# Efectividad del Test-Driven Development: Un experimento replicado

Oscar Dieste.
Efraín Fonseca C.
Geovanny Raura
Priscila Rodríguez.





Julio 4 y 5 de 2015

# AGENDA

1. Antecedentes
2. Trabajos relacionados
3. Experimento Original y Replicación
4. Resultados obtenidos
5. Conclusiones y trabajo futuro

# Desarrollo de Software Ágil

- Desde su introducción en la década de los 90's, las metodologías agiles han venido ganando adeptos y actualmente se configuran como una de las aproximaciones más utilizadas para desarrollar software.
- Las metodologías ágiles se basan en una serie de prácticas, tales como la programación por pares o el desarrollo dirigido por pruebas (TDD, Test-Driven Development).
- Las técnicas de desarrollo ágil prometen mejorar la **calidad** del producto software y la **productividad** de los desarrolladores.

# User Stories (US)– Historias de Usuarios

- ✓ Relato acerca de qué problema debe resolver el sistema. Representa una parte de la funcionalidad del sistema que es coherente para el cliente.
- ✓ La historia de usuario debe responder a tres preguntas: ¿Quién se beneficia?, ¿qué se quiere? y ¿cuál es el beneficio?..

*Historia: Responder a comentarios*

*Como: Lector del Blog*

*Quiero: adicionar comentarios a las entradas y responder a comentarios de otros lectores*

*Para: mantenerme en contacto con los demás usuarios del blog*

3

Como (rol) quiero (algo) para poder (beneficio).

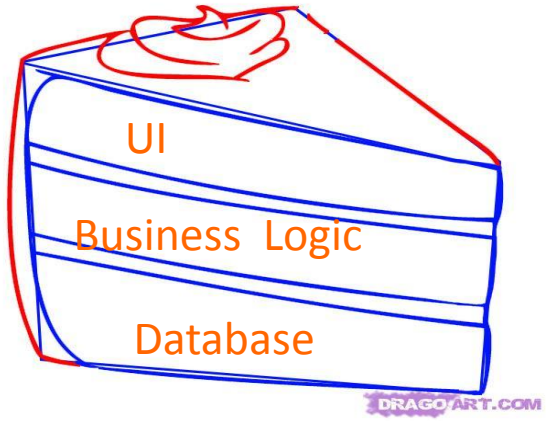
Mike Cohn

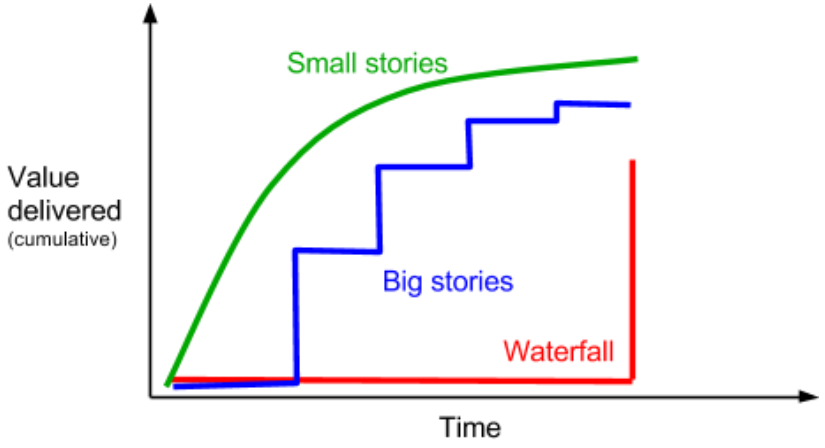
# Slicing

- ✓Una Historia de Usuario puede ser dividida en varias partes (Slicing).
- ✓Se prefiere una división vertical (Cumplir con: INVEST)

