



XVII Escuela Internacional de Informática

**Ingeniería del software basada en la
evidencia (IBSE)
(PARTE I)**

Marcela Genero Bocco

Grupo Alarcos

marcela.genero@uclm.es

Universidad de Castilla-La Mancha

Escuela Superior de Informática

Departamento de Tecnologías y Sistemas de Información

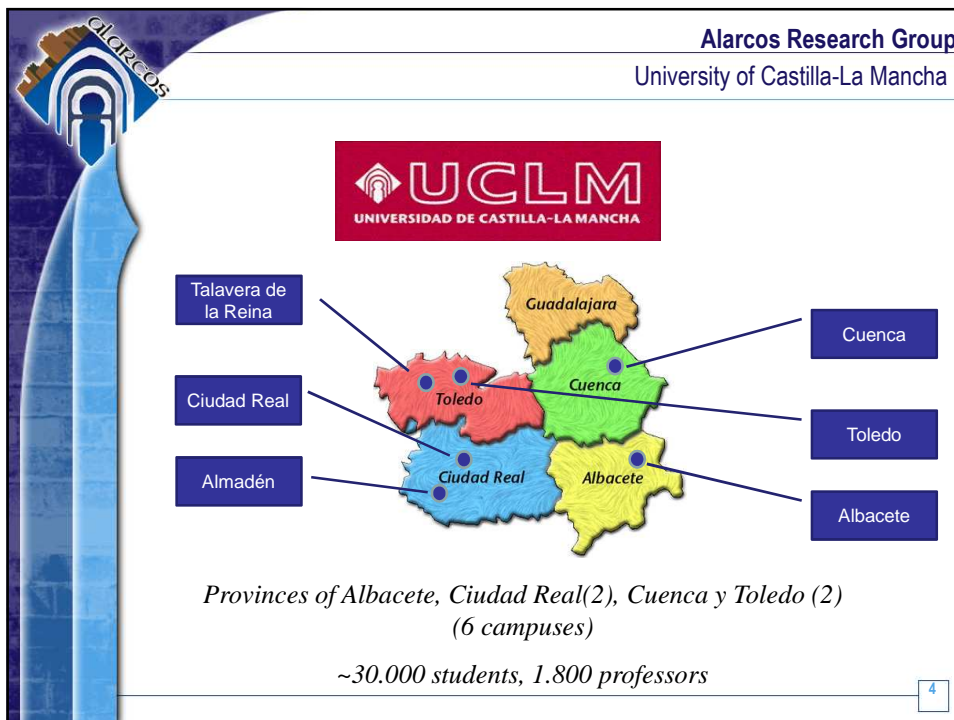
Ciudad Real - España




**ALARCOS RESEARCH GROUP
(1997-2013)**


<http://alarcos.esi.uclm.es>





Alarcos Research Group
University of Castilla-La Mancha





Escuela Superior de Informática
(School of Computer Science)



TOLEDO

Escuela Superior de Informática

Ciudad Real

1000+ students, 65 professors
BSc, MSc., and Ph.D. courses in CS

5

Alarcos Research Group
Led by ...





Prof. Mario Piattini

Table 2
Top scholars in the field of systems and software engineering (2004–2008).

Rank	Scholar	Institution
1	Magne Jørgensen	Simula Research Laboratory, Norway
2	Shih-Chien Chou	National Dong Hwa University, Taiwan
3	Hai Zhuge	Chinese Academy of Sciences, China
4	Lefteris Angelis	University of Thessaloniki, Greece
5	Lionel Briand	Simula Research Laboratory, Norway
5	Barbara Kitchenham	Keele University, UK
7	Chin-Yu Huang	National Tsing-Hua University, Taiwan
8	Chin-Chen Chang	National Chung Cheng University, Taiwan
8	Atif M. Memon	University of Maryland, USA
10	Chin-Wan Chung	Korean Advanced Institute of Science & Technology, Korea
11	Mark Harman	King's College London, UK
12	James Miller	University of Alberta, Canada
13	Hans van Vliet	Vrije Universiteit, Netherlands
13	Amrit Tiwana	Iowa State University, USA
13	Mario Piattini	University of Castilla-La Mancha, Spain

Wong et al. (2011). Wong, W., Tse, T., Glass, R., Basili, V., Chen, T. (2011). An assessment of systems and software engineering scholars and institutions (2003-2007 and 2004-2008). Journal of Systems and Software, 84(1), 162-168. Journal of Systems and Software, 84(1), 162-168.

6

Alarcos Research Group
Members

Alarcos Research Group: Full time Professors



Manuel Serrano



Prof. Mario Piattini



Marcela Genero



Macario Polo



Marian Moraga



Eduardo Fernández-Medina



Aurora Vizcaino



Felix Garcia



David Garcia



Ismael Caballero



José Cruz-Lemus



Prof. Coral Calero



Ignacio Garcia



Prof. Francisco Ruiz

7

Alarcos Research Group
Members

- 17 professors
 - 14 (Full-time) PhD. in Computer Science
 - **Dr. Mario Piattini (Full Professor)**
 - Dr. Ismael Caballero
 - Dr. Coral Calero (Full Professor)
 - Dr. José Antonio Cruz Lemus
 - Dr. Eduardo Fernández-Medina
 - Dr. Félix García
 - Dr. David García
 - Dr. Ignacio García Rodríguez de Guzmán
 - Dr. Marcela Genero
 - Dr. M. Ángeles Moraga de la Rubia
 - Dr. Macario Polo
 - Dr. Francisco Ruiz (Full Professor)
 - Dr. Manuel Ángel Serrano
 - Dra. Aurora Vizcaíno
- 40 PhD thesis defended
- 10 PhD students

8

Alarcos Research Group
Lab- ESI



ESCUOLA SUPERIOR DE INFORMATICA



18/01/2006

9

Alarcos Research Group
Lab- ITSI



UCLM
INSTITUTO DE TECNOLOGÍAS Y SISTEMAS DE LA INFORMACIÓN




10

Alarcos Research Group

Teaching

- Databases
- Software Engineering
- Information Systems
- Programming
- Project Management
- Software quality
- Research Methods
- Testing
- Security
- Business process
- Knowledge Management



Ph.D. Courses

11

Alarcos Research Group

Research objectives



12



Alarcos Research Group
Research areas

- **Software maintenance and reengineering:** Macario Polo, Ignacio García, Francisco Ruiz
- **Quality in model-based development:** Marcela Genero , José A. Cruz-Lemus
- **Data quality:** Ismael Caballero
- **Software Sustainability/Web quality:** Coral Calero, M. Ángeles Moraga
- **IS security:** Eduardo Fernández-Medina , David García
- **Software and business processes:** Francisco Ruiz, Félix García
- **Software testing:** Macario Polo
- **Global software development / knowledge management:** Aurora Vizcaíno
- **Research methods:** Marcela Genero; José A. Cruz-lemus
- **Software quality modelling and measurement:** M. Angeles. Moraga, Coral Calero, Marcela Genero, Manuel Serrano

13



Alarcos Research Group
PUBLICATIONS: Journals






























14

Alarcos Research Group
Conferences



MODELS

UML

CAISE

ER

ESEM

PROFES

SEKE

METRICS

ICSM

ICEIS

ISESE

CSMR

ICSR

IQ

WISE

CLEI

ICCI

QSIC

IDEAS

BPMN

CIBSE

ASSE

Objetivos

- ¿Qué es y porqué es relevante la **IBSE**?
- ¿Qué **estrategias** existen para **recopilar evidencia** en la Ingeniería del Software?
- ¿Cómo **validar/evaluar/comparar** **tecnologías software** en Ingeniería del Software?
- ¿Cómo realizar **experimentos**?
- ¿Cómo realizar **revisiones sistemáticas de la literatura**?
- Proporcionar **ejemplos relevantes**
- Proporcionar **literatura relevante**

16



XIX Congreso Argentino de Ciencias de la Computación




XVII Escuela Internacional de Informática

Ingeniería del software basada en la evidencia (IBSE) (PARTE I)

Marcela Genero Bocco
Grupo Alarcos
marcela.genero@uclm.es

Universidad de Castilla-La Mancha
Escuela Superior de Informática
Departamento de Tecnologías y Sistemas de Información
Ciudad Real - España




Ingeniería del software basada en la evidencia

Introducción

“Sólo trabajando con **evidencias** científicas en lugar de con suposiciones, el desarrollo de software se convertirá en una **verdadera** disciplina de **ingeniería**”

18



Ingeniería del software basada en la evidencia

Introducción


Objetivo de la IBSE:

“proporcionar los medios por los que la mejor evidencia actual de la investigación se pueda integrar con la experiencia práctica y los valores humanos en el proceso de la toma de decisiones sobre el desarrollo y mantenimiento de software”

*Barbara A. Kitchenham, Tore Dybå, Magne Jørgensen:
Evidence-Based Software Engineering. ICSE 2004: 273-281*

*Tore Dybå, Barbara A. Kitchenham, Magne Jørgensen:
Evidence-Based Software Engineering for Practitioners. IEEE
Software 22(1): 58-65 (2005)*

19




Ingeniería del software basada en la evidencia

Introducción

La IBSE puede proporcionar:

- Un objetivo común para los investigadores y grupos de investigación para asegurar que su **investigación** se dirige a las **necesidades de la industria** y otros grupos interesados.
- Un medio por el cual los profesionales de la industria puedan **tomar decisiones racionales** sobre la adopción de tecnología.
- Un medio para **mejorar la fiabilidad** de los sistemas intensivos en **software**, como resultado de una mejor elección de las tecnologías de desarrollo.
- Un medio para aumentar la **aceptabilidad** de los sistemas intensivos de **software** que interactúan con los **ciudadanos**.
- Una entrada a los procesos de **certificación**.


20

 **Ingeniería del software basada en la evidencia**
Introducción


¿Por qué es importante impartir enseñanza sobre IBSE?

- Proporciona a los futuros profesionales el conocimiento, la experiencia, la actitud y las habilidades que les permitan tomar mejores decisiones.
- Es importante enseñar la IBSE:
 - La **industria del software se beneficiará** de moverse en la dirección de la IBSE.
 - Las universidades deben tener un mayor enfoque en **enseñar la forma de adquirir nuevos conocimientos y habilidades**.
 - La IBSE puede apoyar la **evaluación crítica** y sistemática de argumentos. Permite una evaluación más crítica de los métodos de desarrollo y tácticas de marketing.

21


 **Ingeniería del software basada en la evidencia**
Introducción

- **No es posible obtener resultados válidos universalmente**
- **No tienen sentido las preguntas universales**
"¿Qué es mejor: programación en solitario o programación por pares? ¿Probar al inicio o al final?"



Contextualizar la evidencia
¿Qué funciona para **quién**, **dónde**, **cuándo**, y **por qué**?

22




Ingeniería del software basada en la evidencia

Métodos para recopilar evidencia en la IS

- **Metodos primarios:** utilizados para realizar estudios con el objetivo de obtener evidencia empírica sobre un tema de interés.
 - Experimentos
 - Estudios de casos
 - Encuestas
- **Métodos secundarios:** permiten sintetizar la evidencia obtenida en estudios primarios realizados.
 - Revisiones sistemáticas de la literatura (SLRs)
 - Mapeos sistemáticos de la literatura (SMSs)
 - Estudios terciarios


23



Contenido


1. Estudios empíricos en la Ingeniería del Software
2. Revisiones sistemáticas de la literatura

24



Estudios empíricos en la Ingeniería del Software (IS)

25



Contenido

- **Introducción**
- Tipos de estudios empíricos
- Experimentos
 - Conceptos generales
 - Proceso experimental
- Ejemplo
- Réplicas
- Agregación de resultados
- Recomendaciones
- Conclusiones
- Literatura relevante

26

Introducción

- Los **estudios empíricos** permiten crear conocimiento que puede mejorar la práctica de la IS
 - Aplicar los conceptos y técnicas en la práctica de la IS
 - Añadir experiencias sobre su aplicación al "cuerpo de conocimiento" de la IS

Teoría, Cuerpo de conocimiento de IS
Práctica, Proyectos de IS


27

Introducción

Objetivos de los estudios empíricos

- Construir conocimiento y comprensión sobre la IS en la práctica
 - Para usar en futuros proyectos
 - Para comparación (benchmarking)
- Evaluar nuevas tecnologías
 - Por ej.: ¿Mejora una nueva técnica/métodos la calidad?
 - Comprender en qué contexto funciona mejor una técnica

28



Introducción
Temas de los estudios empíricos

Personas

- personalidad de los desarrolladores y su impacto en ...

