Maduración de Conocimiento Mediante una Familia de Experimentos

Sira Vegas

Facultad de Informática. Universidad Politécnica de Madrid.
Campus de Montegancedo. 28660 Boadilla del Monte, Madrid, España
svegas@fi.upm.es

Resumen. Este artículo parte de un estudio preliminar llevado a cabo sobre los experimentos que se han realizado con técnicas de pruebas. El objetivo es examinar con precisión una de las familias que se identificaron para averiguar si los mecanismos de maduración de conocimiento que se han utilizado en la familia han sido suficientes para lograr un fragmento consolidado de conocimiento. La familia elegida ha sido la más aventajada de las que se encontraron, resultando ser la que compara la efectividad relativa de técnicas pertenecientes a los grupos funcional, estructural y revisión de código. Tras detallar cada uno de los experimentos que componen la familia, se hace un análisis comparativo de los mismos, así como de los mecanismos de maduración de conocimiento usados en la misma. Este análisis ha arrojado a la luz que los mecanismos utilizados en la familia: paquetes de laboratorio y asociación a experimentos anteriores, no son suficientes, siendo necesaria una mayor interacción o cooperación entre experimentadores.

1 Introducción

Continuamente se oyen afirmaciones del tipo "la nueva herramienta X mejora la productividad de los desarrolladores" o "la técnica de desarrollo Y permite construir software de mayor calidad". Sin embargo, la mayoría de estas afirmaciones no tiene base científica [6]. Es decir, estas afirmaciones no están respaldadas por evidencia empírica. Pero para hacer de la IS una auténtica ingeniería, los desarrolladores necesitan trabajar con hechos y no con suposiciones, como hacen actualmente. Para la obtención de conocimiento de calidad científica, es necesaria la experimentación.
A pesar de que hace más de 25 años que aparecieron las primeras publicaciones de experimentos en IS, ésta sigue siendo escasa hoy en día. Bien es cierto que desde entonces la comunidad de IS experimental no ha hecho más que crecer: Existen redes a nivel mundial: ISERN desde 1993 cuyo número de miembros aumenta con los años, ESERNET, que la unión europea comenzó a financiar en 1999, y recientemente ha aparecido una conferencia propia, el ISESE. Sin embargo, no parece que la IS empírica esté teniendo el impacto que era de esperar en la comunidad de IS.

Nosotras hemos querido comprobar si la experimentación que se lleva realizando durante todos estos años está alcanzando sus objetivos; es decir, está produciendo conocimiento empíricamente maduro (hechos) que contrarreste con el conocimiento

subjetivo (opiniones y suposiciones) manejado típicamente en IS. Para ello, hemos elegido el campo de técnicas de pruebas. El motivo ha sido que es un campo en el que se han realizado una gran cantidad de experimentos.

En [8] presentábamos una visión general de los experimentos sobre técnicas de pruebas que se han realizado hasta la fecha. En él, se pretendía extraer el conocimiento empírico actual sobre técnicas de pruebas. Sin embargo, una de las conclusiones de dicho artículo fue que el conocimiento empírico actual sobre técnicas de pruebas se halla en un estado muy inmaduro. Las causas de ello eran variadas y se detallan en [9] y [13]. Estas causas giraban en torno a tres temas: la falta de rigor con que se realiza la experimentación, la escasa correspondencia que a menudo ocurre entre el experimento y la realidad, y la falta de existencia de una estrategia de experimentación dirigida a la maduración de conocimiento.

Este artículo pretende estudiar con más detalle una de las familias analizadas en [8]. Más concretamente, se quiere averiguar si los mecanismos relacionados con el tema de la estrategia de experimentación dirigida a la maduración de conocimiento que se utilizó en la familia, han sido suficientes para obtener un fragmento de conocimiento lo suficientemente maduro. Es por esto que de todas las familias examinadas en [8] se ha elegido la familia que aparece como la más aventajada.

Para ello, el artículo se ha organizado del siguiente modo: la Sección 2 explica las características de la familia a tratar. La Sección 3 describe con detalle cada experimento de la familia. La Sección 4 compara los resultados obtenidos por cada experimento. La Sección 5 discute si los métodos empleados en la familia fueron los adecuados para obtener un fragmento de conocimiento maduro. La Sección 6 presenta las conclusiones.

2 Una Familia de Experimentos sobre Técnicas de Pruebas

En *Reviewing 25 years of Testing Technique Experiments* [8] se describía, a alto nivel, la totalidad de experimentos que se han realizado sobre técnicas de pruebas. Por falta de espacio, no se entró en el detalle de cada experimento. Como ya se ha mencionado, este artículo pretende profundizar en una de las familias de experimentos ya examinadas, analizando sus contenidos al detalle, con el fin de analizar las razones por las que, a pesar de contar con un buen número de experimentos, el cuerpo de conocimiento empírico no esté madurando como se esperaría.