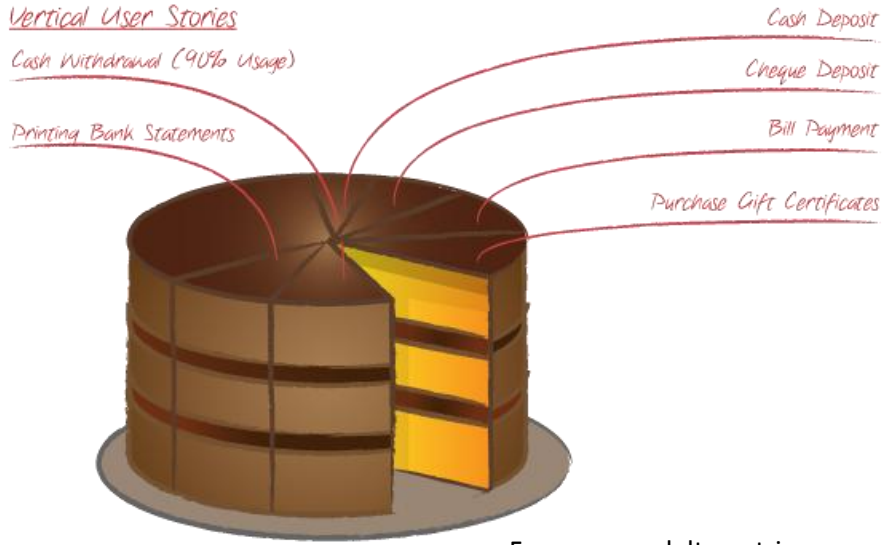
Independent, Negotiable, Valuable, Estimable, Small & Testable

- ✓Las historias se dividen para:
  - Mejorar la comprensión, la estimación, el establecimiento de prioridades
  - Realizar progresos visibles, mayor satisfacción del equipo
  - Obtener información más rápido.