Procesos

- Ágiles, cascada, RUP, CMM, ...


Prácticas

- Programación por pares, ...

Tecnología

- OO vs. Non-OO, ...

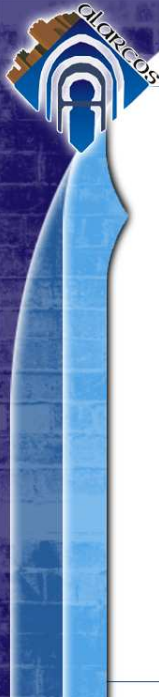
29






Introducción
Objetivos de los estudios empíricos

- Para probar (o refutar) teorías, mitos, creencias populares
 - Por ejemplo: “Creo que el método A es mejor que el B”
- Encontrar relaciones
 - Por ejemplo: Relación entre la mantenibilidad y LoC
- Para elegir entre modelos/técnicas/herramientas:
 - Por ejemplo: UML y ER para el modelado de datos

30


Contenido	
	<ul style="list-style-type: none"> • Introducción • Tipos de estudios empíricos • Experimentos <ul style="list-style-type: none"> ▪ Conceptos generales ▪ Proceso experimental • Ejemplo • Réplicas • Agregación de resultados • Recomendaciones • Conclusiones • Literatura relevante
	31

Tipos de estudios empíricos	
 	Encuestas
	<p>¿Cuándo son apropiadas?: para un análisis exploratorio temprano, se suelen hacer en retrospectiva cuando una técnica/herramienta ha sido utilizada durante un tiempo. Permite descubrir factores de influencia, diferencias y similitudes, etc.</p> <p>Nivel de control: bajo</p> <p>Recolección de datos: (online) cuestionarios, encuestas</p> <p>Análisis de datos: comparar diferentes poblaciones entre los encuestados, análisis de asociación y de tendencias, la consistencia de las puntuaciones</p> <p>Pros: la generalización de los resultados es más fácil (que en los estudios de casos), aplicable en la práctica</p> <p>Cons: poco control de variables, el diseño del cuestionario es difícil (validez, fiabilidad), la búsqueda de encuestados puede ser un reto, necesita un gran número de encuestados, la ejecución puede tardar más tiempo.</p>
	32



Tipos de estudios empíricos

Encuestas



Available online at www.sciencedirect.com

SCIENCE @ DIRECT®

Information and Software Technology 47 (2005) 383–397

www.elsevier.com/locate/inftel

Does UML make the grade? Insights from the software development community

Martin Grossman^{a,*}, Jay E. Aronson^{b,1}, Richard V. McCarthy^{c,2}

^aDepartment of Management, Bridgewater State College, Bridgewater, MA 02325, USA
^bDepartment of Management Information Systems, Terry College of Business, University of Georgia, Athens, GA 30602, USA
^cLeader School of Business, Quinnipiac University, Hamden, CT 06518, USA

Received 25 January 2004
Available online 11 November 2004

Abstract


The Unified Modeling Language (UML) has become the de facto standard for systems development and has been promoted as a technology that will help solve some of the longstanding problems in the software industry. However, there is still little empirical evidence supporting the claim that UML is an effective approach to modeling software systems. Indeed, there is much anecdotal evidence suggesting the contrary, i.e. that UML is overly complex, inconsistent, incomplete and difficult to learn. This paper describes an investigation into the adoption and use of UML in the software development community. A web-based survey was conducted eliciting responses from users of

Encuesta realizada via web sobre la adopción y el uso de UML en la industria

1. Introduction


composed of notations and textual components to express object-oriented system designs [16]. During the early 1990s,

33



Tipos de estudios empíricos

Estudios de casos



¿Cuándo son apropiados?: el fenómeno bajo estudio no se puede separar del contexto; queremos evaluar el cambio en una situación típica. Son estudios observacionales, se realizan generalmente en entornos industriales.

Nivel de control: medio



Recolección de datos: medidas de productos y de procesos, cuestionarios, entrevistas

Análisis de datos: generalmente cualitativo, pero también puede ser cuantitativo

Pros: aplicable en proyectos reales, ayudan a contestar preguntas del tipo “porque”, “como” pueden generar nuevas

Cons: la generalización es a menudo difícil y cuestionada por los revisores, el impacto de los factores de confusión es difícil evaluar / aislar, el análisis de los resultados es subjetivo, por lo general es una labor muy intensa en la recolección de datos.

34

Tipos de estudios empíricos

Estudio de casos

Empir Software Eng (2006) 11: 203–225
DOI 10.1007/s10664-006-6402-8

Integrating agile software development into stage-gate managed product development

Daniel Karlström · Per Runeson

© Springer Science + Business Media, Inc. 2006
Editor: Martin Zalkowitz



Abstract Agile methods have evolved as a bottom-up approach to software development. However, as the software in embedded products is only one part of development projects, agile methods must coexist with project management models typically of the stage-gate type. This paper presents a qualitative case study of two large independent software system projects that have used eXtreme Programming (XP) for software development within contexts of stage-gate project management models. The study is comprised of open ended interviews with managers as well as practitioners, followed by a structured, fully traceable, qualitative analysis. We conclude that it is possible to integrate XP in a gate model context. Key issues for success are the interfaces towards the agile subproject and management attitudes towards the agile approach.

Keywords Agile methods · Stage-gate project management · Qualitative study · Extreme programming · Case studies

Integración de XP con modelos de gestión tradicionales (Ericsson y ABB)

Kähkönen, 2003; Fuqua and Hammer, 2003; Rasmussen, 2003; Schuh, 2001; Grenning, 2001; Poole and Huisman, 2001; Murru et al., 2003; Sharp and Robinson, 2004; Karlström, 2002). The agile methodologies offer down-to-earth approaches to software development that focus on simplifying and improving the software development process within the customer, the developer and the end-product etc.

35

Tipos de estudios empíricos

Experimentos

¿Cuándo son apropiados?: control sobre quien esta usando una tecnología, cuándo, dónde y bajo que circunstancias. Investigación de las tareas autónomas, donde se pueden obtener resultados de inmediato. Se realizan off-line (en laboratorios)

Nivel de control: alto


Recolección de datos: medición de productos y procesos, cuestionarios

Análisis de datos: estadísticos descriptivos, de test de hipótesis

Pros: ayudan a establecer relaciones causales, confrmimar teorías

Cons: representatividad? Es un reto planificarlos en en un entorno real, la aplicación en el contexto industrial requiere compromisos

36




Tipos de estudios empíricos

Experimentos

IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, VOL. 32, NO. 6, JUNE 2008 305

The Impact of UML Documentation on Software Maintenance: An Experimental Evaluation

Erik Arisholm, *Member, IEEE*, Lionel C. Briand, *Senior Member, IEEE*,
Siv Elisabeth Hove, and Yvan Labiche, *Member, IEEE*



Abstract—The Unified Modeling Language (UML) is becoming the de facto standard for software analysis and design modeling. However, there is still significant resistance to model-driven development in many software organizations because it is perceived to be expensive and not necessarily cost-effective. Hence, it is important to investigate the benefits obtained from modeling. As a first step in this direction, this paper reports on controlled experiments, spanning two locations, that investigate the impact of UML documentation on software maintenance. Results show that, for complex tasks and past a certain learning curve, the availability of UML documentation may result in significant improvements in the functional correctness of changes as well as the quality of their design. However, there does not seem to be any saving of time. For simpler tasks, the time needed to update the UML documentation may be substantial compared with the potential benefits, thus motivating the need for UML tools with better support for software maintenance.

Index Terms—Maintenance, UML, experiment.


1 INTRODUCTION

SOFTWARE maintenance is often performed by individuals who were not involved in the original design of the system being changed. This is why documenting software specifications and designs often has been advocated as a necessity to help software engineers remain in intellectual control while changing complex systems [8], [12]. Indeed, it is expected that many aspects of a software system need to

Unified Modeling Language (UML) [8], an evolving standard that is now widespread across the software industry. However, despite its growing popularity, there is little reported evaluation of the use of UML-based development methods [1] and many perceive the documentation of analysis and design models in UML to be a wasteful activity [7]. Hence, such practices are viewed as

¿Ayuda el uso de los diagramas UML a realizar el mantenimiento del software mejor y más rápido?




37



Tipos de estudios empíricos

Comparación

- **Control de la ejecución.** Cantidad de control que el investigador tiene sobre el estudio
- **Control de la medición.** Nivel en que el investigador puede decidir sobre que medidas deben ser recolectadas y ser incluidas o excluidas durante la ejecución del estudio (en una encuesta sólo tenemos opiniones no atributos medibles directamente)
- **Coste de la investigación.** Relativo a la necesidad de recursos.
- **Facilidad de réplica.** Facilidad de volver a llevar a cabo el mismo estudio bajo las mismas condiciones pero con distinta población

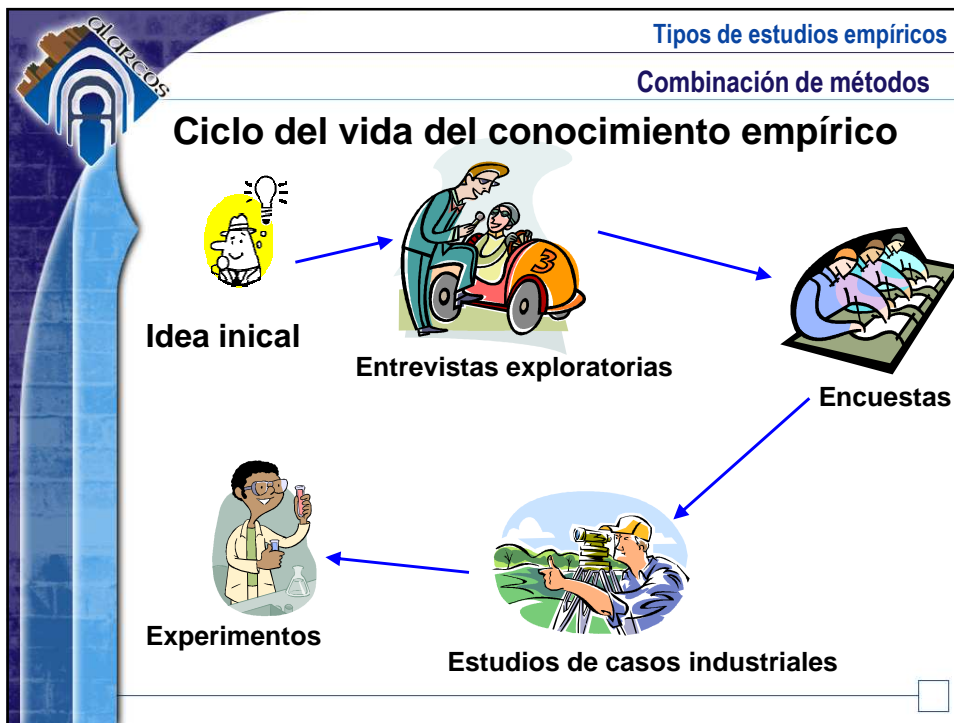





38

Tipos de estudios empíricos
Comparación

Factor	Encuesta	Estudio de caso	Experimento
Control de ejecución	NO	NO	SI
Control de medición	NO	SI	SI
Coste de investigación	BAJO	MEDIO	ALTO
Facilidad de réplica	ALTO	BAJO	ALTO

39







Contenido

- Introducción
- Tipos de estudios empíricos
- **Experimentos**
 - **Conceptos generales**
 - Proceso experimental
- Ejemplo
- Réplicas
- Agregación de resultados
- Recomendaciones
- Conclusiones
- Literatura relevante

41





Experimentos
Conceptos generales

Un experimento es una investigación empírica que investiga las **relaciones de causa-efecto**. Los experimentos se llevan a cabo cuando el investigador quiere el control de la situación, con la manipulación directa, precisa y sistemática del comportamiento del fenómeno a estudiar

Ventaja: Permiten investigar las situaciones en las que las hipótesis son verdaderas y proporcionan el contexto en el que se recomiendan normas, métodos y herramientas para su uso.

