l'estimation, la planification, la supervision et l'évaluation de l'exhaustivité (couverture) des tests.

Du point de vue du testeur, un modèle aide à reconnaître des aspects particuliers du système pouvant faire l'objet d'un test. Le modèle focalise l'attention sur les zones du système dignes d'intérêt.

Exemple

Imaginons que nous voulions tester la manière dont une voiture (un modèle à boîte de vitesse automatique) accélère depuis l'arrêt jusqu'à sa vitesse maximale, afin de vérifier qu'elle remplit notre objectif de performance (par exemple, passer d'un départ arrêté à 100 km/h en 8 secondes). Nous pouvons modéliser ce système comme suit :

1. Une masse (celle du véhicule et du conducteur) agissant à un centre de gravité défini, qui accélère selon la deuxième loi de Newton
2. Une source d'énergie (le moteur) dont la puissance utile varie entre une valeur minimale et une valeur maximale selon la position de la pédale d'accélérateur
3. Une pédale d'accélérateur pouvant avoir une position variable
4. Des formules qui lient la position de la pédale d'accélérateur, la puissance utile et l'accélération

Nous pouvons certainement obtenir toutes les informations dont nous avons besoin pour notre modèle à partir du document de conception de la voiture.

À l'aide du modèle, nous pouvons concevoir un test comme celui-ci : « À partir de l'arrêt, régler la pédale à la puissance maximale pendant une période de dix secondes. Utiliser nos formules pour calculer une vitesse prévue pour chaque seconde d'accélération. Comparer la vitesse réelle à la vitesse prévue à chaque seconde du test. »

Lorsque nous effectuons ce test dans une vraie voiture, nous comparons sa vitesse à chaque seconde à la vitesse prévue par le modèle. De cette manière, nous pouvons déterminer si la voiture remplit son objectif de performance. Par exemple, si le système testé (la voiture) n'atteint pas la vitesse requise à chaque point de mesure dans le test, nous changeons de voiture ou de modèle (notre interprétation de la conception de la voiture), voire même de conception.

Rien de plus simple, non ?

Les modèles simplifient à l'extrême, il faut donc en utiliser plusieurs

Néanmoins, dans le cas d'un test réel, notre voiture risque de ne pas se comporter comme prévu, car notre modèle ignore plusieurs aspects clés du comportement de la voiture et du contexte. Nous pouvons raisonnablement nous poser les questions suivantes :

- Un vrai conducteur serait-il plus agressif ou plus souple avec la pédale d'accélérateur ?
- Quelle est la force et la direction du vent ?
- Quel est l'état de la chaussée (humide, sèche, goudron, saleté, etc.) ?
- Quelle est la charge transportée par la voiture, en plus du conducteur ?
- La voiture se trouve-t-elle sur une route plane, en montée ou en descente ?
- Quelle est l'efficacité énergétique du système ?

Notre modèle est grossièrement simplifié ; il comporte de nombreuses hypothèses implicites et nécessiterait de considérables ajustements pour devenir une représentation exacte d'une voiture réellement testée. Tous les modèles simplifient le contexte des tests à des degrés variés, raison pour laquelle nous utilisons généralement plusieurs modèles (ou « diverses demi-mesures »³) afin d'élargir notre vision et notre couverture. Le défi consiste à sélectionner des modèles qui sont une représentation suffisamment précise de notre système testé et à interpréter avec soin les résultats de test.

En général, tous les modèles de test, y compris ceux proposés dans les manuels scolaires, sont heuristiques, c'est-à-dire qu'ils sont utiles dans certaines situations, mais toujours incomplets et faillibles. Avant d'adopter un modèle, nous devons savoir quels aspects du comportement, de la conception, des modes d'échec ou des schémas d'utilisation ce modèle nous permet d'identifier, et quelles hypothèses et simplifications il inclut (explicitement ou implicitement).

Nos cerveaux sont de fantastiques moteurs de modélisation

Nos cerveaux sont capables de modéliser et de remodeler nos environnements. Pour effectuer une tâche aussi « simple » que de marcher, notre cerveau a besoin de comprendre, seconde après seconde (et bien plus vite que cela), la configuration des principaux os de notre corps. Il doit comprendre les tensions dans près d'une centaine de muscles. Pour faire le moindre pas, notre cerveau doit envoyer des impulsions à tous ces muscles de sorte qu'ils obéissent à des schémas extrêmement complexes qui nous permettent de marcher et d'atteindre nos destinations sans nous cogner. Cependant, notre cerveau n'est pas assez rapide pour traiter toutes ces informations. Pour simplifier et relever ce défi, il utilise des modèles. La modélisation et la visualisation sont des compétences innées et essentielles, requises dans notre vie quotidienne. Même les robots les plus perfectionnés sur la planète restent vraiment rudimentaires en comparaison.