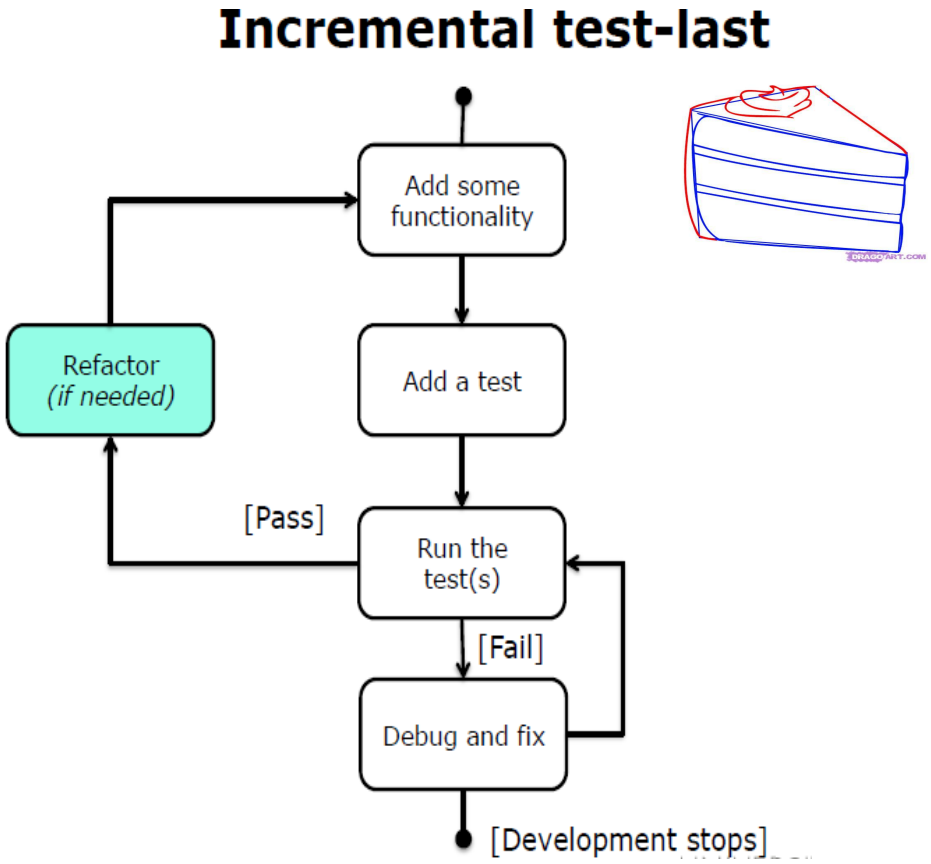

From: Elephant Carpaccio facilitation guide




From: www.deltamatrix.com

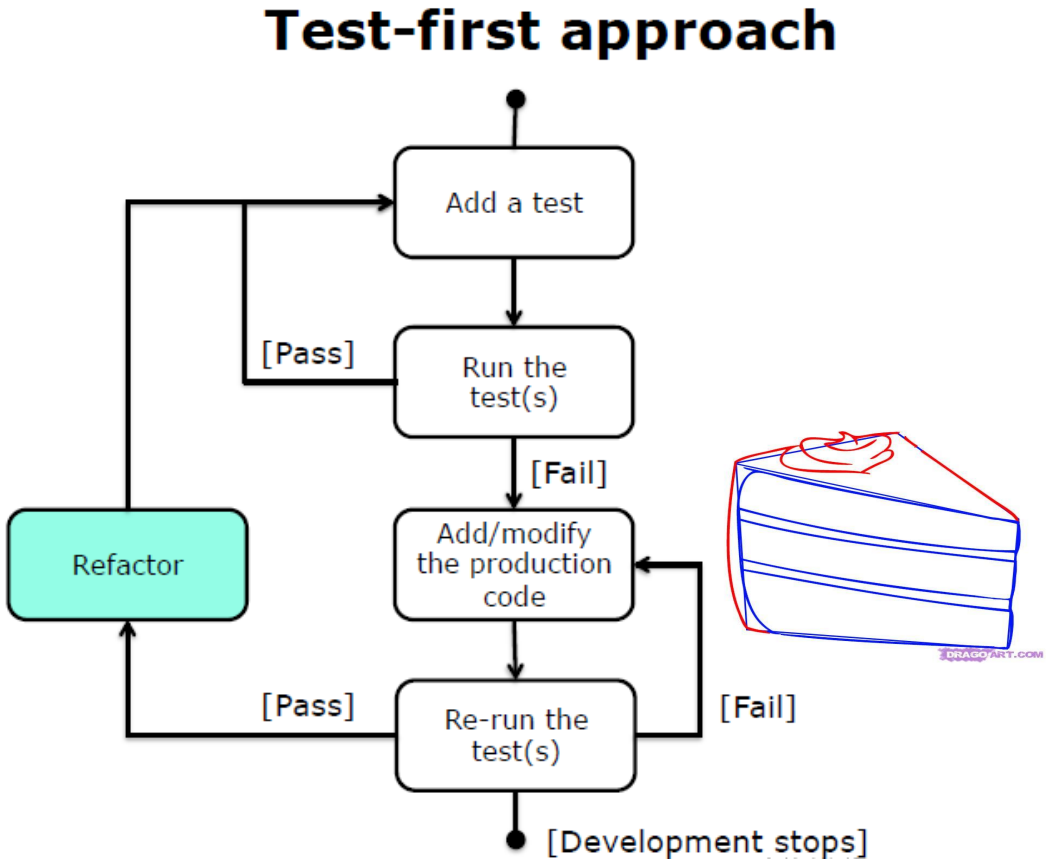
# Incremental test last programming - ITL

- ✓ El enfoque típico de desarrollo es ejecutar las pruebas después de que el código ha sido completado
  - 1 US -> 1 sesión de pruebas
  - **Test-last-Development TLD**
- ✓ Cuando una US es dividida, cada "slice" se puede probar de forma individual- **Incremental Test-last - ITL**



# Test Driven Development- TDD

- ✓ Propuesto por Kent Beck en su libro Test-Driven Development by Example, 2002
- ✓ Propone que en lugar de realizar algún diseño o modelo de software, se debe enfrentar el desarrollo en base a la generación de pruebas unitarias antes de la generación efectiva del código.
- ✓ TDD sigue el enfoque **Test First**



# TDD Vs. ITL/TLD

- ✓ Existen múltiples estudios empíricos que comparan TDD frente a ITL / TLD.
- ✓ Los aspectos típicamente analizados son la calidad interna y externa del software y la productividad de los desarrolladores.
  - La **calidad externa** de un sistema es medida usualmente como el número de casos de prueba pasados y el número de defectos por unidad de tamaño del código (por ejemplo, líneas de código u otra medida adecuada).
  - La **productividad** mide la velocidad de desarrollo del equipo.
  - La **calidad interna** de un sistema, en general hace referencia a la calidad en el diseño, esto es que sea simple, modular y fácil de entender y mantener.