42






Tipos de estudios empíricos
Experimentos

Orientados a personas:
Un experimento en el cual los individuos o los equipos llevan a cabo una o más tareas de ingeniería de software para comparar diferentes poblaciones, procesos, métodos, técnicas, lenguajes o herramientas.
Por ejemplo se quiere comparar el desarrollo “tradicional vs. ágil” del software

Orientados a la tecnología:
Generalmente se aplican diferentes herramientas a diferentes objetos.
Por ejemplo al mismo programa se le aplican dos técnicas de generación de casos de prueba diferentes.

43

Experimentos
Conceptos generales

Variable independiente (factor-tratamiento)
Tecnología



Variable dependiente (salida)
Tiempo, costes, calida

Efecto

Otras variables independientes (controladas)
(experiencia de los sujetos, complejidad de los tareas y de los objetos, etc.)

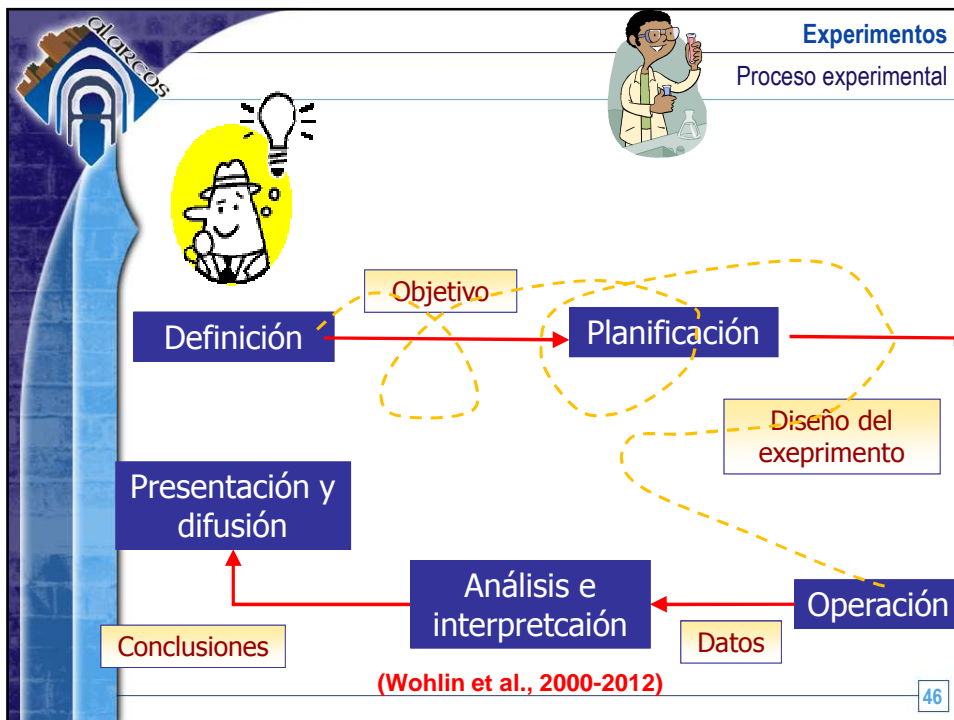
44

Contenido

- Introducción
- Tipos de estudios empíricos
- **Experimentos**
 - Conceptos generales
 - **Proceso experimental**
- Ejemplo
- Réplicas
- Agregación de resultados
- Recomendaciones
- Conclusiones
- Literatura relevante

45



Experimentos
Proceso Experimental: Definición

¿Por qué se realiza el experimento?

Plantilla GQM para definición de objetivos:

Analizar **Método orientado a objetos y estructurado**
(¿qué es lo que se estudia?)

con el propósito de **Evaluar**
(¿cuál es la intención?)

con respecto a **Productividad**
(¿cuáles es el efecto estudiado?)

desde el punto de vista del **Investigador**
(¿a la vista de quién?)

en el contexto de **Estudiantes de grado**
(¿dónde se lleva a cabo el estudio?)

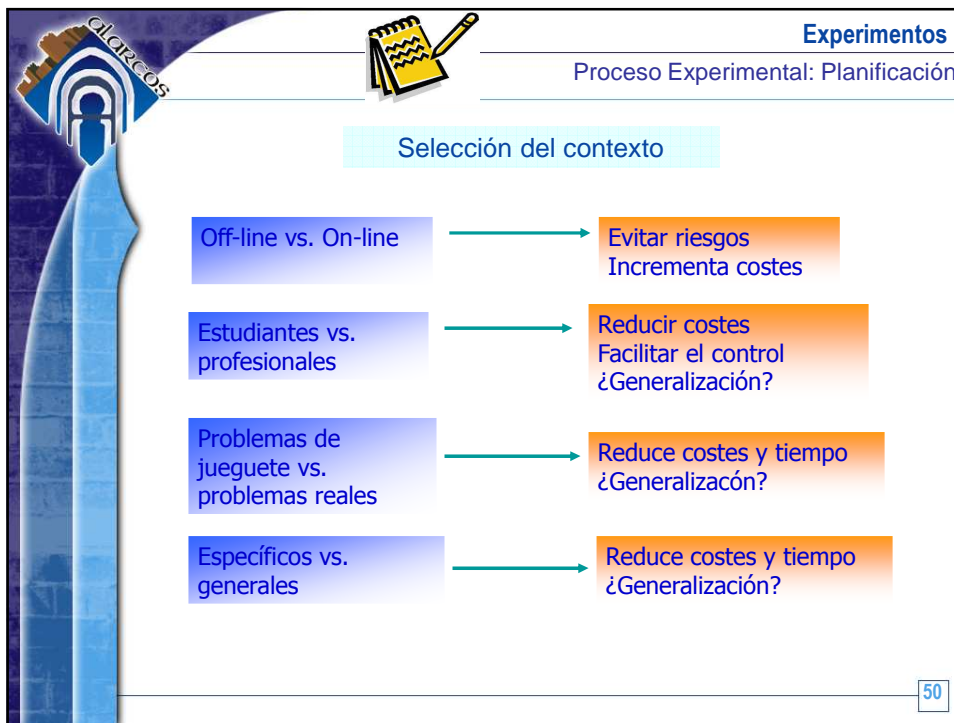
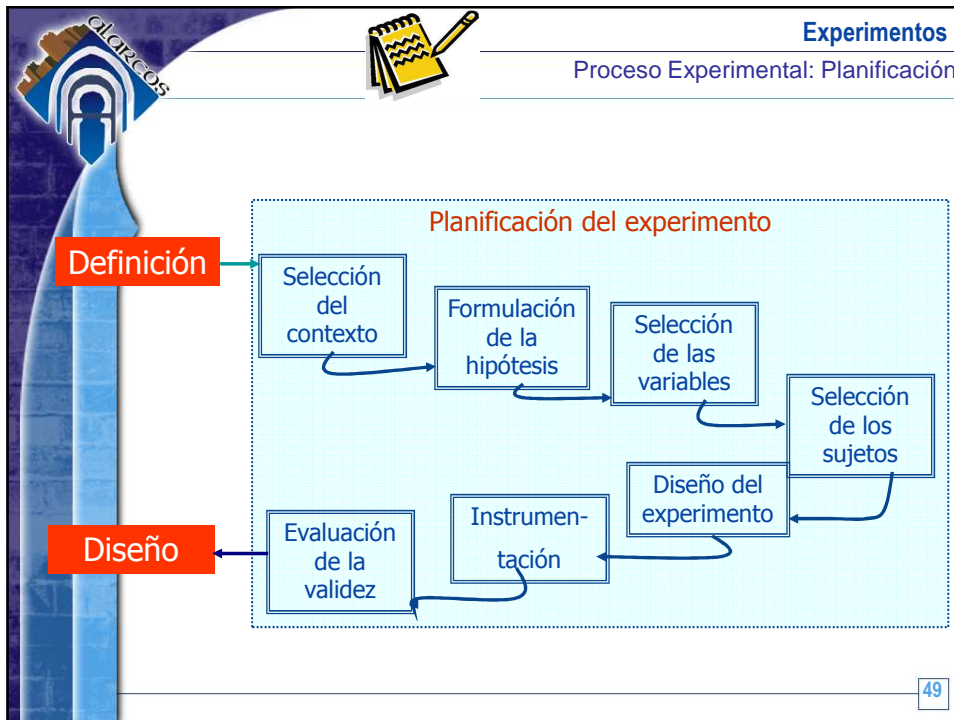
47

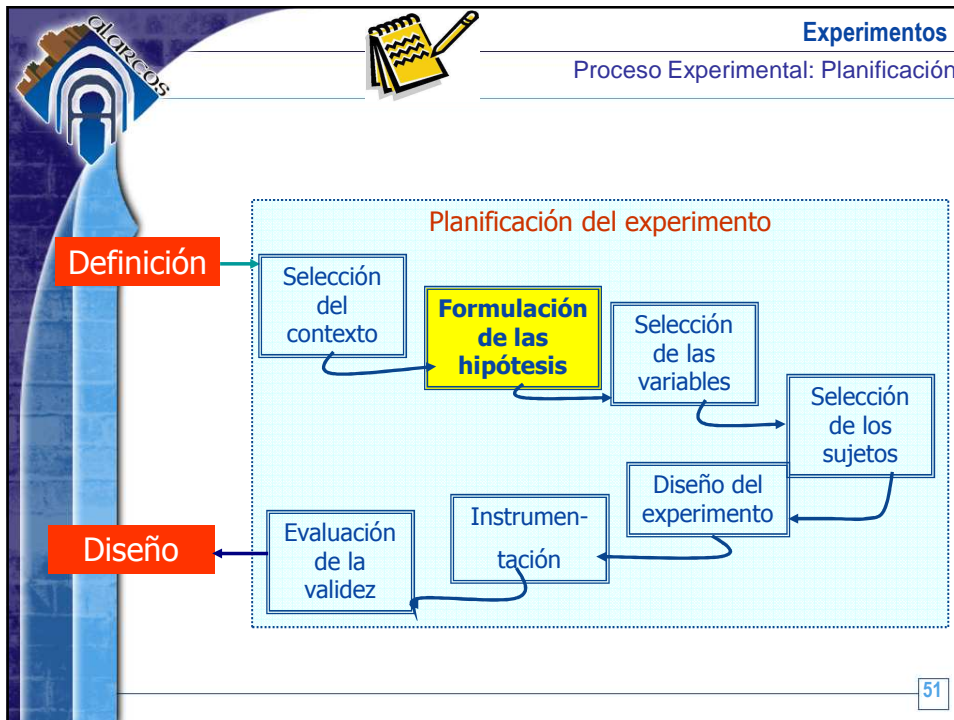
Experimentos
Proceso Experimental: Definición

Ejemplos de definición de objetivos

Objeto de estudio	Propósito	Enfoque de calidad	Perspectiva	Contexto
Producto	Caracterizar	Efectividad	Desarrollador	Sujetos
Proceso	Supervisar	Costo	Mantenedor	Objetos
Modelo	Evaluar	Fiabilidad	Director de proyectos	
Métrica	Predecir	Mantenibilidad	Cliente	
Teoría	Controlar	Portabilidad	Usuario	
	Cambiar		Investigador	