La familia que se ha elegido para estudiar, es la que cumple en mayor medida los criterios mencionados en [9]. Más concretamente, se caracteriza porque: todos sus experimentos están realizados con rigor (hablando tanto del diseño, como del análisis de datos y de sus conclusiones); hay una alta correspondencia entre estos experimentos y el mundo real (las variables bajo estudio son de interés para los desarrolladores, se hace un uso realista de la técnica, y los programas y faltas son bastante representativos de la realidad); y hay una cierta coordinación entre los experimentos realizados (la mayoría proporcionan suficientes detalles para la replicación, y hay un cierto encadenamiento de experimentos). En definitiva, se puede decir que en conjunto, esta familia es superior al resto de las estudiadas.

Aunque las raíces de esta familia se pueden trazar a los trabajos de Hetzel [4] y de

Myers [12], estos estudios no se tendrán en cuenta por no estar publicado el primero de ellos, y no estar formalmente explicado el segundo. De este modo, se considera que el experimento original es el realizado por Basili y Selby [2]. Basili y Selby realizaron tres experimentos, durante los años 1982, 1983 y 1984. Posteriormente, Kamsties y Lott [11] realizan dos experimentos más entre 1994 y 1995, que aunque pueden encuadrarse dentro de la misma familia que los de Basili y Selby, no son replicaciones exactas de los mismos. Más tarde, Wood, Roper, Brooks y Miller [14] realizaron una replicación exacta de los experimentos de Kamsties y Lott. Finalmente, Juristo y Vegas [7] realizaron dos experimentos, en 2000 y 2001 respectivamente, utilizando el paquete de replicación creado por Kamsties y Lott, pero partiendo esta vez de los resultados obtenidos por Wood *et al.*

El objetivo de los experimentos de Basili y Selby, es caracterizar la efectividad de las pruebas en función de:

- las técnicas utilizadas,
- el tipo de software,
- el tipo de falta, y
- la experiencia del tester.

Los aspectos de las pruebas que se analizan son:

- la efectividad,
- el coste y
- los tipos de faltas en la detección de defectos.

Aunque el experimento se halla detallado en [1] y [2], no se elaboró un paquete de laboratorio que permitiera su replicación.

Aunque Kamsties y Lott hablan de una replicación del experimento de Basili y Selby, lo cierto es que lo único que mantienen inalterado es el objetivo, que es el estudio de la efectividad de las pruebas. Dado que Basili y Selby no desarrollaron ningún paquete de laboratorio, Kamsties y Lott parten únicamente de la información disponible en [1] y [2]. Esto significa que la comunicación entre los investigadores se realiza a través de los artículos de Basili y Selby.

Wood *et al*. realizan una replicación casi exacta del experimento de Kamsties y Lott, ya que solamente cambian una de las técnicas del experimento. En este caso, esto es posible gracias al paquete de laboratorio que Kamsties y Lott realizaron [10]. De este modo, la comunicación entre investigadores se amplía, pasando de ser vía artículo a realizarse a través del paquete de laboratorio. El paquete aumenta la calidad de la coordinación, ya que amplía el contenido de los artículos correspondientes, proporcionando detalles necesarios para una replicación exacta: programas faltas, etc. Sin embargo, no llega a haber comunicación directa entre los investigadores más allá de la necesaria para pasarse el paquete de laboratorio.

En el caso de los experimentos realizados por Juristo y Vegas, ellas reciben el paquete a través de Wood *et al.*, previa petición, y trabajan a partir de él y de las publicaciones anteriores. De nuevo, se vuelve a repetir el tipo de comunicación anterior, en la que ésta se limita a la necesaria para pasarse el paquete.

Es interesante hacer notar que no ha habido interacción entra los distintos investigadores a la hora de tomar decisiones relativas a la preparación del experimento, ni tampoco a la hora de relacionar los resultados obtenidos con los que se obtuvieron en experimentos anteriores. La comunicación y coordinación entre los experimentadores

se reduce a un paso de testigo, siendo éste en el peor caso una publicación, o en el mejor caso un paquete de laboratorio. A continuación se estudiará si este modo de coordinación entre investigadores y de estrategia de generación de conocimiento empírico es suficiente para lograr un fragmento de conocimiento maduro.

3 Descripción de la Familia de Experimentos

La Tabla 1, muestra una comparativa de los diseños realizados para los distintos experimentos. En ella, se han señalado en negrita los cambios de cada experimento con respecto a su inmediato anterior.