# Estudios empíricos sobre la productividad y calidad externa de TDD vs ITL/TLD

| Revisión | Año | Calidad Externa | Productividad |
|---|---|---|---|
| **Kollanus [19]** | 2010 | Experimento controlado: Sin diferencia<br>Estudios de caso: Mejora<br>Otros: Mejora<br>PROMEDIO: **Mejora** | Experimento controlado: No concluyente<br>Estudios de caso: reducción<br>Otros: Mejora<br>PROMEDIO: **Reducción** |
| **Turhan et al. [10]**<br>**(*) Shull et al [16]** | 2010 | Experimento controlado: no concluyente<br>Estudios piloto: Mejora<br>Industria: Mejora<br>PROMEDIO: **Mejora** | Experimento controlado: mejora<br>Estudios piloto: no concluyente<br>Industria: Disminución<br>PROMEDIO: **No Concluyente** |
| **Rafique & Misic [11]** | 2013 | Experimento académico: sin diferencia<br>Industria: Mejora<br>Test Last: Mejora<br>Iterative Test-Last: no concluyente (potencial disminución)<br>PROMEDIO: **Mejora** | Experimento académico: mejora<br>Industria: disminución<br>Test Last: disminución<br>Iterative Test-Last: no concluyente (potencial mejora)<br>PROMEDIO: **No concluyente** |
| **Munir et al. [16]** | 2014 | Estudios de Alto Rigor y Alta Relevancia (A): Mejora<br>Estudios de Bajo Rigor y alta Relevancia (B1): Mejora<br>Estudios de Alto Rigor y baja Relevancia (B2): sin diferencia<br>Estudios de Bajo Rigor y Baja Relevancia (C) : no concluyente<br>PROMEDIO: **Mejora** | Estudios de Alto Rigor y Alta Relevancia (A): disminuye<br>Estudios de Bajo Rigor y alta Relevancia (B1): disminuye<br>Estudios de Alto Rigor y baja Relevancia (B2): sin diferencia<br>Estudios de Bajo Rigor y Baja Relevancia (C) : no concluyente<br>PROMEDIO: **No concluyente** |

## Conclusiones importantes respecto a investigaciones sobre TDD

- ✓ Parece ser que TDD mejora la calidad externa.
- ✓ Los estudios sugieren que TDD no posee efecto alguno, ni positivo ni negativo, sobre la productividad, al contrario de lo que sugieren los promotores de esta técnica.
- ✓ Adicionalmente, y a diferencia de la práctica de programación por pares, pocos estudios empíricos han estudiado posibles variables moderadoras (ej: la experiencia de los programadores).
- ✓ Las limitaciones en el conocimiento científico acerca de TDD han propiciado que algunos investigadores continúen realizando estudios experimentales en TDD.

# Información sobre el experimento original

- ✓ El experimento original fue realizado por N. Juristo (Investigadora de la Universidad Politécnica de Madrid) y su equipo en el marco del proyecto ESEIL (https://sites.google.com/site/diproeseil/)
- ✓ El objetivo de este experimento fue estudiar la **efectividad** de **TDD** en **comparación** con Incremental Test-Last (**ITL**).
- ✓ El experimento original ensayó como **factor principal** la **aproximación de desarrollo** medida en este estudio como **la Calidad Externa y la Productividad**, con los niveles ITL y TDD
- ✓ Se usó como **factor secundario** la **tarea** que los sujetos debían resolver. La tarea tuvo cuatro niveles, que correspondían con cuatro katas ampliamente usados en experimentos acerca de TDD: MarsRover (MR), MusicPhone (MP), BowlingScoreKeeper (BSK) y Sudoku (SDKU).

# Factores y Variables Respuesta

- ✓ En el experimento original se han estudiado dos variables: la calidad externa (QLTY) y la productividad (PROD)
- ✓ **QLTY** representa el **grado de corrección del código** desarrollado por los sujetos, y se define como:

$$QLTY = \frac{\sum_{i=1}^{\#tus} QLTY_i}{\#TUS}$$

En el donde QLTYi ,es la calidad de la historia de usuario i-esima implementada por el sujeto. QLTYi se define como:

$$QLTY_i = \frac{\#Assert_i(Pass)}{\#Assert_i(All)}$$

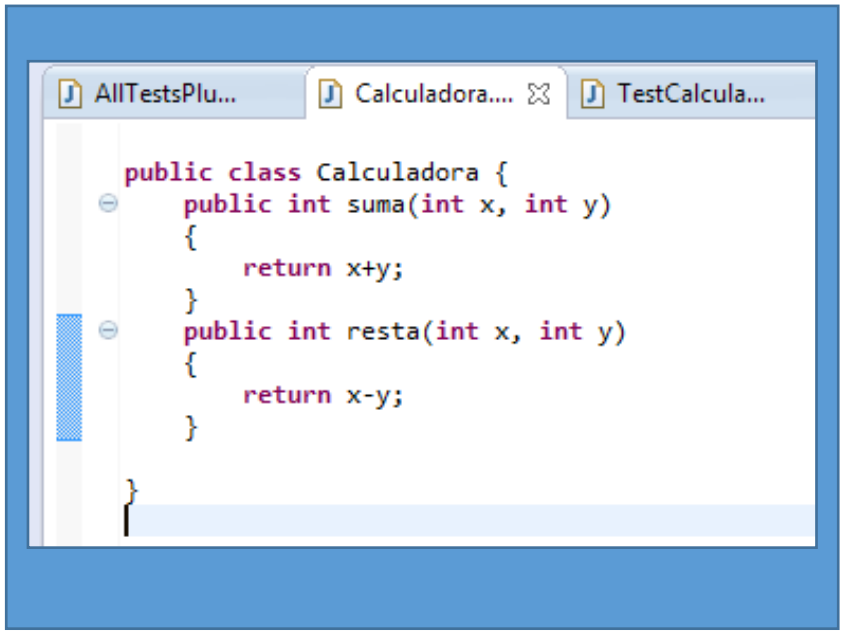
Mientras que #TUS (Tacklet User Stories) es:

$$\#TUS = \sum_{i=1}^{\#us} \#Assert_i(Pass) \geq 0 \mapsto True$$

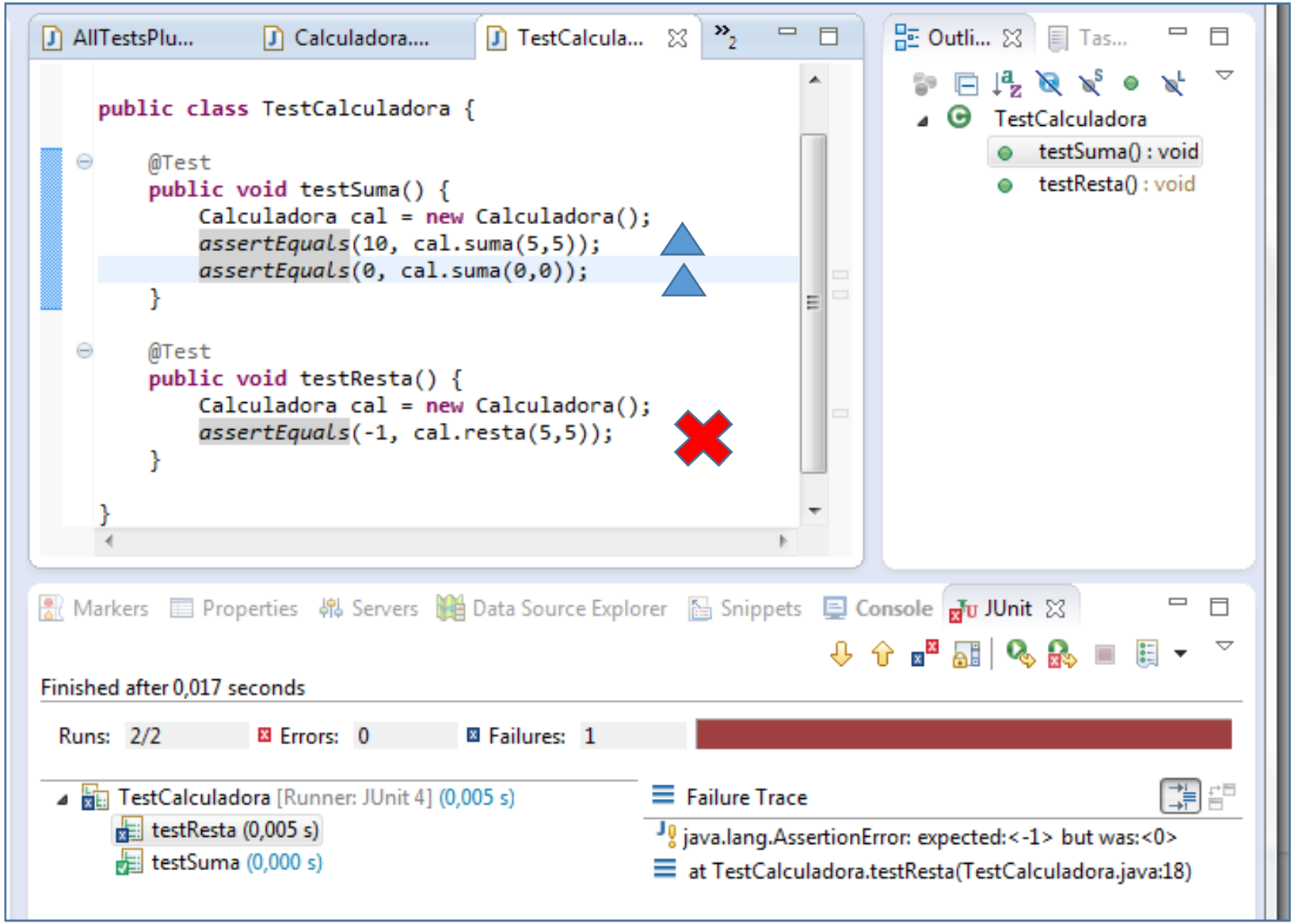
En ambos casos, #Asserti(Pass) representa el numero de aserciones de jUnit