48





Experimentos
Proceso Experimental: Planificación

Formulación de la hipótesis

Se deriva del objetivo: una o más H_0

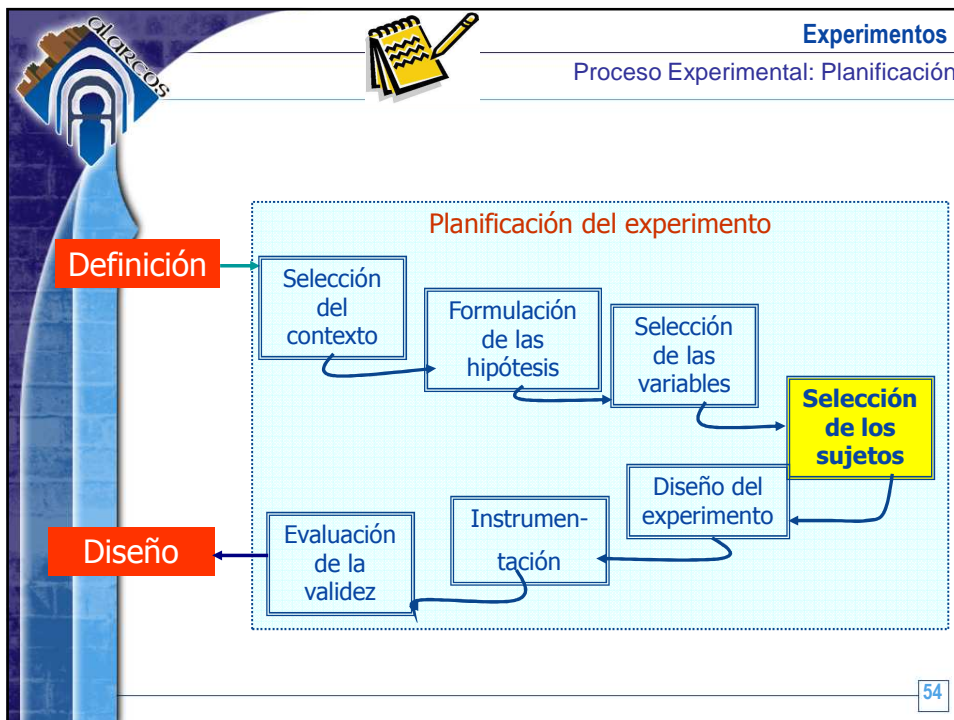
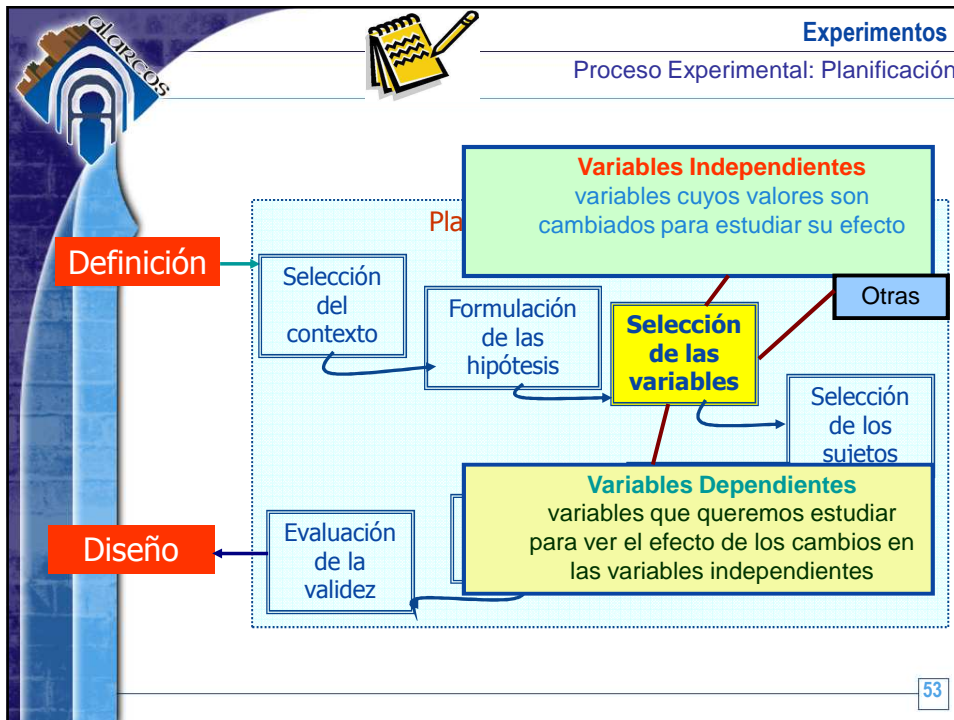
Plantilla GQM

- Analizar *métodos OO y estructurado*
- Con el propósito de *Evaluar* con respecto a su *productividad*
- Desde el punto de vista del *investigador*
- En el contexto de *estudiantes de grado*

H_0 : Productividad OO = Productividad OP
(*conservadora, no existe efecto del tratamiento*)

H_1 : Productividad OO \neq Productividad OP
(*alternativa*)

52



Experimentos
Proceso Experimental: Planificación

Selección de los Sujetos

- ¿Cómo seleccionar los sujetos?
 - Muestras probabilísticas o no probabilísticas
 - Muestreo simple aleatorio, sistemático, estratificado aleatorio
 - Muestreo por conveniencia, por cuota
- ¿Y el tamaño de la muestra?
 - Si hay gran variabilidad en la población se necesita un muestra de mayor tamaño
 - La técnica de análisis puede influenciar la elección del tamaño de la muestra

La muestra debe ser representativa de la población

55

Experimentos
Proceso Experimental: Planificación

Planificación del experimento

```

    graph LR
      subgraph Definición
        A[Selección del contexto] --> B[Formulación de las hipótesis]
        B --> C[Selección de las variables]
      end
      subgraph Diseño
        D[Evaluación de la validez] --> E[Instrumentación]
        E --> F[Diseño del experimento]
      end
      C --> G[Selección de los sujetos]
      F --> G
      G --> E
  
```

56

Experimentos
Proceso Experimental: Planificación

Diseño del Experimento

- Describe cómo se organizan los tests
- Relacionado con el análisis, interpretación y conclusiones del experimento
- Preguntas Relevantes:
 - ¿Cuántas variables independientes?
 - Sólo una → Experimentos simples
 - Más de una → Experimentos factoriales
 - ¿Cuántos tratamientos por sujeto? Medidas Repetidas
 - ¿Cómo controlar los factores extraños? Bloqueo / Aleatorización
- Estas respuestas dependen de las amenazas a la validez que queramos controlar

57

Experimentos
Proceso Experimental: Planificación

Diseño del Experimento: Taxonomía

Único Tratamiento por Sujeto

Simple
(1 Variable Ind)

Se asignan todos los Tratamientos por Sujeto

Factoriales
(>= 2 Variables Ind)

Inter-sujetos (IES) { Grupos aleatorios
Grupos aleatorios con bloques

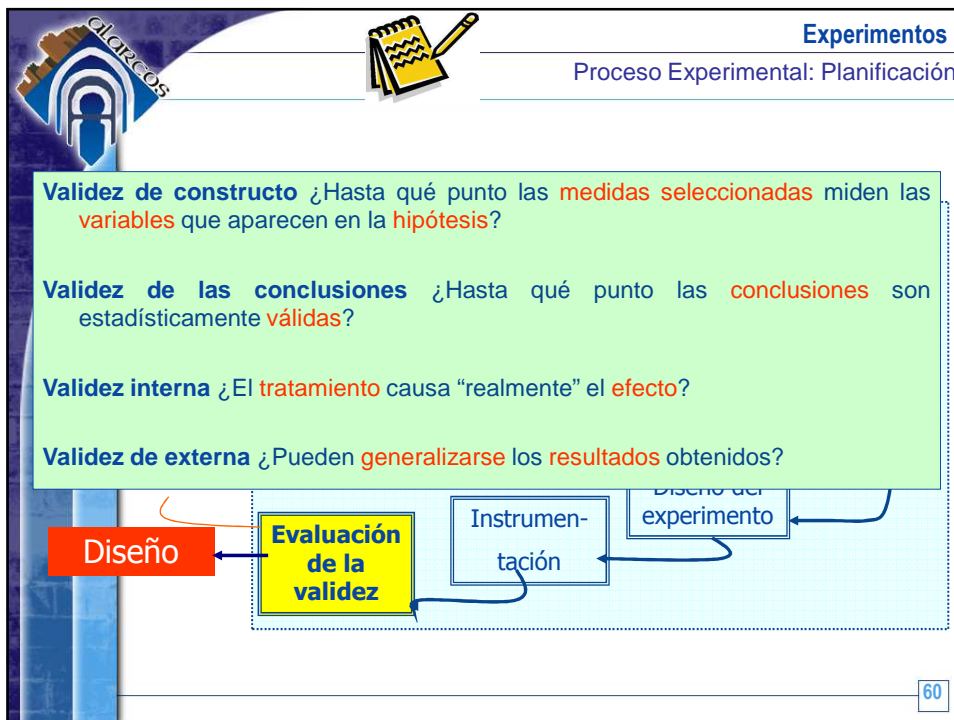
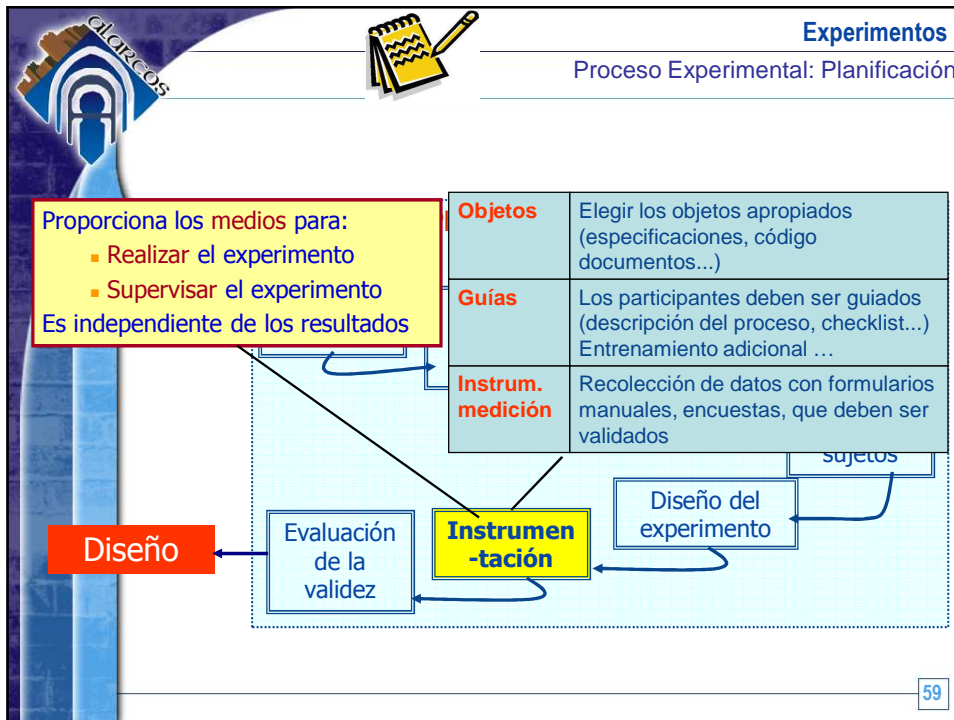
Intra-sujetos (IAS) { Aleatorización
Reequilibrado
...


