



Examen de integración



Martin Fowler
16 de enero de 2018

Las pruebas de integración determinan si las unidades de software desarrolladas de forma independiente funcionan correctamente cuando están conectadas entre sí. El término se ha vuelto borroso incluso por los estándares difusos de la industria del software, por lo que he tenido cuidado de usarlo en mis escritos. En particular, muchas personas asumen que las pruebas de integración son necesariamente de amplio alcance, mientras que se pueden realizar de manera más efectiva con un alcance más limitado.

Como a menudo con estas cosas, es mejor comenzar con un poco de historia. Cuando me enteré de las pruebas de integración, fue en la década de 1980 y la cascada fue la influencia dominante del pensamiento de desarrollo de software. En un proyecto más grande, tendríamos una fase de diseño que especificaría la interfaz y el comportamiento de los diversos módulos en el sistema. Los módulos se asignarían a los desarrolladores para programar. No era inusual que un programador fuera responsable de un único módulo, pero esto sería lo suficientemente grande como para que pudiera llevar meses construirlo. Todo este trabajo se hizo de forma aislada, y cuando el programador creyó que estaba terminado, se lo entregarían a QA para que lo probaran.

La primera parte de las pruebas serían las pruebas unitarias, que probarían ese módulo por sí mismas, en comparación con las especificaciones que se habían hecho en la fase de diseño. Una vez que se completó, pasamos a las pruebas de integración, donde los diversos módulos se combinan juntos, ya sea en todo el sistema o en subsistemas significativos.

El punto de las pruebas de integración, como su nombre lo indica, es probar **si muchos módulos desarrollados por separado trabajan juntos como se esperaba**. Se realizó activando muchos módulos y ejecutando pruebas de nivel superior contra todos ellos para garantizar que operaran juntos. Estos módulos podrían ser partes de un ejecutable único o separado.

Mirándolo desde una perspectiva más de 2010, estas dos cosas combinadas diferentes:

- probar que los módulos desarrollados por separado funcionen juntos correctamente
- probar que un sistema de múltiples módulos funcionó como se esperaba.

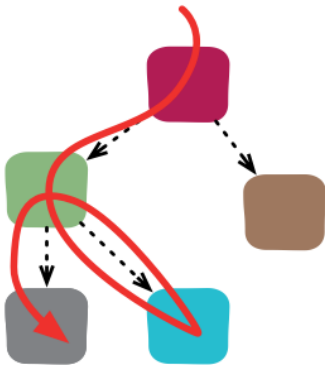
Estas dos cosas eran fáciles de combinar, después de todo ¿cómo probarías los módulos fribile y twibbler sin activarlos a ambos en un solo entorno y ejecutando pruebas que ejercitaran ambos módulos?

La perspectiva de los años 2010 ofrece otra alternativa, una que rara vez se consideró en los años ochenta. En la alternativa, probamos la integración de los módulos fribile y twibbler ejercitando la parte del código en fribile que interactúa con twibbler, ejecutándola contra un **TestDouble** de twibbler. Si el doble de prueba es un fiel doble de twibbler, podemos probar todo el comportamiento de interacción de twibbler sin activar una instancia de twibbler completo. Esto puede no ser un gran problema si son módulos separados de una aplicación monolítica, pero es un gran problema si twibbler es un servicio separado, que requiere sus propias herramientas de compilación, entornos y conexiones de red. Para los servicios, tales pruebas pueden ejecutarse contra una prueba en proceso doble, o contra un doble sobre el cable, usando algo como **saltimbanqui**

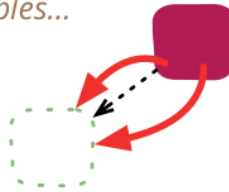
Una captura obvia con pruebas de integración contra un doble es si ese doble es realmente fiel. Pero podemos probar eso por separado usando **ContractTests** .

Usando esta combinación de usar pruebas de integración estrechas y pruebas de contrato, puedo estar seguro de integrarme en un servicio externo sin ejecutar pruebas contra una instancia real de ese servicio, lo que facilita mucho mi proceso de compilación. Los equipos que hacen esto, aún pueden hacer algún tipo de prueba integral del sistema con todos los servicios reales, pero si es así, es solo una prueba final de humo con un rango muy limitado de rutas probadas. También ayuda a tener un **control de calidad** maduro en capacidad de **producción** , y si eso es lo suficientemente maduro, puede que no haya pruebas de sistema de extremo a extremo realizadas.

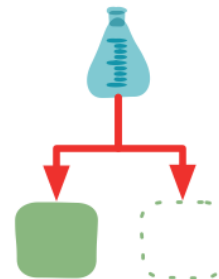
*Integration Testing
commonly refers to
broad tests done with
many modules active...*



*...but it can be done with
narrow tests of interactions
with individual Test
Doubles...*



*...supported by Contract
Tests to ensure the
faithfulness of the double*



El problema es que tenemos (al menos) dos nociones diferentes de lo que constituye una prueba de integración.

pruebas de integración estrechas

- ejercitar solo esa parte del código en mi servicio que habla con un servicio separado
- usa dobles de prueba de esos servicios, ya sea en proceso o remota
- por lo tanto, consiste en muchas pruebas de ámbito limitado, a menudo no más grandes que una prueba unitaria (y generalmente se ejecuta con el mismo marco de prueba que se utiliza para las pruebas unitarias)

amplias pruebas de integración

- requiere versiones en vivo de todos los servicios, lo que requiere un ambiente de prueba sustancial y acceso a la red
- ejercitar las rutas de código a través de todos los servicios, no solo el código responsable de las interacciones

Y hay una gran cantidad de desarrolladores de software para quienes la "prueba de integración" solo significa "amplias pruebas de integración", lo que genera mucha confusión cuando se topan con personas que utilizan el enfoque restringido.

Si sus únicas pruebas de integración son amplias, debe considerar explorar el estilo estrecho, ya que es probable que mejore significativamente la velocidad de prueba, la facilidad de uso y la resistencia. Debido a que las pruebas de integración limitadas tienen un alcance limitado, a menudo se ejecutan muy rápido, por lo que pueden

ejecutarse en las primeras etapas de un **DeploymentPipeline** , lo que proporciona un feedback más rápido si se ponen rojas.

Todo esto es por lo que soy cauteloso con la "prueba de integración". Cuando lo leo, busco más contexto, así que sé a qué tipo realmente se refiere el autor. [1] Si hablo de pruebas de integración amplias, prefiero usar "prueba del sistema" o "prueba de extremo a extremo". No tengo mejor nombre para pruebas de integración estrechas, así que uso eso (pero con "estrecho" para ayudar a señalar al lector la naturaleza de estas pruebas).

Expresiones de gratitud

Birgitta Böckeler, Brian Oxley, Dave Rice, Deepti Mittal, Jonny Leroy, Kief Morris, Raimund Klein, Rogerio Chaves y Tiago Griffo discutieron los borradores de esta publicación en nuestra lista de correo interna.

Notas

1: aunque prefiero centrar la definición en la interacción de los módulos construidos por separado, de vez en cuando veo "prueba de integración" usada para significar algo más grande que una prueba unitaria. Y para algunos usuarios de pruebas de unidades solitarias, los he visto describir pruebas de unidades sociables como "pruebas de integración".

Compartir:   

si encontraste útil este artículo, por favor compártelo. Agradezco los comentarios y aliento

*Encuentra
artículos
similares en la
etiqueta*

categorias de prueba

ThoughtWorks®