- ✓ **PROD** representa la **cantidad de trabajo** realizada por los sujetos, y se define como:

$$PROD = \frac{\#Assert(Pass)}{\#Assert(All)}$$




Assertions FAILED :33:%
Assertions SUCCEDED :67:%
Assertions ERRORS :0:%
RUN IN TOTAL 2:  TEST CASES
TUS: 1
PERTUS: 100.0%
QLTY: 66.67%
PROD: 66.67%



# Hipótesis

- ✓ El experimento original posee dos hipótesis experimentales; la primera hace referencia a que la calidad del producto software no se ve alterada por el uso de ITL o TDD:

$$H_{10} : QLTY_{ITL} = QLTY_{TDD}$$
$$H_{11} : QLTY_{ITL} <> QLTY_{TDD}$$

- ✓ La segunda hipótesis afirma lo mismo respecto a la productividad:

$$H_{20} : PROD_{ITL} = PROD_{TDD}$$
$$H_{21} : PROD_{ITL} <> PROD_{TDD}$$

# Diseño experimental

- ✓ Debido al previsible **reducido número de sujetos** experimentales, los investigadores originales decidieron utilizar un **diseño de medidas repetidas** para aumentar el poder estadístico.

| Secuencia Temporal | | | |
|---|---|---|---|
| Sesión 1 | Sesión 3 | Sesión 3 | Sesión 4 |
| ITL | TDD-1 | TDD-2 | TDD-3 |

- ✓ Este diseño puede calificarse como ABBB, ya que el nivel de interés (TDD) se aplica repetidas veces para mejorar las habilidades de los sujetos y poder detectar más fácilmente sus efectos.

- ✓ El **factor secundario tarea** fue **contrabalanceado** en las cuatro sesiones experimentales para evitar confundir los factores **tarea** y **aproximación de desarrollo.**

# Amenazas a la validez

- ✓ Los diseños de medidas repetidas poseen generalmente las siguientes amenazas a la validez: fatiga, práctica, carry over y orden/periodo. En el presente experimento opera sin duda la amenaza de fatiga, ya que las sesiones son contiguas en el tiempo.
- ✓ Creemos que las restantes amenazas no aplican, por las siguientes razones:
  - Práctica: TDD es una aproximación nueva para la mayoría de los sujetos experimentales. La práctica obtenida mediante la aplicación repetida del nivel TDD no representa una amenaza a la validez sino una condición necesaria para alcanzar los objetivos experimentales.
  - Carry over: ITL utiliza estrategias parecidas a TDD, por lo que el carryover, al igual que la práctica, resulta favorable para el experimento.
  - Orden/periodo: Las sesiones experimentales son contiguas en el tiempo. No existe ninguna razón que sugiera la existencia de un efecto de orden/periodo.