Completos (2 factores)

Fraccionales Parciales

Factorial		Factor A	
		M1	M2
Factor B	Herr	a1 b1	a1 b2
	NHerr	a2 b1	a2 b2

58





  **Experimentos**
Proceso Experimental: Planificación

Evaluación de la validez: **amenazas a la validez interna**

Historia	Tratamientos diferentes aplicados sobre el mismo objeto en momentos diferentes... ¿las circunstancias son las mismas?
Maduración	Los sujetos reaccionan cuando el tiempo pasa (cansancio, aburrimiento, aprendizaje)
Selección	¿La muestra de la población es representativa para toda la población?
Instrumentación	¿Se han diseñado correctamente los artefactos utilizados en el experimento? Documentos que van a ser inspeccionados ..
Mortalidad	Personas que abandonan el experimento.
Orden de presentación	¿Cuánto conoce el sujeto sobre el tratamiento? Se altera para disminuir efectos de aprendizaje.

61

  **Experimentos**
Proceso Experimental: Planificación

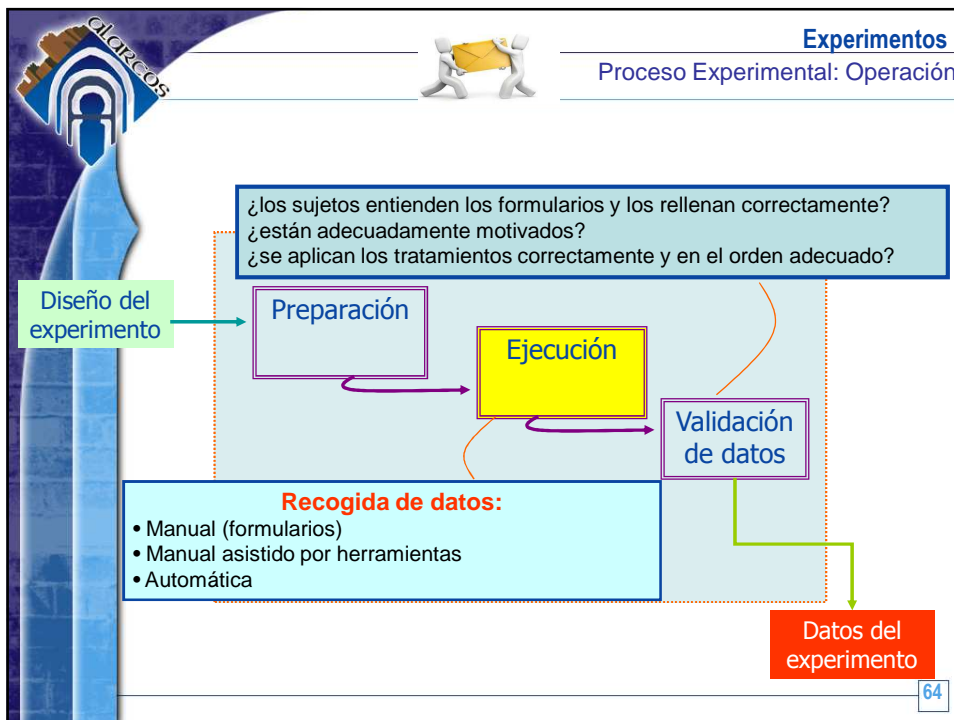
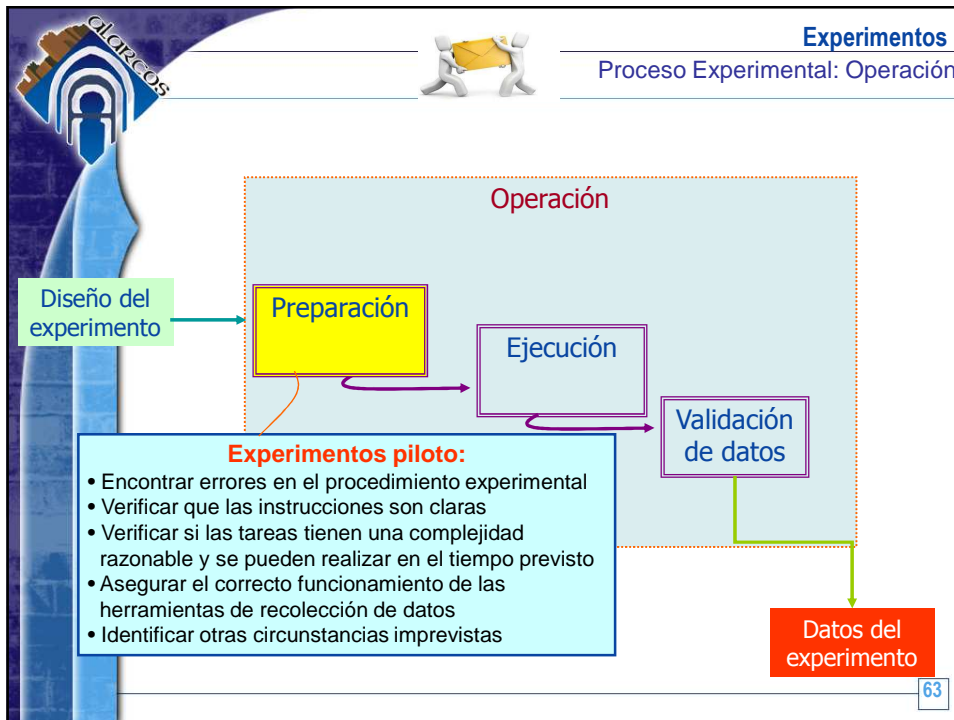
Evaluación de la validez: **amenazas a la validez externa**

- Interacción entre selección y tratamiento
 - La muestra no es representativa de la población
- Interacción entre el material y el tratamiento
 - Material no representativo
 - Problemas de juguete, métodos obsoletos
- Interacción entre la historia y el tratamiento
 - El experimento se lleva a cabo en un momento especial que afecta a los resultados

Evaluación de la validez: **amenazas a la validez de las conclusiones**

- Bajo poder estadístico, violación suposición tests
- Buscar o “pescar” un determinado resultado
-

62



Experimentos
Proceso Experimental: Análisis e Interpretación

Interpretación Cuantitativa

Datos experimentales

Estadísticos descriptivos

Reducción del conjunto de datos

Contraste de las hipótesis

Graphic Case Two

BPM	Answer Times	
	Understandability	Modifiability
1	188	308
2	165	438
3	192	443
4	213	1011
5	291	425
6	223	375
7	237	433
8	198	355
9	199	386
10	238	383

* Value higher

65

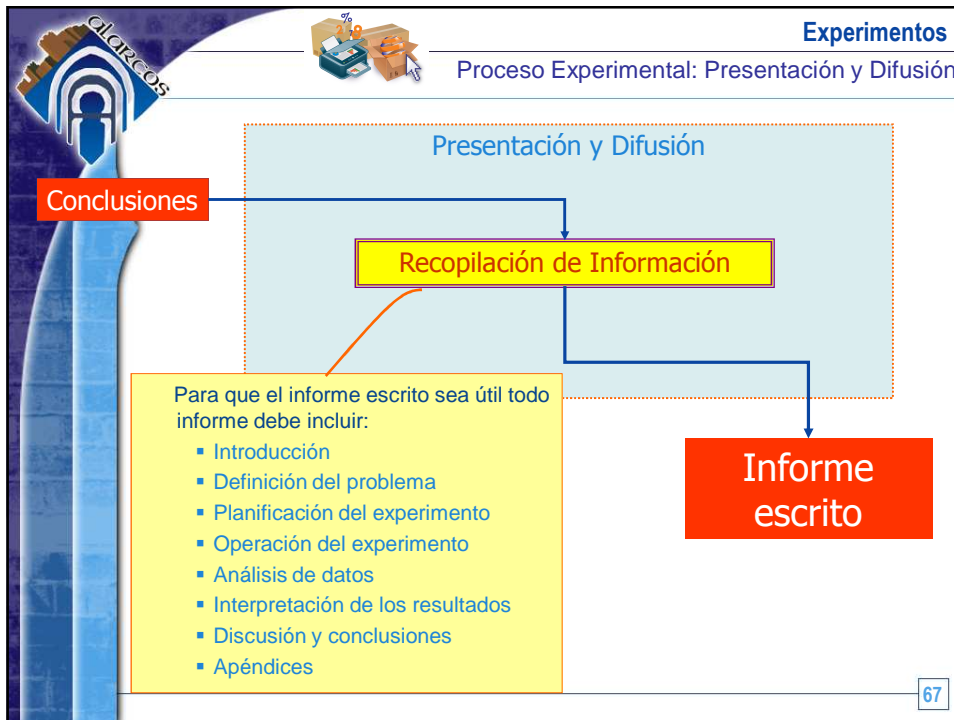
Experimentos
Proceso Experimental: Análisis e Interpretación

Contraste de Hipótesis

Diseño	Param.	No param.
Un factor, un tratamiento		Chi-2 Test binomial
Un factor, dos tratamientos	Test t Test F Test t emp	
Un factor, más de dos tratamientos	ANOVA	
Más de un factor	ANOVA	

- Otras Técnicas (IA) :
 - Árboles de clasificación, Clustering, Redes neuronales, "Machine Learning"

66




Contenido


- Introducción
- Tipos de estudios empíricos
- Experimentos
 - Conceptos generales
 - Proceso experimental
- **Ejemplos**
- Réplicas
- Agregación de resultados
- Recomendaciones
- Conclusiones
- Literatura relevante

68

Ejemplos

Temas sobre experimentos relacionados con UML





Budgen, D. Burn, A.J., Brereton, P., Kitchenham, B., Pretorius, R. (2011). **Empirical evidence about the UML: a systematic literature review**, Software Practice and Experience, 41(4): 363-392.

- 49 relevant publications published up to the end of 2008.

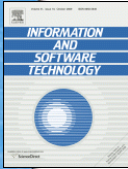
Genero, M., Fernández-Sáez, A.M., Nelson, H.J., Poels, G., Piattini, M. (2011). **A Systematic Literature Review on the Quality of UML Models**, Journal of Database Management, 22(3): 46-70.

- 266 papers published between 1997-2009.
- 30 % includes any kind of empirical study (24 % experiments, 5 % case studies, 1% surveys).

69

Ejemplos

Temas sobre experimentos relacionados con UML

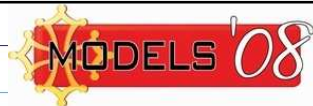


Ana M. Fernández-Sáez, Marcela Genero, Michel R. V. Chaudron. (2013). **Empirical studies concerning the maintenance of UML diagrams and their use in the maintenance of code: A systematic mapping study**. Information and Software Technology 55(7): 1119-1142 .

- 38 artículos, 1997-2010.
- 63 experimentos y 3 estudios de casos
- Sólo 3 → uso de UML en el mantenimiento
- 60 relacionados con la comprensión y modificación de los propios diagramas.

70

- **Impact of variants of notation on comprehension and modification**
- **Impact of quality of UML model on comprehension and modification**
- **Comparison of UML and ER on comprehension and modification**
- **Validation of measures as indicators of UML model comprehension and modification**
- **Benefits of using UML in software development life-cycle**

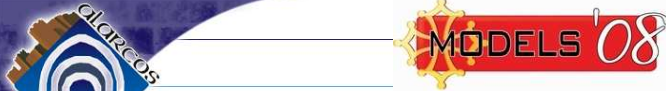


Assessing the influence of stereotypes on the comprehension of UML sequence diagrams: A controlled experiment.

- Genero, M., Cruz-Lemus, J.A., Caivano, D., Abrahão, S., Insfran, E., Carsí, J.A.
- 11th international conference on Model Driven Engineering Languages and Systems **MoDELS'08** (LNCS 5301 pp. 280-294).