Observando la Tabla 1, podemos ver que las **variables respuesta** de los experimentos se mantienen durante los tres experimentos de Basili, reduciéndose éstas al estudio de la efectividad para el caso de Kamsties y Lott. Sin embargo, Kamsties y Lott amplían el concepto de efectividad, introduciendo nuevas variables respuesta, como las relacionadas con la localización de las faltas en el código, o el tiempo invertido en cada paso de aplicación de las técnicas. De nuevo, Wood *et al.* reducen el número de variables respuesta, pasando a centrarse única y exclusivamente en la cantidad de fallos observados, siendo el mismo para el primer experimento de Juristo y Vegas. Sin embargo, Juristo y Vegas en su segundo experimento vuelven a distinguir entre faltas observables y detectadas, como hacían Kamsties y Lott y Basili y Selby. Se desconoce las razones de estos cambios, pero bien pudieran deberse a que el primer experimento de la familia es siempre el que más ramas de investigación abre (caso de Basili y Selby), ciñéndose los subsiguientes a algunas de las ramas ya abiertas por el primigenio.

Los **factores**, por su parte, merecen ser tratados por separado:

- Las *técnicas* utilizadas van variando cada vez que se realiza una replicación por nuevos investigadores. Así, mientras Basili y Selby utilizan cobertura de sentencias, Kamsties y Lott pasan a utilizar cobertura de predicados, y Wood et al y Juristo y Vegas utilizan cobertura de decisión. En lo que se refiere a las técnicas funcionales, Juristo y Vegas deciden cambiar la técnica utilizada, pasando a ser análisis de valores límite. Siempre que se produce un cambio en la técnica, los autores lo acusan a que creen que es el más frecuente entre los desarrolladores, y por tanto el que se debe investigar.

  Los *programas* varían entre el experimento de Basili y Selby y el resto de experimentos. La creación del paquete de replicación de Kamsties y Lott facilita al resto de investigadores la utilización de sus programas. También es de interés comentar que dentro de los experimentos de Basili, los programas van cambiando. Esto se debe a que Basili parte de cuatro programas inicialmente, pero en cada fase del experimento utiliza sólo tres, por cuestiones de diseño. También cabe destacar que por cuestiones de diseño, Juristo y Vegas utilizan cuatro programas en su primer experimento llevándoles esto a crear un nuevo programa.

Tabla 1. Concepción de cada experimento (1/3).

		Basili	**Basili 2**	**Basili 3**
Planteamiento	**Response variables**	• #, % faults detected • Total fault detection time • Fault detection rate • # computer runs • cpu-time consumed • Max. statement cov. achieved • connect time used • #, % faults observable • % faults observable observed by tester	• #, % faults detected • Total fault detection time • Fault detection rate • # computer runs • cpu-time consumed • Max. statement cov. achieved • connect time used • #, % faults observable • % faults observable observed by tester	• #, % faults detected • Total fault detection time • Fault detection rate • # computer runs • cpu-time consumed • Max. statement cov. achieved • connect time used • #, % faults observable • % faults observable observed by tester
	Factors	• Testing techniques - Boundary value analysis - Statement coverage - Code reading by stepwise abstraction • Software type - Test processing - Numeric abstract data type - Database maintainer • Expertise level - Intermediate - Junior	• Testing techniques - Boundary value analysis - Statement coverage - Code reading by stepwise abstraction • Software type - Test processing - Numeric abstract data type - **Mathematical plotting** • Expertise level - Intermediate - Junior	• Testing techniques - Boundary value analysis - Statement coverage - Code reading by stepwise abstraction • Software type - Test processing - **Database maintainer** - Mathematical plotting • Expertise level - **Advanced** - Intermediate - Junior
	Parameters	• Programming language: SIMPL-T	• Programming language: SIMPL-T	• **Programming language: FORTRAN**
Design	**Type**	Fractional-factorial	Fractional-factorial	Fractional-factorial
	Combinations	• Subjects of all experience levels • Subjects apply the three techniques to three programs	• Subjects of all experience levels • Subjects apply the three techniques to three programs	• Subjects of all experience levels • Subjects apply the three techniques to three programs
	Operation	• Classes: Time not specified • Training + 3 sessions + follow-up • 8 subjects	• Classes: Time not specified • Training + 3 sessions + follow-up • **11 subjects**	• **4-hour tutorial** • Training + 3 sessions + follow-up • **13 subjects**

Tabla 1(cont). Concepción de cada experimento (2/3).