# Ejecución del experimento original – Contexto y Participantes

- ✓ Contexto y participantes:
  - El experimento original se realizó en la academia, utilizando como sujetos experimentales 16 estudiantes de maestría de la UPM.
  - Todos los sujetos poseen titulaciones relacionadas con la informática, y una experiencia profesional media-baja (menor a 4 años, con pocas excepciones)
  - Todos han usado lenguajes procedurales y orientados a objetos.
  - Tres sujetos reportan haber usado TDD como metodología de desarrollo por un breve lapso de tiempo.

# Ejecución del experimento original - Resultados

- ✓ Resultados:
  - El experimento original **ha sido incapaz de obtener efectos significativos** de la **aproximación de desarrollo** tanto para la variable respuesta calidad como productividad, si bien en esta última se aprecia una cierta tendencia a la significación estadística (p-valor = 0,116). Por el contrario, se ha podido constatar la **influencia de la tarea tanto en calidad como en productividad** (p-valor < 0 en ambos casos).

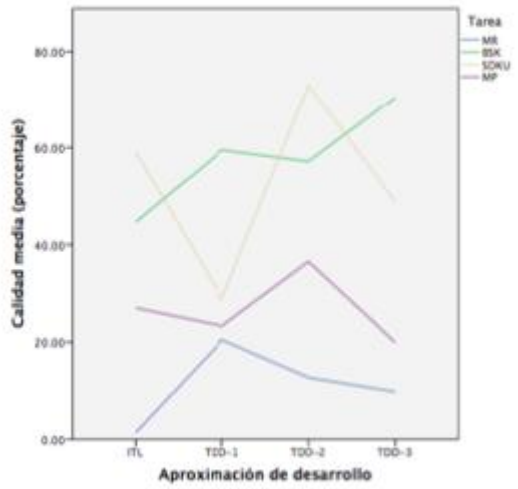



**Fig. 1.** Gráfico de perfil (variable respuesta calidad)

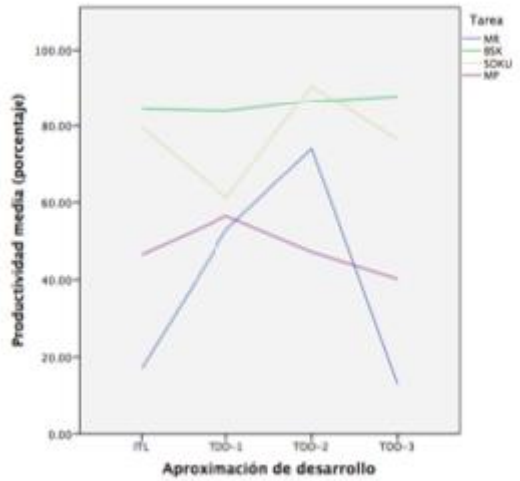



**Fig. 2.** Gráfico de perfil (variable respuesta productividad)

# Información acerca de la replicación

- ✓ La replicación fue realizada en la Universidad de las Fuerzas Armadas ESPE de Ecuador - Sede Latacunga (ESPEL en lo que sigue), en el marco del curso de Verificación y Validación de Software de la Maestría en Ingeniería de Software.
- ✓ La razón principal que motivó la realización de la replicación fue confirmar los resultados del experimento original o, en el caso de encontrar diferencias, identificar los factores o parámetros que podrían haber causado las desviaciones.
- ✓ La replicación fue guiada por uno de los experimentadores originales (O. Dieste) durante todo el ciclo experimental y asistida por un investigador local (G.Raura).
- ✓ La replicación puede calificarse como literal (es decir, la replicación se asemeja al experimento original tanto como sea posible), conjunta (algunos de los experimentadores originales participaron en la replicación) y externa (la replicación se llevó a cabo en un sitio diferente).
- ✓ La diferencia, y aún así no sustancial, reside en el diseño experimental. La replicación tuvo una duración de 4 días, lo que exigió eliminar una de las sesiones TDD. El diseño de la replicación fue ABB.
- ✓ Las amenazas a la validez son las mismas del experimento original

# Ejecución de la Replicación– Contexto y Participantes

- ✓ Contexto y participantes:
  - El experimento original se realizó en la academia, utilizando como sujetos experimentales 17 estudiantes de maestría de ESPEL.
  - Todos los sujetos poseen titulaciones relacionadas con la informática, y una experiencia profesional considerable aunque no todos los sujetos reportaron su experiencia.
  - Todos han usado lenguajes procedurales y orientados a objetos, sin embargo un 50% se califican como sin experiencia o novatos en programación.
  - Ningún sujeto recibió formación específica sobre TDD aunque uno reporta haber usado TDD en entornos ágiles durante 1 año.