Material available at:
<http://alarcos.esi.uclm.es/ExpStereotypes>




Motivation

The influence of the use stereotypes in the comprehension of class diagrams has been studied

- F. Ricca, M. Di Penta, M. Torchiano, P. Tonella, M. Ceccato, **Developers' experience and ability influence web application comprehension tasks supported by UML stereotypes: a series of four how experiments**, IEEE TSE 36 (1) (2010) 96–118.
- M. Staron, L. Kuzniarz, C. Wohlin, **Empirical assessment of using stereotypes to improve comprehension of UML models: a set of experiments**, The Journal of Systems and Software 79 (2006) 727–742

The influence of stereotypes on the comprehension of sequence diagrams has not been investigated yet.

73



Motivation

UML sequence diagrams → widely used technique for reasoning about object interactions.

UML sequence diagram comprehension → essential in the validation of requirements specifications among developers and stakeholders.

A set of stereotypes for UML sequence diagrams → proposed to enrich the semantics of interaction messages in the context of a Requirements Engineering approach for MDD (**signal, service, query, connect**).

74

calarcos

MODELS '08

Stereotypes defined for SD

- «**signal**» a message between an actor and the system.
- «**service**» a message that updates the state of an object. The properties *new/destroy/update* can be used if the object is to be created, destroyed or modified.
- «**query**» a message to query the state of a specific object or set of objects.
- «**connect**» a message which is used to indicate that an object from the sender object type needs to be connected (or related) to an object from the receiver object type.

75

calarcos

MODELS '08

Stereotypes for UML Sequence Diagrams

Items-Sale Use Case

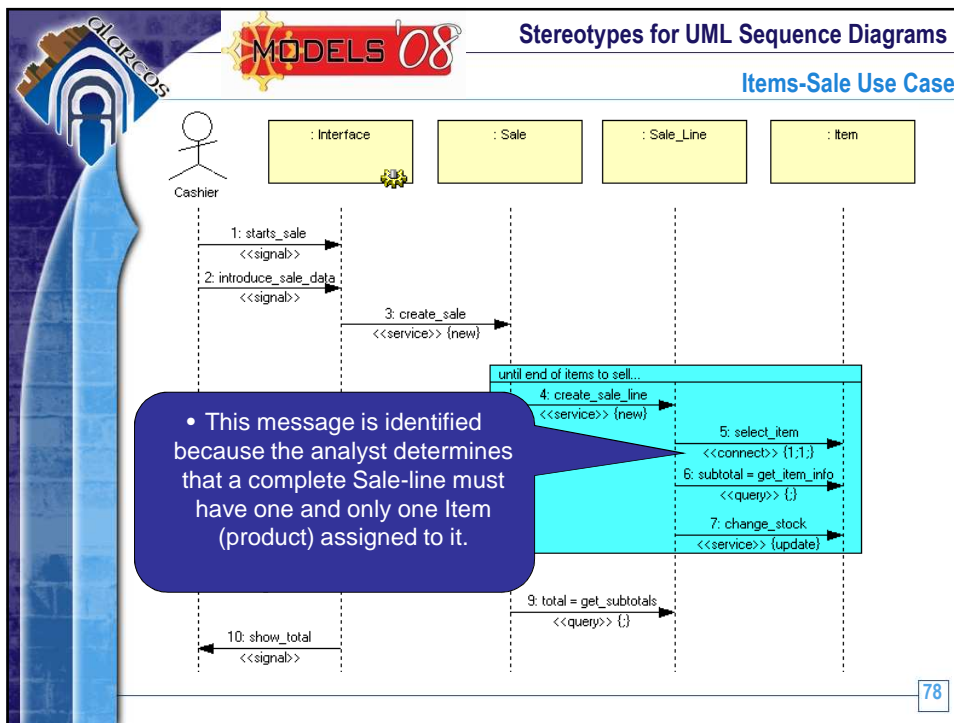
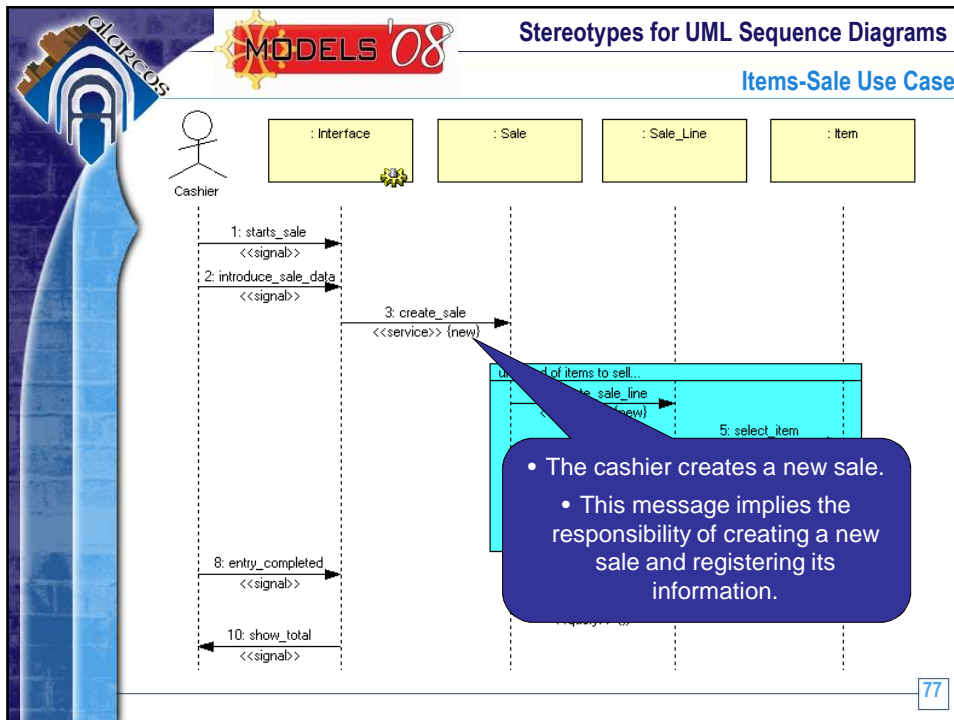
```

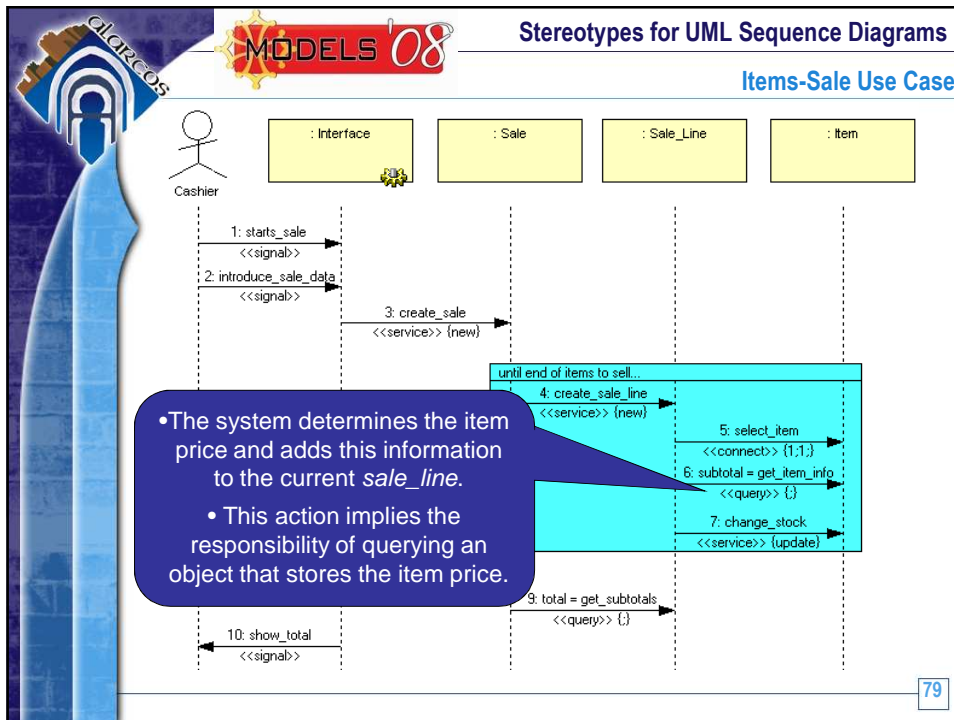
sequenceDiagram
    actor Cashier
    participant Interface
    participant Sale
    participant Sale_Line
    participant Item

    Cashier->>Interface: 1: starts_sale <<signal>>
    Cashier->>Interface: 2: introduce_sale_data <<signal>>
    Interface->>Sale: 3: create_sale <<service>>
    Sale->>Sale_Line: 4: add_item <<service>>
    Sale_Line->>Item: 5: create_item <<service>>
    Item->>Sale_Line: 6: change_stock <<query>>
    Sale_Line->>Sale: 7: change_stock <<service>>
    Sale->>Interface: 8: entry_completed <<signal>>
    Interface->>Cashier: 10: show_total <<signal>>
    Cashier->>Sale: 9: total = get_subtotals <<query>>
  
```

76

- The cashier starts a sale. This message represents the commencement of communication between the Cashier and the Interface of the software system.
- The message *introduce_sale_data* provides the data which is necessary to perform the operation.





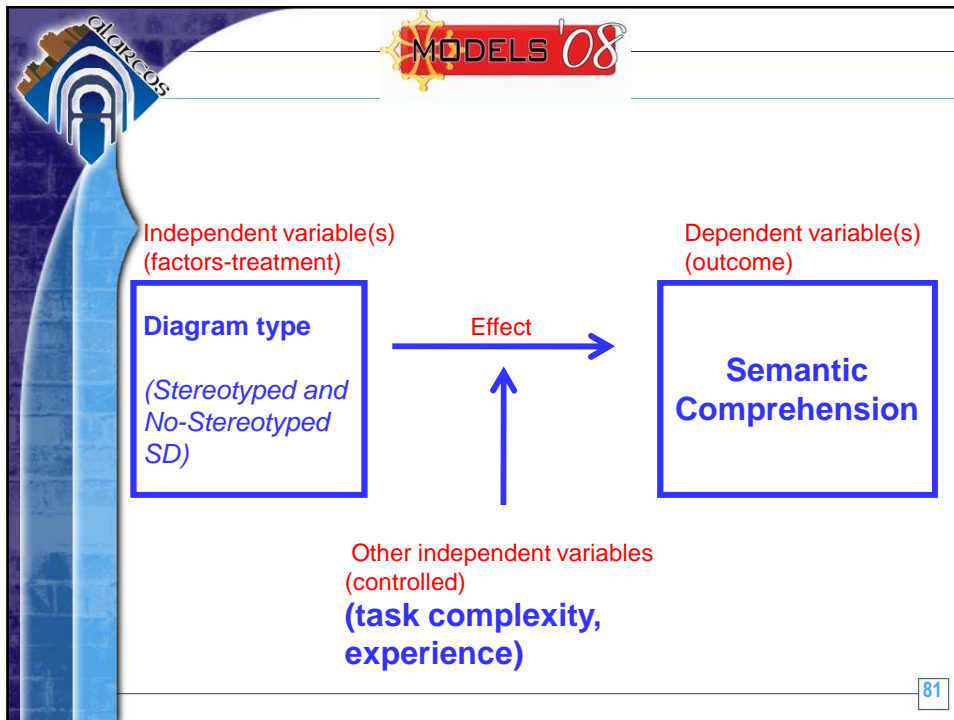
calarcos MODELS '08

We believe that the benefit of using these stereotypes is twofold :

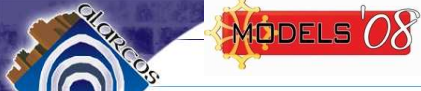
- provide specific information about how to deal with each individual source element in the **automatic transformation** of **UML sequence diagrams** into **class diagrams**, in the MDD process.
- **improve the comprehension of UML sequence diagrams** ???

Goal → to assess whether the use of stereotypes improves the understanding of UML sequence diagrams.