		K&L 1	K & L 2	W et al
Planteamiento	**Response variables**	• % observed failures[1] • **% isolated faults total** • **% faults isolated by chance** • **% faults isolated with technique** • **Time spent per step**	• % observed failures • % isolated faults total • % faults isolated by chance • % faults isolated with technique • Time spent per step	• % observed failures
	Factors	• Testing techniques - Boundary value analysis - **Predicate coverage** - Code reading by stepwise abstraction • **Software type** - **Cmdline** - **Nametbl** - **ntree** • **Group (1-3)**	• Testing techniques - Boundary value analysis - Predicate coverage - Code reading by stepwise abstraction • Software type - Cmdline - Nametbl - ntree • **Group (1-6)**	• Testing techniques - Boundary value analysis - **Brnach coverage** - Code reading by stepwise abstraction • Software type - Cmdline - Nametbl - ntree
	Parameters	• ***Programming language: C*** • **Program length: 4 pages**	• *Programming language: C* • Program length: 4pages	• *Programming language: C* • Program length: 4pages
	Blocking variables	-	-	• **Group (1-6)**
Design	**Type**	3^2x6 randomized fractional-factorial	3^2x6 randomized fractional-factorial	3^2x6 randomized fractional-factorial
	Combinations	Each subject apply three techniques on the three programs	Each subject apply three techniques on the three programs	Each subject apply three techniques on the three programs
	Operative	• **6 hour training + 3 sessions of 3 hours + follow-up** • **27 subjects**	• 6 hour training + 3 sessions of 3 hours + follow-up • **23subjects day 1, 19 day 2, and 15 day 3**	• **Classes: Time not specified** • 6 hour training + 3 sessions of 3 hours + follow-up • **47 subjects**

[1] This is equivalent to the faults detected in Basili and Selby's experiments.

Tabla1(cont). Concepción de ada experimento (3/3).

		Juristo & Vegas1	Juristo & Vegas2
Planteamiento	**Response variables**	• % observed failures	• %revealed failures • **% observed failures**
	Factors	• Testing techniques - **Equivalence class partitioning** - Branch coverage - Code reading by stepwise abstraction • Software type - Cmdline - Nametbl - Ntree - **trade** • **Fault type** - Commission, omission - Cosmetic, initialisation, control	• Testing techniques - Equivalence class partitioning - Statement coverage - Code reading by stepwise abstraction • Software type - Cmdline - Nametbl - ntree • Fault type - Commission, omission - Cosmetic, initialisation, control, **computation**
	Parameters	• *Programming language: C* • Program length: 4 pages • # defects/type: 1 or 2	• *Programming language: C* • Program length: 4 pages • # defects/type: 1 or 2
	Blocking variables	• Group (1-6)	• **Version (1-2) ???** • Group (1-6)
Design	**Type**	3^2x6 randomized fractional-factorial	3^2x6 randomized fractional-factorial
	Combinations	**Each subject apply one technique on one program**	**Each subject apply three techniques on the three programs**
	Operative	• Reminder • Training + 3 sessions • **86 subjects**	• **Classes** • Training + 3 sessions • **46 subjects**

- Basili y Selby trabajan con el factor *experiencia*, que desaparece en los posteriores experimentos. Esto se debe a que los sujetos utilizados en los subsiguientes experimentos, tienen todos la misma experiencia.
- Kamsties y Lott introducen un factor denominado *grupo*. Aunque Wood et al y Juristo y Vegas tienen en cuenta la existencia de grupos, ellos la tratan como variable de bloque y no como factor.
- Juristo y Vegas introducen como nuevo factor el *tipo de falta*. Esto es necesario para cumplir los objetivos del experimento y continuar con las conclusiones del experimento de Wood *et al*.

También introducen, en su segundo experimento, una **variable de bloque** nueva Juristo y Vegas, que no aparecía en ninguno de los experimentos anteriores denominada *versión*. Esta variable aparece por la necesidad de tener más replicaciones de faltas. Al ser pequeños los programas, admiten sólo un número reducido de defectos. Dos versiones de un mismo programa se distinguen única y exclusivamente en los defectos que contienen (aunque siempre tienen el mismo número y del mismo tipo). La necesidad de introducir esta variable surge por la incorporación del factor tipo de falta. Los anteriores experimentos no tuvieron que incorporarla, porque no tenían este factor.

En lo que se refiere al tipo de diseño, no hay variación en el mismo a lo largo de todos los experimentos, siendo siempre un diseño factorial fraccional.

Sin embargo, en lo relativo a las combinaciones, Juristo y Vegas, en su primer experimento, se desvían del resto de experimentos, ya que deciden que cada sujeto aplique una vez una sola técnica sobre un programa, en lugar de las tres técnicas sobre tres programas. Esto lo hacen porque quieren evitar el efecto de aprendizaje de los alumnos con las pruebas de software, y porque tenían demasiados alumnos en comparación con los recursos disponibles para realizar el experimento.

