

I modelli al centro di tutto

In una serie di articoli Paul Gerrard, guru del testing e consulente, discute vari argomenti collegati all'ambito del testing. I modelli sono fondamentali per i test: in questo articolo, Paul ci parla dell'arte di crearli e utilizzarli. Se i tester vengono "spostati a sinistra" e si trovano associati agli sviluppatori, o quantomeno a lavorare a più stretto contatto con loro, gli uni e gli altri devono essere in grado di creare modelli, imparare come articularli e condividerli e supportare una migliore collaborazione.

La progettazione dei test si basa sui modelli

In questo articolo vorrei occuparmi del concetto più essenziale per il testing: l'arte (o è una scienza?) della creazione e dell'utilizzo dei modelli. Boris Beizer ha affermato nel 1990¹:

"Il testing è un processo in cui creiamo modelli mentali dell'ambiente, del programma, della natura umana e dei test stessi. Ogni modello viene utilizzato finché accettiamo il comportamento come corretto o finché il modello non è più sufficiente allo scopo".

La progettazione dei test è il processo attraverso il quale selezioniamo, tra gli infiniti test disponibili, quelli che riteniamo saranno più validi per noi e i nostri stakeholder. Il nostro modello di test ci aiuta a selezionare i test in modo sistematico. I modelli di test sono fondamentali per il testing e il resto di questo articolo ne illustra importanza e utilizzo.

Processo di test essenziale

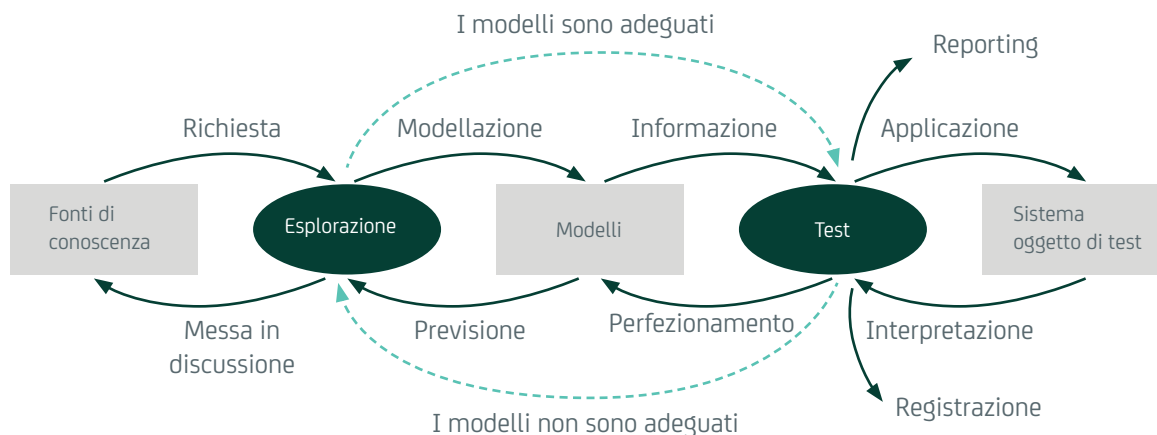
In *The New Model For Testing*² suggerisco che, al livello più essenziale, tutti i test possono essere descritti come segue:

1. Identifichiamo ed esploriamo fonti di conoscenza per costruire modelli di test.
2. Usiamo questi modelli per mettere alla prova e convalidare le fonti.
3. Usiamo questi modelli per modellare (lo sviluppo e) il testing.

Esiste una distinzione tra esplorazione e testing: la differenza principale, rispetto al punto di vista comune sul testing, è che userò il termine "esplorazione" per indicare la raccolta di conoscenze sul sistema da testare da fonti di conoscenza.

Figura A.

Un nuovo modello per il testing: i modelli al centro dei processi di pensiero del tester



Fonti di conoscenza

Costruiamo i nostri modelli a partire dalle informazioni raccolte dalle fonti di conoscenza. Data una finalità per il test, il nostro primo compito consiste nell'identificare queste fonti. Queste fonti di conoscenza possono essere:

- **Documentazione:** specifiche, progetti, requisiti, standard, linee guida e così via.
- **Persone:** stakeholder, utenti, analisti, progettisti e sviluppatori e altri soggetti.
- **Esperienza:** la tua conoscenza ed esperienza di sistemi simili (o dissimili), le tue preferenze, pregiudizi, congetture, intuizioni, convinzioni e inclinazioni.
- **Sistema:** il sistema oggetto di test, se esistente, disponibile e accessibile.

Raccogliamo informazioni dalle nostre fonti di conoscenza per ricavarne modelli che usiamo per mettere alla prova le nostre fonti e progettare e/o testare i nostri sistemi.

Tutte le nostre fonti di conoscenza sono fallibili e incomplete; di conseguenza, lo sono anche i nostri modelli. I tester utilizzano esperienza, abilità e buon senso per vagliare queste fonti, confrontarle e contrapporre, metterle in discussione e arrivare a una tesi condivisa. Queste capacità, naturalmente, di solito sono associate agli analisti di sistema o di business.