80



-
- Definition
- Using GQM template our goal is:
- *Analyze the use of stereotypes*
 - *for the purpose of evaluating them*
 - *with respect to the comprehension of UML sequence diagrams*
 - *from the point of view of the researcher*
 - *in the context of fourth year undergraduate students in Computer Science from the University of Bari.*
- 82



Planning


Selection of subjects. (students vs. professionals, probabilistic and non-probabilistic samples)

- 77 fourth year Computer Science students
- Selection by convenience

Selection of variables:

- **Independent (“factor”):**
 - **The diagram type**
 - Treatments: S (stereotyped) and N (non-stereotyped)
- **Dependent (“outcome”, “effect”):**
 - **Semantic comprehension:** the ability to comprehend the semantics of the models.

83



Planning

How to measure the dependent variable?

- SCEffec: # correct answers / # questions
- SCEffic: SCEffec / Time

Hypotheses formulation

Null Hypotheses (to be rejected):

- H1,0: stereotypes do not improve the subjects' SCEffec when attempting to comprehend a UML sequence diagram.
- H2,0: stereotypes do not improve the subjects' SCEffic when attempting to comprehend an UML sequence diagram

Alternative Hypotheses:

- H1,1: stereotypes improve the subjects' SCEffec when attempting to comprehend a UML sequence diagram.
- H2,1: stereotypes improve the subjects' SCEffic when attempting to comprehend an UML sequence diagram

84

alarcos MODELS '08 Planning Design

Relevant questions

- How many independent variables (factors)?
 - One factor → Simple design
 - More than one factor → Factorial design
- How many subjects per treatment ?
 - Between subjects: each subject receives only one treatment.
 - Within subjects: each subject receives all the treatments.

Independent variables → Diagram type + Domain
 Domain (A-extras car rental, B-book hotel room)

85

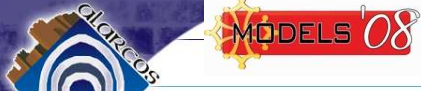
alarcos MODELS '08 Planning Design

RUN 1		DIAGRAM TYPE	
		S	N
DOMAIN	A	G1	G2
	B	G4	G3

RUN 2		DIAGRAM TYPE	
		S	N
DOMAIN	A	G3	G4
	B	G2	G1

A factorial within subjects design
 with interaction confounded

86



alazarcos

MODELS '08

Planning
Instrumentation


Experiments objects can be obtained through:

- Make yourself
- Make by students
- Find in internet
- Ask a company
- Find in a repository (ReMoDD)

Experiment objects. 4 diagrams corresponding to the four treatments prepared by the experimenters:

- DA-S: stereotyped diagram, domain A (*extras car rental*)
- DA-N: non-stereotyped diagram, domain A.
- DB-S: stereotyped diagram, domain B (*book hotel room*)
- DB-N: non-stereotyped diagram, domain B.

87



alazarcos



MODELS '08

Planning
Instrumentation

Tasks:

- Answer a test with 10 Yes/No questions concerning the semantics of the diagrams (SCEffec and SCEffic).
- Write down the start/end time.
- Manual tasks.

88



Planning
Validity evaluation



Internal validity: Is it the treatment that really causes the effect?

External validity: Can the results be generalised?

Construct validity: Do the experimental settings actually reflect the construct under study?

Conclusion validity: Are the results statistically valid?

89



Planning
Validity evaluation

Internal validity: Is it the treatment that really causes the effect?

- Subjects with similar experiences
- Fatigue effect was avoided (time duration)
- Learning effect was avoided (material distributed in different order)
- Random assignment of subjects into groups

External validity: Can the results be generalised?

- Selection of objects and subjects



Construct validity: Do the experimental settings actually reflect the construct under study?

- Select well known domains
- Sequence diagrams with similar size
- Select appropriate measures
- Take care with evaluation apprehension

Conclusion validity: Are the results statistically valid?

- Select the appropriate statistical tests according the design
- Check assumptions of the statistical tests

90



Operation
Preparation



Pilot study

Why? Check the experimental material, the experiment duration.

When? Two months before the experiment execution.

Who? 4 PhD students of the Alarcos research group at UCLM (Spain).

91

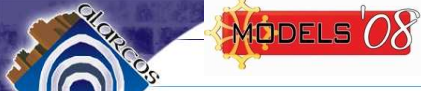


Operation
Preparation

Training session

- A training session of **three hours** took place the day before the experiment was carried out.
- The session included various tasks:
 - **To introduce** the students the **stereotypes** defined for UML sequence diagrams.
 - **To show** the students an **example**, similar to the material used in the experiment, in order to assure that the students had understood the experimental material and tasks. We solved the tasks in the example with the students.

92



Operation
Execution


The day after the training session, the experiment took place:

- 30 minutes → explanation on how to perform the experiment
- Randomly assignment of the subjects to the balanced groups
- Experiment execution in two runs.

The experiment was conducted in a classroom, where the students were supervised and no communication among them was allowed.

The groups were located in the same room.


93



Data analysis and interpretation
Descriptive statistics

- The subjects obtained better global results in the stereotyped diagrams.
- To some extent, the **stereotyped diagrams** were more **comprehensible**, less error prone and required less time to understand than non-stereotyped diagrams.

94



alarcos


MODELS 08

Data analysis and interpretation

Hypotheses testing

- Check normality of data
- Apply ANOVA test
- Results
 - Stereotypes**
 - There are no significant values.
 - The powers of the tests, i.e. the ability to detect an effect if there is one, are low.
 - We cannot reject the null hypotheses investigated in this work and accepting the alternative hypotheses could be risky and misleading.
 - Domain**
 - In most cases, the domain affects the comprehension of the diagrams.
 - The subjects performed worse with the diagrams of domain A (extras car rental). This could imply that the subjects found the extras car rental domain more difficult.

95



alarcos

MODELS 08

Data analysis and interpretation

Threats to validity

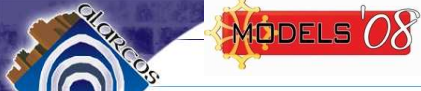
After experiment execution threats must be analyzed

Internal validity: We assumed a subjects' homogenous background, but this was not evaluated before the experiment was carried out.

Construct validity: The extras car rental diagram was found to be much more difficult than that of the book hotel room.

External validity: for generalising the results further replication is needed.


96



Conclusions

- Unfortunately the obtained results cannot be assumed to be final.
- The ***p-values*** observed in the hypotheses test are **non-significant** and thus the **null-hypotheses cannot be rejected**.
- Nonetheless, descriptive statistics show a **slight tendency in favor of the use of stereotypes**.
- **Further replication is needed.**

97



Contenido

- Introducción
- Tipos de estudios empíricos
- Experimentos
 - Conceptos generales
 - Proceso experimental
- Ejemplo
- **Réplicas**
- Agregación de resultados
- Recomendaciones
- Conclusiones
- Literatura relevante


98



Réplicas

- “El uso de experimentos precisos y repetible es el signo de una disciplina madura o de ingeniería”
- Es importante replicar los experimentos (corroborar/refutar resultados):
 - **Internas:** réplicas realizadas por los investigadores que realizaron el experimento original.
 - **Externas:** réplicas realizadas por experimentadores que no tuvieron nada que ver con el experimento original.
- Es importante construir “paquetes de laboratorio”: descripción del proceso experimental, decisiones tomadas, etc.
 - Favorecen las réplicas externas
 - Para lograr réplicas externas efectivas es necesario:
 - Comunicación efectiva entre el experimentador original y los experimentadores externos.
 - Transmitir el conocimiento tácito


99



Contenido


- Introducción
- Tipos de estudios empíricos
- Experimentos
 - Conceptos generales
 - Proceso experimental
- Ejemplo
- Réplicas
- **Agregación de resultados**
- Recomendaciones
- Conclusiones
- Literatura relevante

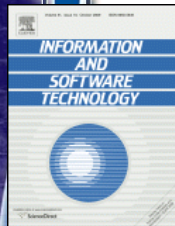
100

 **Agregación de resultados**

- **Objetivo** → Agregar la evidencia empírica de experimentos que persiguen la misma hipótesis para construir un cuerpo de conocimiento sobre el tema investigado.
- Técnicas para recopilar/agregar resultados:
 - **Revisones sitemáticas de la literatura**
 - **Meta-análisis**

101

 **Agregación de los resultados**
Una familia de experimentos

 Cruz-Lemus, J.A., Genero, M., Caivano, D., Abrahão, S. Insfrán, E., Carsí, J.A.. **Assessing the influence of stereotypes on the comprehension of UML sequence diagrams: A family of experiments. *Information and Software Technology* 53 (2011) 1391–1403.**

- **UBari**: 77 students (Computer Science – 4th year) (Italy)
- **UCLM**: 29 students (Computer Science – 3rd year) (Spain)
- **UCLM**: 36 students (Computer Science – 5th year) (Spain)



The set of stereotypes presented in this work seem to be helpful for a better comprehension of UML sequence diagrams, especially with not well-known domains in the context of novice analysts.


Contenido	
•	Introducción
•	Tipos de estudios empíricos
•	Experimentos <ul style="list-style-type: none">▪ Conceptos generales▪ Proceso experimental
•	Ejemplo
•	Réplicas
•	Agregación de resultados
•	Recomendaciones
•	Conclusiones
•	Literatura relevante

103

Recomendaciones	
•	Kitchenham et al. (2002) proponen una serie de recomendaciones para la investigación empírica en la Ingeniería de Software.
•	Jedlitschka et al. (2008) proponen recomendaciones sobre ¿Cómo reportar experimentos?



104


  **Recomendaciones**
Que hacer?

 **Planificación**

- Seleccionar objetos de tamaño y complejidad similar
- Asignar las mismas tareas a cada tratamiento
- Tratar de aliviar tanto como sea posible las amenazas a la validez
- Preparar pre-cuestionarios para recopilar la experiencia de los sujetos
- Preparar post-cuestionarios para recopilar el feedback de los sujetos
- Preparar el material en el lenguaje nativo de los sujetos


105

  **Recomendaciones**
Que hacer?

 **Operación**

- Realizar un estudio piloto para chequear el material
- Hacer una estimación precisa del tiempo de duración
- Estimular a los sujetos para que participen
- Capacitar a los sujetos en el tema bajo estudio
- Antes de la ejecución del experimento realizar un ejemplo con los sujetos
- Formar grupos balanceados por la experiencia
- Explicar claramente el procedimiento a seguir para la ejecución del experimento

106





Recomendaciones
Que hacer?




Análisis de datos e interpretación

- Tener suficiente conocimiento estadístico o contactar a un experto
- Chequear las suposiciones de los test estadísticos
- Contextualizar los resultados

107



Algunas Recomendaciones
Que hacer?






Presentación y difusión

- Seguir las directrices disponibles para reportar experimentos
- Reportar las amenazas que no se pudieron evitar
- Reportar la justificación de cada decisión tomada a lo largo del proceso experimental
- Preparar un “paquete de laboratorio” (material experimental, datos recolectados, etc.) y ponerlo disponible en la web.


Realizar réplicas

108

  **Algunas Recomendaciones**
Que NO hacer?


 **Definición**

- No trate de validar “demasiadas cosas” en un único experimento

 **Planificación**

- No desvele la hipótesis que se está investigando a los sujetos antes de la realización del experimento
- No desvele cual es el tratamiento que usted esta tratando de probar que es mejor

109

 **Contenidos**

- Introducción
- Tipos de estudios empíricos
- Experimentos
 - Conceptos generales
 - Proceso experimental
- Ejemplos
- Réplicas
- Agregación de resultados
- **Conclusiones**
- Literatura relevante

110

Conclusiones

- La **Ingeniería del Software basada en la Evidencia** promueve la toma de decisiones en la Ingeniería del Software basándonos en la evidencia.
- La **Ingeniería del Software Empírica** proporciona los mecanismos para recopilar la evidencia empírica → **estudios empíricos**
- Los **estudios empíricos** permiten evaluar/validar métodos/técnicas en la práctica:
 - El feedback sobre las técnicas/métodos lleva a mejorar
- Existen diferentes tipos de estudios empíricos (estrategias)
 - Dependiendo del objetivo y el contexto
 - Con diferentes Pros y Cons
- Es importante una **buena formación** en estudios empíricos, estadística, etc.
- Existe gran cantidad de literatura
- Los **estudios empíricos** en la Ingeniería del Software están en **aumento**

111

Conclusiones

Esencial !!!!