Vous avez certainement vu des golfeurs répéter leur swing avant de faire un coup. Ils répètent et visualisent le coup, la trajectoire de la balle et la cible. Dans de nombreux sports, les coachs filment les athlètes et analysent leurs mouvements en détails pour les aider à les visualiser, donc à les contrôler, souvent dans un contexte de stress physique intense. Les athlètes modélisent sciemment leur univers pour atteindre la perfection ou le contrôle, et certains l'appellent « la zone ».

Toutefois, les modèles ne représentent pas uniquement des mouvements physiques. Par exemple, lorsque Stephen Hawking a perdu ses capacités physiques, il a inventé une série d'outils mentaux puissants (des modèles) qui lui ont permis de continuer à travailler sur la physique, sans tableau noir ni formules écrites.

Nous utilisons les mêmes compétences de modélisation pour développer et tester les systèmes. Nos cerveaux sont des moteurs de modélisation incroyablement sophistiqués et rapides, et la modélisation mentale domine notre pensée.

Les modèles au cœur des tests

J'avance² que l'exploration et la modélisation réalisées par les développeurs et les testeurs sont très similaires. Si les testeurs interviennent de plus en plus tôt, de pair avec les développeurs ou du moins plus étroitement, les testeurs (et les développeurs) doivent être capables de créer des modèles, de savoir comment les articuler et les partager, ainsi que d'encourager une meilleure collaboration.

Dans un blog⁴, « Courage and Ambition in Teaching and Learning », je défends qu'il est nécessaire de repenser la manière dont nous enseignons la conception des tests. La plupart du temps, les techniques de conception de tests sont enseignées comme des procédures administratives. Nous devons apprendre aux testeurs comment créer des modèles, les apprécier, les sélectionner et les supprimer. Nous devons enseigner les modèles qui sous-tendent les techniques, et non pas juste les procédures permettant de les implémenter.

La réflexion et la modélisation sont au cœur du travail des testeurs. L'évolutivité, la complexité et les taux de modification dans les systèmes numériques et l'Internet des objets (IoT) connaissent tous une forte croissance. Si nous voulons relever ces défis, nous devons concentrer nos efforts sur l'adoption d'une nouvelle manière de penser, ainsi que sur l'acquisition et l'amélioration de nos compétences en modélisation.

À propos de l'auteur

Paul Gerrard est consultant, enseignant, auteur, webmaster, développeur, testeur, conférencier, entraîneur d'aviron et éditeur. Il a assumé de nombreuses missions de conseils concernant tous les aspects des phases de test et d'assurance qualité logiciels, et s'est spécialisé dans l'assurance qualité des tests. Il a animé des présentations et des didacticiels lors de nombreuses conférences sur le thème des tests, en Europe, aux États-Unis, en Australie et en Afrique du Sud, et a été récompensé à plusieurs reprises.

Diplômé des universités d'Oxford et Imperial College London, Paul a été lauréat en 2010 du prix d'excellence Eurostar European Testing, et en 2013 du prix The European Software Testing Awards (TESTA) Lifetime Achievement.

En 2002, Paul a coécrit l'ouvrage « Risk-Based E-Business Testing » avec Neil Thompson. En 2009, il a écrit « The Tester's Pocketbook ». En 2011, il a coécrit « The Business Story Pocketbook » avec Susan Windsor et a rédigé « Lean Python » en 2014.

Cette année-là, Paul a été également président de programme à la conférence EuroSTAR de Dublin.

Il est directeur de Gerrard Consulting Limited, directeur de TestOpera Limited et hôte du Test Management Forum.

Courriel : paul@gerrardconsulting.com

Twitter : [@paul_gerrard](https://twitter.com/paul_gerrard)

Site Internet : gerrardconsulting.com

Pour plus d'informations, rendez-vous sur la page **Développement et test** de CA Technologies.



Restez connecté à CA Technologies sur ca.com/fr



CA Technologies (NASDAQ : CA) fournit les logiciels qui aident les entreprises à opérer leur transformation numérique. Dans tous les secteurs, les modèles économiques des entreprises sont redéfinis par les applications. Partout, une application sert d'interface entre une entreprise et un utilisateur. CA Technologies aide ces entreprises à saisir les opportunités créées par cette révolution numérique et à naviguer dans « l'Économie des applications ». Grâce à ses logiciels pour planifier, développer, gérer la performance et la sécurité des applications, CA Technologies aide ainsi ces entreprises à devenir plus productives, à offrir une meilleure qualité d'expérience à leurs utilisateurs et leur ouvre de nouveaux relais de croissance et de compétitivité sur tous les environnements : mobile, Cloud, distribué ou mainframe. Pour plus d'informations, rendez-vous sur ca.com/fr.

Références

1. « Software Testing Techniques », Boris Beizer, 1990
2. « A New Model for Testing », Paul Gerrard, <http://dev.sp.qa/download/newModel>
3. « Lessons Learned in Software Testing », Kaner, Bach, Pettichord, 2002