Che cos'è un modello di test?

Un modello di test potrebbe essere un elenco di controllo o un insieme di criteri; uno schema ricavato da un documento di progetto o da un'analisi di testo narrativo. Molti modelli di test non vengono mai messi nero su bianco: possono essere modelli mentali costruiti appositamente per guidare il tester mentre esplora il sistema oggetto di test.

Utilizziamo i modelli di test per:

- Semplificare il contesto del test. Dettagli irrilevanti o trascurabili vengono ignorati nel modello.
- Focalizzare l'attenzione su un particolare aspetto del comportamento del sistema. Può trattarsi di caratteristiche essenziali o che presentano rischi, aspetti tecnici o operations utente di interesse, o aspetti particolari della costruzione o dell'architettura del sistema.
- Generare un insieme di test univoci (nel contesto del modello) che siano diversificati (rispetto a quel modello).
- Rendere possibili stima, programmazione, monitoraggio e valutazione della completezza del test (copertura).

Dal punto di vista del tester, un modello aiuta a riconoscere aspetti particolari del sistema che potrebbero essere oggetto di un test. Il modello concentra l'attenzione sulle aree del sistema che sono di interesse.

Un esempio

Supponiamo di voler verificare come una macchina (un modello a cambio automatico) accelera da ferma fino alla sua velocità massima, e accertarci che soddisfi il nostro obiettivo di performance (ad esempio, andare da 0 a 100 in 8 secondi). Potremmo modellare questo sistema come segue:

1. Una massa (complesso del veicolo più conducente) che agisce a un centro di gravità definito e accelera in base alla seconda legge di Newton.
2. Una fonte di alimentazione (il motore) con una potenza variabile da un valore minimo a un valore massimo, dipendente dalla posizione del pedale dell'acceleratore.
3. Un pedale dell'acceleratore che può avere una posizione variabile.
4. Formule relative alla posizione dell'acceleratore, alla potenza erogata e all'accelerazione.

Possiamo probabilmente ottenere tutte le informazioni che ci servono per il nostro modello dal documento di progetto per l'auto.

Utilizzando il modello, potremmo progettare un test come quello che segue: "Da fermi, premere il pedale a fondo per un periodo di dieci secondi. Utilizzare le nostre formule per calcolare una velocità prevista per ogni secondo di accelerazione. Confrontare la velocità effettiva con la velocità prevista in ogni secondo del test".

Quando effettuiamo il test su un'auto reale, mettiamo a confronto la sua velocità in ogni secondo con quella prevista dal modello. In questo modo, potremo determinare se la macchina risponde all'obiettivo di performance. Ad esempio, se il sistema oggetto di test (la macchina) non raggiunge la velocità desiderata in ciascun punto di misurazione nel test, possiamo sostituirla, oppure modificare il modello (la nostra interpretazione del progetto della macchina), o il progetto stesso.

Tutto ok apparentemente, no?

I modelli operano un'eccessiva semplificazione, quindi è bene utilizzarne più di uno

Tuttavia, durante il test vero e proprio, la nostra macchina potrebbe non comportarsi come ci aspettiamo, perché il nostro modello ignora alcuni aspetti chiave del comportamento della vettura e del contesto. Qualche domanda ragionevole che potremmo porci:

- Un conducente reale sarebbe più aggressivo o più delicato con il pedale dell'acceleratore?
- Quali sono velocità del vento e la direzione?
- Quali sono le condizioni della strada (bagnato, asciutto, asfalto, presenza di detriti...)?
- Quale carico trasporta la macchina, oltre al conducente?
- La vettura viaggia in piano, in salita o in discesa?
- Qual è l'efficienza energetica del sistema?

Il nostro modello è grossolanamente semplificato, include molte presunzioni implicite e avrebbe bisogno di un notevole perfezionamento per rappresentare con precisione una vera auto oggetto di test. Tutti i modelli semplificano il contesto di test in misura variabile; di conseguenza utilizziamo normalmente più modelli per ampliare la visione e la copertura (si parla di "mezze misure disparate"³). La sfida consiste nel selezionare modelli che rappresentino con sufficiente accuratezza il nostro sistema oggetto di test e nell'interpretare i risultati ottenuti con cura.

In generale tutti i modelli di test, anche quelli proposti dai libri di testo, sono euristici, nel senso che sono utili in alcune situazioni, ma restano sempre incompleti e fallibili. Prima di adottare un modello, dobbiamo sapere quali aspetti del comportamento e della progettazione, quali modalità di insuccesso o schemi di utilizzo il modello ci aiuta a identificare e quali ipotesi e semplificazioni include (esplicitamente o implicitamente).