Colaboración entre universidad e industria

Universidad

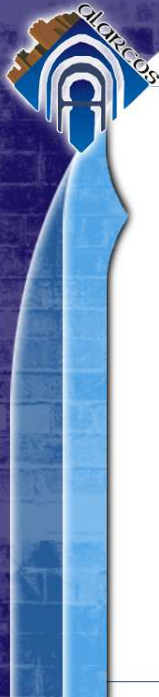




Industria



112

Contenido	
	<ul style="list-style-type: none">• Introducción• Tipos de estudios empíricos• Experimentos<ul style="list-style-type: none">▪ Conceptos generales▪ Proceso experimental• Ejemplo• Réplicas• Agregación de resultados• Recomendaciones• Conclusiones• Literatura relevante

113

Literatura relevante	
	Red y conferencias
	<ul style="list-style-type: none">▪ Red<ul style="list-style-type: none">▪ ISERN (International Software Engineering Research Network) http://www.iese.fhg.de/ISERN/▪ Conferencias<ul style="list-style-type: none">▪ EASE (Evaluation and Assessment in Software Engineering)▪ ESEM (Empirical Software Engineering and Measurement)▪ ICSE (International Conference on Software Engineering)▪ CISE 2013 (workshop en ICSE 2013)▪ Etc...

114

Literatura relevante
Revistas

Revistas donde se publican estudios empíricos



**EMPIRICAL
SOFTWARE
ENGINEERING**
AN INTERNATIONAL JOURNAL

Springer



IEEE TRANSACTIONS ON
**SOFTWARE
ENGINEERING**

Top Ranked Journal



The Journal of
Systems and Software



**INFORMATION
AND
SOFTWARE
TECHNOLOGY**



Journal of
Software:
Evolution and Process



Software:
Practice and Experience

115

Literatura relevante
Libros




Métodos de Investigación en Ingeniería del Software

*Marcela Genero Bocco, José Antonio Cruz-Lemus,
Mario Piattini Velthuis*

(marzo de 2014)





116






Literatura relevante

Libros

- Wohlin, C., Runeson, P., Höst, M., Ohlson, M., Regnell, B., Wesslén, A. (2012). **Experimentation in Software Engineering: An Introduction**. Springer.
- Juristo, N., Moreno, A. (2001). **Basics of Software Engineering Experimentation**. Kluwer Academic Publishers.
- Shull, Forrest; Singer, Janice; Sjøberg, Dag I.K. (Eds.) (2008). **Guide to Advanced Empirical Software Engineering**. Springer.

117



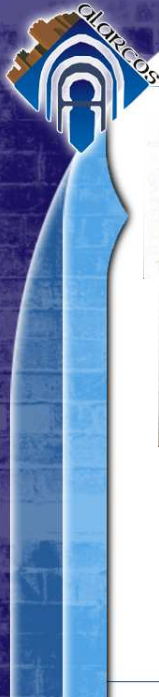


Lietaratura relevante

Libros


- Per Runeson, Martin Host, Austen Rainer, Bjorn Regnell. (2012). **Case Study Research in Software Engineering: Guidelines and Examples**. 1st ed., Wiley.
- Yin, R. K. (2008). **Case Study Research**, 4th ed., SAGE Publications.

118




Literatura relevante


Libros

- 

 - Fink, Arlene. **The Survey Handbook**, Sage Publications, (2003).

- 

 - Fink, Arlene. **How to report on surveys**, Sage Publications, 2nd ed. (2003).

- 

 - Kitchenham, B., Pfleeger, S. (2008). **Personal opinion survey**. Chapter 3, Guide to Advanced Empirical Software Engineering. Springer.

119




Literatura relevante

Directrices

- Kitchenham B., Pfleeger S., Pickard L., Jones P., Hoaglin D., El-Emam K. and Rosenberg J. (2002). **Preliminary Guidelines for Empirical Research in Software Engineering**. IEEE TSE, 28(8) 721-734.
- Jedlitschka, A., Ciolkowski, M. and Pfahl D. (2008). **Reporting experiments in Software Engineering**. Chapter 8, Guide to Advanced Empirical Software Engineering. Springer.
- Carver, J.. (2010). **Towards Reporting Guidelines for Experimental Replications: A Proposal**. 1st International Workshop on Replication in Empirical Software Engineering Research (RESER) [Held during ICSE 2010]. Cape Town, South Africa.
- Runeson, P., Höst, M. **Guidelines for conducting and reporting case study research in software engineering**. Empirical Software Engineering (2009) 14:131–164.
- Kitchenham, B. Pfleeger, S. **Principles of survey research**. ACM SIGSOFT Software Engineering Notes. Part 1 26(6) : 16-18 (2001), Part 2 27(1): 18-20 (2002), Part 3 27(2): 20-24 (2002), Part 4 27(3): 20-23 (2002), Part 5 27(5): 17-20 (2002), Part 6 28(2): 24-27 (2003)..

120




Literatura relevante

Ejemplos de experimentos

- Dzidek, W. J., Arisholm, E., Briand, L. C. (2008). **A realistic empirical evaluation of the costs and benefits of UML in software maintenance.** IEEE Transactions on Software Engineering, 34(3), 407–432.
- Arisholm, E., Briand, L. C., Hove, S. E., & Labiche, Y. (2006). **The impact of UML documentation on software maintenance: An experimental evaluation.** IEEE Transactions on Software Engineering, 32(6), 365–381.
- Jeffrey C. Carver, Nachi Nagappan, Alan Page. **The Impact of Educational Background on the Effectiveness of Requirements Inspections: An Empirical Study.** IEEE Trans. Software Eng. 34(6): 800-812 (2008)
- Erik Arisholm, Hans Gallis, Tore Dybå, Dag I. K. Sjøberg. **Evaluating Pair Programming with Respect to System Complexity and Programmer Expertise.** IEEE Trans. Software Eng. 33(2): 65-86 (2007)
- Lionel C. Briand, Yvan Labiche, Massimiliano Di Penta, Han (Daphne) Yan-Bondoc. **An Experimental Investigation of Formality in UML-Based Development.** IEEE Trans. Software Eng. 31(10): 833-849 (2005)

121

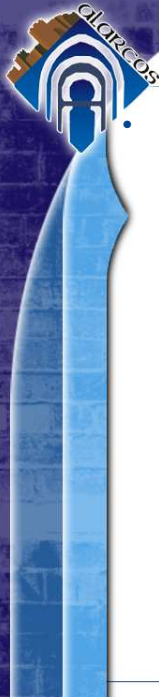


Literatura relevante

Ejemplos de experimentos

- Lionel C. Briand, Christian Bunse, John W. Daly. **A Controlled Experiment for Evaluating Quality Guidelines on the Maintainability of Object-Oriented Designs.** IEEE Trans. Software Eng. 27(6): 513-530 (2001)
- Iris Reinhartz-Berger, Dov Dori. **OPM vs. UML - Experimenting with Comprehension and Construction of Web Application Models.** Empirical Software Engineering 10(1): 57-80 (2005)
- Silvia Teresita Acuña, Marta Gómez, Natalia Juristo Juzgado. **How do personality, team processes and task characteristics relate to job satisfaction and software quality?** Information & Software Technology 51(3): 627-639 (2009)
- Scaniello, G., Gravino, C., Genero, M., Cruz-Lemus, J.A., Tortora, G. **The Impact of UML Models Produced in the Requirements Analysis Process on Source Code Comprehensibility and Modifiability: Results from a Family of Four Experiments.** ACM TOSEM (2013).

122




Literatura relevante

Ejemplos de experimentos

- **Mas experimentos en:**
 - Sjoberg, D., Hannay, J., Hansen, O., Kampenes., V., Karahasanovic., A., Liborg, N., Rekdal, A. (2005). **A Survey of Controlled Experiments in Software Engineering**. IEEE TSE 31(9), 2005.
 - 103 artículos que describían 113 experimentos controlados, 1993 -2002.

123




Literatura relevante

Ejemplos de estudios de casos

- Damian, D., Chisan, J., Pal, Y. **Requirements Engineering and Downstream Software Development: Findings from a Case Study**. Empirical Software Engineering, 10, 255–283, 2005
- Karlstrom, D., Runeson, P. **Integrating agile software development into stage-gate managed product development**. Empirical Software Engineering. (2006) 11: 203–225.
- Z. Li, N. H. Madhavji, S. S. Murtaza, M. Gittens, A. V. Miransky, D. Godwin, and E. Cialini. **Characteristics of multiple-component defects and architectural hotspots: a large system case study**. Empirical Software Engineering, 16(5):667–702, 2011.
- Segal, J. **When Software Engineers Met Research Scientists: A Case Study**. Empirical Software Engineering, 10, 517–536, 2005.
- Morisio, M. Romano, D., Stamelos, I. **Quality, Productivity, and Learning in Framework-Based Development: An Exploratory Case Study**. IEEE TSE, 28(9), (2002), 876-888.
- A. Rainer, **The longitudinal, chronological case study research strategy: A definition and an example from IBM Hursley Park**, Information and Software Technology, Volume 53, Issue 7, July 2011, Pages 730-746

124



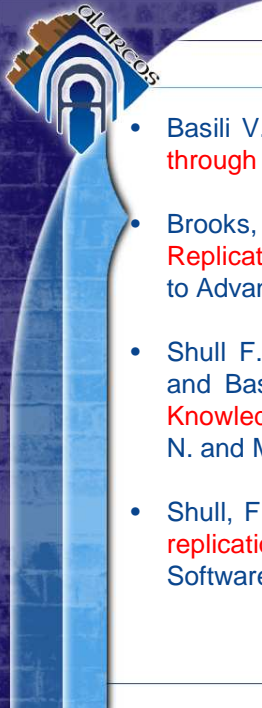
Literatura relevante

Ejemplos de encuestas

- Grossman, M., Aronson, J. E., & McCarthy, R. V. (2005). **Does UML make the grade? Insights from the software development community.** Information and Software Technology 47(6) 383-397. Newton, MA, USA.
- M. Jørgensen, K. Moløkken-Østvold, **How large are software cost overruns? A review of the 1994 CHAOS report,** Information and Software Technology, 48(4):297-301, 2006,

Lange, C. F. J., Chaudron, M. R. V., Muskens, J. (2006). **In practice: UML software architecture and design description.** IEEE Software 23(2) 40-46.

125



Literatura relevante

Réplicas

- Basili V., Shull F. and Lanubile F. (1999). **Building knowledge through families of experiments,** IEEE TSE, 25(4) 435-437.
- Brooks, A., Roper, M., Wood, M., Daly, J., Miller, J. (2008). **Replication's role in Software Engineering.** Chapter 14, Guide to Advanced Empirical Software Engineering. Springer.
- Shull F., Carver J., Travassos G., Maldonado J., Conradi R. and Basili V. (2003). **Replicated Studies: Building a Body of Knowledge about Software Reading Techniques.** LNCS (Juristo N. and Moreno A. Eds.), World Scientific, Singapore, 39-84.
- Shull, F., Carver, J., Vegas, S., Juristo, N. (2008). **The role of replications in Empirical Software Engineering.** Empirical Software Engineering 13(2), 211-218.

126



XVII Escuela Internacional de Informática

**Ingeniería del software basada en la
evidencia (IBSE)
(PARTE I)**

Marcela Genero Bocco