Finalmente, en lo que se refiere a la operativa del experimento, podemos observar que prácticamente todos siguen la misma, existiendo pequeñas variaciones. Así, por ejemplo, Basili da un tutorial en lugar de clases a sus sujetos avanzados, igual que hacen Kamsties y Lott en sus dos experimentos y Juristo y Vegas en el primero. En el caso de Kamsties y Lott y Juristo y Vegas, los alumnos habían tomado clases sobre las técnicas previamente. Por otro lado, Juristo y Vegas eliminan el seguimiento del experimento, por dificultades para tener resultados sobre tantos sujetos en un espacio de tiempo tan breve.

4 Fragmento de Conocimiento Empírico Obtenido

Con el fin de poder entender el modo en que se presentan los resultados de los experimentos, la Tabla 2 explica las abreviaturas que se han utilizado. Asimismo, la Tabla 3 y la Tabla 4 muestran los resultados obtenidos en cada experimento. La Tabla 3 muestra aquellos resultados que son comparables, porque se refieren a aspectos tratados por más de un grupo de investigadores, en nuestro caso la efectividad (detección y observable) y el coste de detección de defectos (tiempo y tasa), y subaspectos de estos cuatro aspectos, que han sido tratados más de una vez, no necesariamente por

investigadores distintos, como la experiencia de los sujetos, o la dependencia de grupos. Por su parte, la Tabla 4 muestra aquellos resultados referidos a aspectos tratados por un sólo grupo de experimentadores, como es el caso de los cuatro últimos aspectos de dicha tabla, o resultados referidos a subaspectos de los aspectos tratados en la Tabla 3 que han sido tratados una sola vez, como es el caso de la investigación de la dependencia del tipo de fallo, o del tiempo de ejecución de los caos de prueba.

Tabla 2. Guía para la lectura de resultados.

SÍMBOLO	SIGNIFICADO	SÍMBOLO	SIGNIFICADO	SÍMBOLO	SIGNIFICADO
=	Igual	P	Programa	om	Omisión
>	Mejor	C	Cobertura	in	inicialización
<	Peor	G	Grupo	int	Interfaz
DE()	Depende de	S	Sujeto	cn	Control
COR()	Correlación	V	Versión	co	Computación
INT()	Interacción	f(x)	Falta de tipo x	d	Datos
COM()	Combinación	fi(x)	Fallo de tipo x	cm	comunicación
CPU()	Tiempo de CPU	CON()	Tiempo de conexión	com	comisión
RUN()	N. de ejecuciones				

Observando la Tabla 3, resulta difícil extraer algún resultado concluyente. Por un lado, el comportamiento de las técnicas, para todos y cada uno de los aspectos, no está claro, ya que cada experimento arroja resultados distintos, a veces, incluso entre replicaciones exactas, como es el caso de los experimentos primero y segundo de Basili y Selby. El único resultado consistente para todos los experimentos que se puede extraer, es el hecho de que parece haber una dependencia en cuanto a la efectividad de las técnicas del programa. Esto es, que conseguiremos mejores resultados durante las pruebas en función del programa que estemos probando.

5 Madurez del Conocimiento Obtenido por la Familia de Experimentos

La familia analizada representa los intentos por parte de una serie de investigaciones de profundizar en la relación entre técnicas de evaluación y efectividad de la misma. En dicha familia, se puede observar que aunque el objetivo de todos los investigadores era común (es por lo que se han podido encuadrar todos los experimentos dentro de la misma familia), y ha existido un nivel avanzado de coordinación (a través de paquetes de laboratorio), los resultados de los que se dispone en la actualidad no tienen un nivel de madurez adecuado como para ser incorporados en el BoK de las pruebas de software. Vamos a intentar encontrar los motivos para este suceso.

En [9], se plantean dos criterios que deben cumplir los experimentos en relación a la maduración de los resultados que generan: que los experimentos no ocurran de forma aislada, sino en familias, y que se produzcan replicaciones.

Tabla 3. Conocimiento obtenido en cada experimento (1/2).