# Ejecución de la replicación- Resultados

- ✓ Resultados:
  - En lo que respecta a la aproximación de desarrollo ITL supera a TDD-1 y TDD-2 tanto en calidad como en productividad. En lo tocante a las tareas, BSK alcanza las mayores cotas de calidad y productividad, seguido por SDKU, MP y, finalmente, MR. Las dispersiones son notables tanto para la aproximación de desarrollo como para la tarea

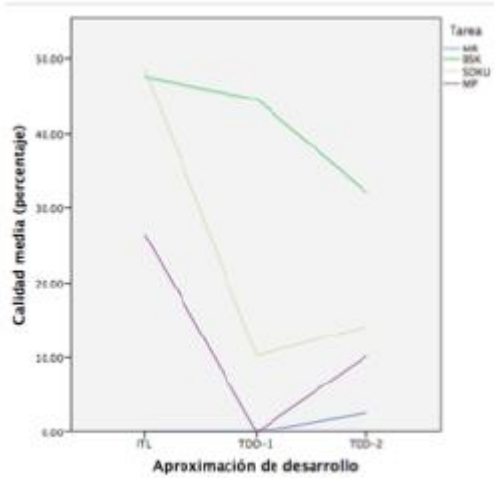



**Fig. 5.** Gráfico de perfil (variable respuesta calidad)

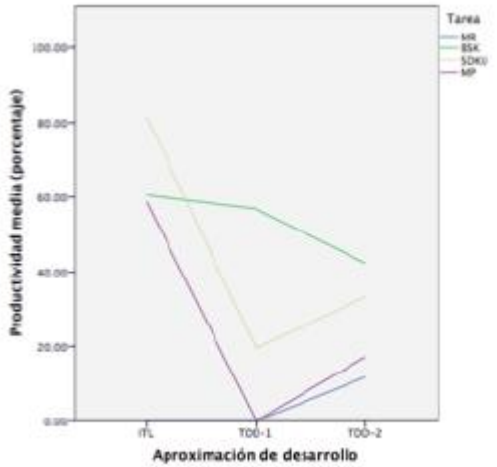



**Fig. 6.** Gráfico de perfil (variable respuesta productividad)

# Comparación de Resultados – Experimento Original y Replicación

- ✓ La replicación es del tipo literal, por tanto los resultados son comparables en todos sus aspectos y coincidentes en su mayor parte.
- ✓ Ni el experimento original ni la replicación fueron capaces de obtener efectos significativos de la aproximación de desarrollo tanto para la variable respuesta calidad como productividad.
- ✓ En lo que respecta a la tarea, los resultados son significativos en ambos casos, y con unas tendencias muy similares. BSK y SDKU obtienen mayores valores de productividad y calidad que MP y MR. , aunque SDKU en ESPEL se destaca menos que en el experimento original.
- ✓ La mayor diferencia entre ambos experimentos son los valores absolutos de las variables respuesta. Los datos de ESPEL son claramente más bajos que en UPM. (Puede deberse a las características de la población – experiencia en programación, o a problemas de motivación y cansancio por el carácter intensivo del curso en ESPEL.
- ✓ Finalmente, se observa una caída muy fuerte de productividad y calidad en ESPEL el primer día que los sujetos aplicaron TDD. Ello podría indicar la necesidad de mayor entrenamiento.

# Conclusiones y trabajo futuro

- ✓ TDD no produce beneficios en calidad o productividad, o al menos no de forma inmediata.
- ✓ Parece necesario que los sujetos experimentales reciban training intensivo para que los efectos de TDD sean evidentes.
- ✓ Se necesita una mayor cantidad de evidencias empíricas para establecer con seguridad los efectos de TDD.

- ❑ Nuevas replicaciones con mejoras en la instrumentación de la experimentación.
- ❑ Realización de replicaciones en la industria.

# GRACIAS POR SU ATENCION...!