Il nostro cervello: un incredibile motore di modellazione

Il nostro cervello è in grado di modellare e rimodellare il nostro ambiente. Per ottenere un risultato "banale" come camminare, il nostro cervello deve sapere, secondo per secondo (più velocemente, anzi) qual è la configurazione delle principali ossa del nostro corpo. Deve conoscere i valori di tensione di circa un centinaio di muscoli. Per compiere un solo passo, il nostro cervello deve inviare impulsi a tutti questi muscoli perché operino in base a modelli altamente complessi, che ci permettono di muoverci e di arrivare a destinazione senza andare a sbattere. Semplicemente il nostro cervello non è abbastanza veloce per elaborare tutte queste informazioni: per semplificare e gestire il problema, utilizza modelli. Modellazione e visualizzazione sono competenze innate ed essenziali necessarie alla nostra vita di ogni giorno. Al confronto, le capacità dei migliori robot sul pianeta sono ancora piuttosto grossolane.

Avrete visto giocatori fare pratica con il loro swing prima di tirare. Provano e visualizzano il colpo, la traiettoria della palla e il bersaglio. In molte discipline sportive, gli allenatori filmano gli atleti e ne descrivono i movimenti in grande dettaglio per favorire la visualizzazione e aiutarli a controllarli, spesso mentre sono soggetti a uno stress fisico enorme. Gli atleti modellano consapevolmente il loro mondo per raggiungere la perfezione o il controllo totale: quello che alcuni chiamano "the zone".

Ma i modelli non rappresentano solo il movimento fisico. Ad esempio, quando Stephen Hawking ha perso le proprie capacità fisiche, si è inventato una raccolta di potenti strumenti mentali, cioè modelli, che gli hanno permesso di continuare a lavorare nell'ambito della Fisica senza utilizzare una lavagna o formule scritte.

Usiamo la stessa capacità di modellazione per sviluppare e testare sistemi. Il nostro cervello è un motore di modellazione incredibilmente sofisticato e veloce, e la modellazione mentale domina il nostro pensiero.

I modelli al centro del processo di testing

La mia idea² è che le attività di esplorazione e modellazione svolte da sviluppatori e tester siano davvero molto simili. Se i tester vengono "spostati a sinistra", e si trovano a essere associati agli sviluppatori, o quantomeno a lavorare a più stretto contatto con loro, gli uni e gli altri devono essere in grado di creare modelli, imparare come articularli e condividerli e supportare una migliore collaborazione.

Nel blog⁴ "Courage and Ambition in Teaching and Learning" affermo che è necessario ripensare il modo in cui insegniamo a progettare i test. Le tecniche di progettazione di test vengono insegnate, nella maggioranza dei casi, come procedure quasi automatiche. Dobbiamo insegnare ai tester come creare i modelli, valutarli, selezionarli e scartarli. Dobbiamo insegnare i modelli sui quali le tecniche si basano, non solo le procedure per implementarli.

Pensiero e modellazione sono al centro dell'attività dei tester. La scala, la complessità e il ritmo del cambiamento nei sistemi digitali e nell'IoT (Internet of Things) sono tutti in aumento. Un nuovo modo di pensare e l'acquisizione e il perfezionamento delle capacità di modellazione devono essere la nostra priorità, se vogliamo far fronte a queste sfide.

L'autore

Paul Gerrard è un consulente, formatore, autore, webmaster, sviluppatore, tester, relatore, allenatore di canottaggio ed editore. Ha svolto incarichi di consulenza su tutti gli aspetti del test e del QA software, specializzandosi nella test assurance. Partecipa con presentazioni e tutorial a conferenze sul testing in Europa, USA, Australia, Sud Africa e, occasionalmente, ha anche vinto dei premi grazie a questi interventi.

Ha studiato presso l'università di Oxford e l'Imperial College di Londra; nel 2010, ha vinto l'Eurostar European Testing excellence Award e, nel 2013, l'European Software Testing Awards (TESTA) Lifetime Achievement Award.

Nel 2002, Paul ha scritto "Risk-Based E-Business Testing" insieme a Neil Thompson. Ha scritto "The Tester's Pocketbook" nel 2009. È stato co-autore di "The Business Story Pocketbook" con Susan Windsor nel 2011 e ha pubblicato "Lean Python" nel 2014.

Sempre nel 2014 Paul è stato Programme Chair della Conferenza EuroSTAR di Dublino.

È Principal di Gerrard Consulting Limited, direttore di TestOpera Limited e host del Test Management Forum.

E-mail: paul@gerrardconsulting.com

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

Web: gerrardconsulting.com

Per ulteriori informazioni, visita **Sviluppo e test** con CA Technologies.



Entra in contatto con CA Technologies all'indirizzo ca.com/it



CA Technologies (NASDAQ: CA) crea software che promuove l'innovazione all'interno delle aziende, consentendo loro di cogliere le opportunità offerte dall'application economy. Il software rappresenta il cuore di qualsiasi business, in ogni settore. Dalla pianificazione allo sviluppo, fino alla gestione e alla sicurezza, CA Technologies lavora con le aziende di tutto il mondo per cambiare il nostro modo di vivere, interagire e comunicare, in ambienti mobile, cloud pubblici e privati, distribuiti e mainframe. Per ulteriori informazioni, visita il sito ca.com/it.

Riferimenti

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