Aspecto	Basili 1	Basili 2	Basili 3	K & L 1	K&L 2	W et al	J & V 1	J & V 2
Effectiveness (detection)	(R=F)>S	R=F=S	R>F>S	F=S=R	F=S=R	-	F>S>R	(F=S)>R
	DE(P)	DE(P)	DE(P)	DE(P)	⌉ DE(P)	-	DE(P)	DE(P)
	⌉DE(C)	⌉DE(C)	⌉DE(C)	-	-	-	-	-
	J=I	J=I	(J=I)<A	-	-	-	-	-
	-	-	-	⌉ DE(G)	⌉ DE(G)	-	-	-
	-	-	-	⌉ DE(S)	⌉ DE(S)	-	-	-
	-	INT(P,T)	INT(T,E)	-	-	INT(P,T)	- IT(F,T) - INT(F,P)	- INT(P,T) - iNT(P,F) - INT(T,F)
	- ∀ f(om), f(in) (R=F)>S - ∀ f(int), (F=S)>R - ∀ f(cn), (R=S)<F - ∀ f(co), (S=F)<R - ∀ f(d), f(cm) F=R=S			- ∀ f(om), (F=S)>R - ∀ f(cn), (F=S)>R - ∀ f(x), S=F=R	∀ f(x), S=F=R	DE(f(x))	- ∀ f(co) R=F=S - ∀ f(com) S≥F>R - ∀ f(om) F>S>R	∀ f(x), (S=F)>R
	COR(PE)			-	-	-	-	-
Effectiveness (observable)	F>S	-	F=S	-	-	-	-	F=S
	⌉DE(C)	⌉DE(C)	⌉DE(C)	-	-	-	-	-
	∀ f(x), S=F			-	-	-	-	-
Fault detection cost (detection rate)	F=S=R	F=S=R	(F=S)<R	-	(S=R)<F	-	-	-
	DE(P)	⌉DE(P)	DE(P)	⌉ DE(P)	⌉ DE(P)	-	-	-
	I>J	I=J	I=J=A	-	-	-	-	-
				⌉ DE(G)	DE(G)	-	-	-
				⌉ DE(S)	DE(S)	-	-	-
	-	-	INT(T,P)	-	-	-	-	-
	COR(PE)			-	-	-	-	-
Fault detection cost (total time)	(F=R)<S	F=S=R	F=S=R	F=S=R	(S=R)<F	-	-	-
	DE(P)	⌉DE(P)	DE(P)	⌉ DE(P)	⌉ DE(P)	-	-	-
	I=J	I=J	I=J=A	-	-	-	-	-
	-	-	-	⌉ DE(G)	DE(G)	-	-	-
	-	-	-	DE(S)	DE(S)	-	-	-
	-	-	INT(T,P)	-	-	-	-	-
	COR(PE)			-	-	-	-	-

Tabla 4. Conocimiento pendiente de replicar.

Aspecto	Basili 3	K & L 1	K&L 2	W et al	J & V 2
Effectiveness (detection)					DE(V)
				↑COM(T)	
Effectiveness (observable)					DE(Fi)
					∀ fi(x) S=F
					INT(P,V,Fi)
Fault detection cost (total time)	CPU(F)>CPU(S) CON(f)<CON(S)				
	RUN(F)=RUN(S)				
Effectiveness (isolated)		F=S=R	(F=R)>S		
		DE(P)	⌉ DE(P)		
		DE(G)	⌉ DE(G)		
		DE(S)	⌉ DE(S)		
		∀ f(x), S=F=R	∀ f(cs), F>(S=R)		
Fault detection cost (observation time)		F>S>R	R>S>F		
		⌉ DE(P)	⌉ DE(P)		
		⌉ DE(G)	DE(G)		
		DE(S)	DE(S)		
Fault detection cost (isolation time)		R>S>F	R>S>F		
		⌉ DE(P)	⌉ DE(P)		
		⌉ DE(G)	⌉ DE(G)		
		⌉ DE(S)	⌉ DE(S)		
Fault detection cost (isolation rate)		F=S=R	(S=R)<F		
		⌉ DE(P)	⌉ DE(P)		
		DE(G)	⌉ DE(G)		
		DE(S)	⌉ DE(S)		

Aunque según Basili [3] existen distintos tipos de replicaciones, a efectos de este artículo vamos a considerar únicamente como replicaciones aquellas que Baili denomina *replicaciones estrictas*, es decir, aquellas que reproduce el experimento anterior de tal forma que ambos experimentos pueden ser considerados el mismo. En [13] se establecen dos tipos de replicaciones: aquellas llevadas a cabo por el mismo investigador, y aquellas llevadas a cabo por distintos investigadores.

Es obvio que el primero de los criterios, relativo a las **familias de experimentos**, sí se cumple, ya que los experimentos aquí analizados ocurren dentro del contexto marcado por los anteriores.

En lo que se refiere a la **existencia de replicaciones**, en la Sección 3 se ha podido observar que las únicas replicaciones que ha habido en los ocho experimentos son el experimento 1 y 2 de Basili y Selby. El resto de experimentos siempre han implicado algún tipo de cambio con respecto a los anteriores. Esto puede haber provocado la obtención de resultados dispares. Por ejemplo, el experimento de Basili y Selby, arrojaba a la luz que parece que las técnicas no tienen un mismo comportamiento, argumento no ratificado por Kamsties y Lott. Sin embargo, ambos no hablan exactamente de las mismas técnicas, luego no podemos hablar de una maduración del mismo resultado.

No obstante, ¿podría haber habido dos replicaciones exactas entre distintos investigadores dentro de esta familia? En el caso de Kamsties y Lott, definitivamente no, puesto que sólo partían de las publicaciones de Basili y Selby. En el caso de Wood *et*

al. y de Juristo y Vegas sí, puesto que ya disponían de un paquete de laboratorio.

Pero si lo que se desea no es replicar exactamente, sino avanzar de otro modo en la maduración de resultados, hay que introducir cambios en los experimentos. En los experimentos de esta familia, los investigadores han ido realizando cambios en los sucesivos experimentos según han ido aprendiendo de los experimentos anteriores. Realizar experimentos en IS es una tarea altamente compleja, donde incluso los investigadores experimentales avezados necesitan un cierto aprendizaje tipo ensayo/error. Por ejemplo, Kamsties y Lott en su primer experimento identifican sólo tres de los seis posibles grupos que podían establecer, error que corrigen en su segundo experimento. Asimismo, Juristo y Vegas en su primer experimento no son capaces de medir la visibilidad de los fallos que aparecen en los programas.

Por otro lado, existen otro tipo de cambios en los experimentos, que no se deben a equivocaciones del experimentador, sino que vienen forzados porque el contexto del experimento cambia, o porque los experimentadores creen mejorar el experimento. Este sería el caso, por ejemplo, de Kamsties y Lott, quienes cambiaron, con respecto al experimento de Basili y Selby, entre otras cosas, una de las técnicas bajo estudio, los programas y la operación experimental. Wood *et al.*, con respecto al experimento de Kamsties y Lott, cambiaron una de las técnicas bajo estudio.

6 Propuesta de Soluciones para Generar Conocimiento Maduro Empíricamente

A pesar de que parece haber una cierta secuencia en la experimentación, el conocimiento no ha madurado convenientemente. Analicemos un poco más en detalle a qué podría deberse. Es interesante notar que los cambios se han hecho siempre por cuenta y riesgo del investigador a cargo del nuevo experimento. Esto ha provocado la toma de decisiones unilateralmente, sin ningún tipo de acuerdo con el experimentador anterior. La falta de discusión entre los experimentadores acerca de los cambios más convenientes a realizar en el experimento podría haber causado la no maduración de resultados anteriores, debido a que no se sigue coherentemente la línea de investigación establecida por experimentos anteriores. Para solucionar este problema, nosotras recomendamos una coordinación entre el investigador que prepara el experimento y los investigadores que realizaron los experimentos anteriormente, mayor que el simple paquete de laboratorio.

Es interesante hacer notar algo que se da en esta familia de experimentos. No hay constancia de que haya habido ningún tipo de interacción entre los experimentadores a la hora de discutir los resultados obtenidos en cada experimento. Actualmente, los experimentadores aportan sus resultados *lanzándolos* a la comunidad en forma de artículos que se centran más que en la presentación de resultados precisos y sólidos, en la descripción de cómo se ha llevado a cabo el experimento en sí. Esto provoca que no se pase por ningún proceso de acumulación o agregación con los resultados de experimentos previos. Parece que los experimentadores esperamos que el conocimiento madure solo mediante la acumulación de resultados, pero sin que nadie haga el esfuerzo de sintetizarlo. Esto impide que haya una maduración de los mismos, que

permita que se pueda ir conformando un BoK.

Así pues, parece que, aunque la existencia de familias y de paquetes de laboratorio ayuda a que no se vaya dando bandazos y se establezca, a grandes rasgos, unas directivas de investigación, no son suficientes para la generación de piezas maduras. Hay dos aspectos importantes que se están descuidando, y es por un lado el establecimiento de una secuencia de avance en la experimentación, y por otro lado, la existencia de un proceso de agregación de resultados cada vez que se concluye un experimento. Estos dos aspectos se podrían resolver con una única acción, que es la coordinación o colaboración de los experimentadores durante el planteamiento del experimento y la elaboración de las conclusiones del mismo.

Durante el planteamiento, la coordinación debería consistir no sólo en proporcionar el paquete de laboratorio, sino también discutir posibles cambios necesarios debido a condiciones particulares del contexto (cantidad de sujetos o tiempo disponible, etc.). Asimismo, también se podrían discutir posibles cambios a realizar sobre el experimento anterior, motivados por la conclusiones que en él se obtuvieron. Por ejemplo, Wood *et al* llegaron a la conclusión en su experimento que ciertas técnicas podrían ser más sensibles que otras a la hora de detectar ciertos tipos de defectos en el software. Juristo y Vegas decidieron explorar esta hipótesis en su primer experimento, pero para hacer esto se requerían ciertos cambios en el diseño del experimento anterior. Esto es lo que se conoce como el ciclo experimentación/aprendizaje, descrito en [5].

La coordinación en la fase de elaboración de conclusiones, consistiría en que se reúnan los experimentadores que han realizado experimentos anteriores en la familia al finalizar cada experimento, para en una primera etapa realizar la agregación o maduración de resultados. Esto permitiría la creación de un BoK maduro y consistente en el área.

7 Conclusiones

Este artículo parte de un estudio preliminar, cuyos resultados aparecen descritos en un artículo anterior, realizado sobre los experimentos llevados a cabo con técnicas de pruebas. Dado que en el artículo anterior no se pudo entrar en el detalle de cada una de las familias de experimentos encontradas, este artículo examina con precisión una de ellas, con el motivo de averiguar si los mecanismos de maduración de conocimiento que se han utilizado en ella han sido suficientes para lograr un fragmento consolidado de BoK.

La familia elegida es la que compara la efectividad relativa de técnicas pertenecientes a los grupos funcional, estructural y revisión de código. El motivo de elegir esta familia de experimentos es porque resultó ser la más aventajada.

Una vez elegida la familia sobre la que se va a trabajar, hemos entrado en el detalle de cada uno de los experimentos que la componen, examinando para cada uno de ellos cosas tales como: cuáles fueron los mecanismos de comunicación entre los investigadores que realizaron el experimento y los precedentes; cómo han ido variando los planteamientos y diseños de cada experimento; y la maduración de los resultados obtenidos en cada uno de ellos.

Finalmente, se hace un análisis de los mecanismos de maduración de conocimiento usados en la familia. Este análisis arroja a la luz que los mecanismos utilizados en la familia: paquetes de laboratorio y asociación a experimentos anteriores, no son suficientes, siendo necesaria una mayor interacción o cooperación entre experimentadores.

Referencias

[1] V.R. Basili and R.W. Selby. Comparing the Effectiveness of Software Testing Strategies. Department of Computer Science. University of Maryland. Technical Report TR-1501. College Park. May 1985,

[2] V.R. Basili and R.W. Selby. Comparing the Effectiveness of Software Testing Strategies. *IEEE transactions on software engineering.* Pages 1278-1296. SE-13 (12), 1987.

[3] V. Basili, F. Sd F. Lanubile. Building Knowledge through families of experiments. Isactions on Software Engineering. Vol 25, N.4. July/August 1999.

[4] Hetzel, W.C. An experimental Analysis of Program Verification Methods. PhD thesis, University of North Carolina, Chapel Hill, 1976.

[5] N. Juristo, AM Moreno. *Basics of Software Engineering Experimentation.* Kluwer. 2001.

[6] N. Juristo and A.M. Moreno. Reliable Knowledge for Software Development. *IEEE Software.* September/October 2002. Pages 98-99.

[7] N. Juristo and S. Vegas. Functional testing, structural testing and code reading: What fault type do they each detect? *Empirical Methods and Studies in Software Engineering- Experiences from ESERNET*. Springer-Verlag. Volume 2785. Chapter 12. Pages 235-261.2003.

[8] N. Juristo, A.M. Moreno and S. Vegas. Reviewing 25 years of testing technique experiments. *Empirical Software Engineering*, Vol 9, N. 1, pages 7-44, 2004.

[9] N. Juristo, A.M. Moreno and S. Vegas. Towards building a solid empirical body of knowledge in testing techniques. *Workshop on Empirical Research in Software Testing*. July 12, 2004. Boston.

[10] E. Kamsties and C. Lott. An empirical evaluation of three defect detection techniques. Technical Report ISERN 95-02, Dept. Computer Science, University of Kaiserslautern, May 1995.

[11] E. Kamsties and C.M. Lott. An Empirical Evaluation of Three Defect-Detection Techniques. *Proceedings of the Fifth European Software Engineering Conference*. Sitges, Spain. September 1995.

[12] Myers, G.J. 1978. A Controlled Experiment in Program Testing and Code Walkthroughs/Inspections. *Communications of the ACM*. Vol. 21 (9). Pages 760—768.

[13] S. Vegas, N. Juristo and A.M. Moreno. Aggregating the results of 25 years of testing technique experiments. *International Software Engineering Research Network. ISERN'04*. 16-18 August. Redondo Beach, California, USA.

[14] M. Wood, M. Roper, A. Brooks and J. Miller. Comparing and Combining Software Defect Detection Techniques: A Replicated Empirical Study. *Proceedings of the 6th European Software Engineering Conference*. Zurich, Switzerland. September 1997.