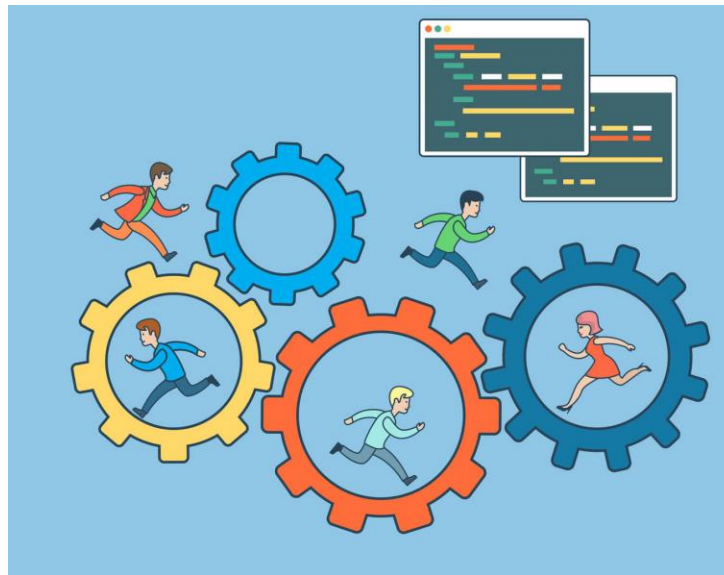


Orientación a Objetos 2

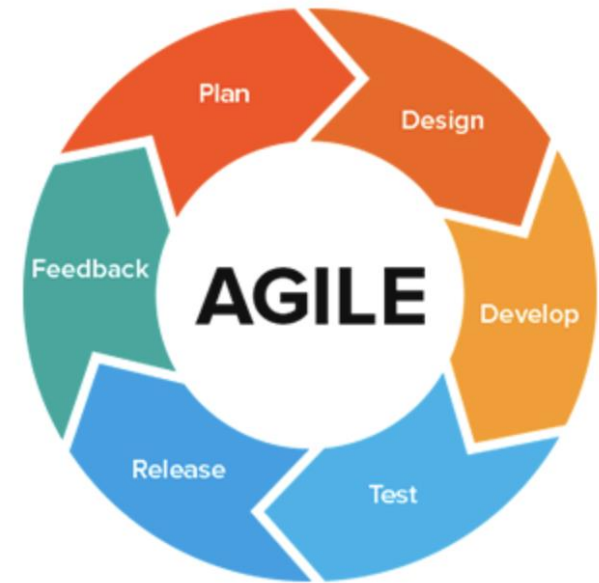
Integración de contenidos



Contexto: métodos y prácticas de desarrollo de hoy

- **Métodos ágiles**

- se desarrolla en iteraciones
- cada iteración extiende código existente con nueva funcionalidad
- poca documentación
- testing automático e integración continua



- Arquitecturas complejas (arquitecturas modulares, capas con distintos concerns: interfaz de usuario, persistencia, seguridad, ...), arquitecturas orientadas a servicios (conexión con código ajeno, desarrollo de APIs, ...)
- El cambio es continuo, el crecimiento es continuo, la complejidad es cada vez mayor

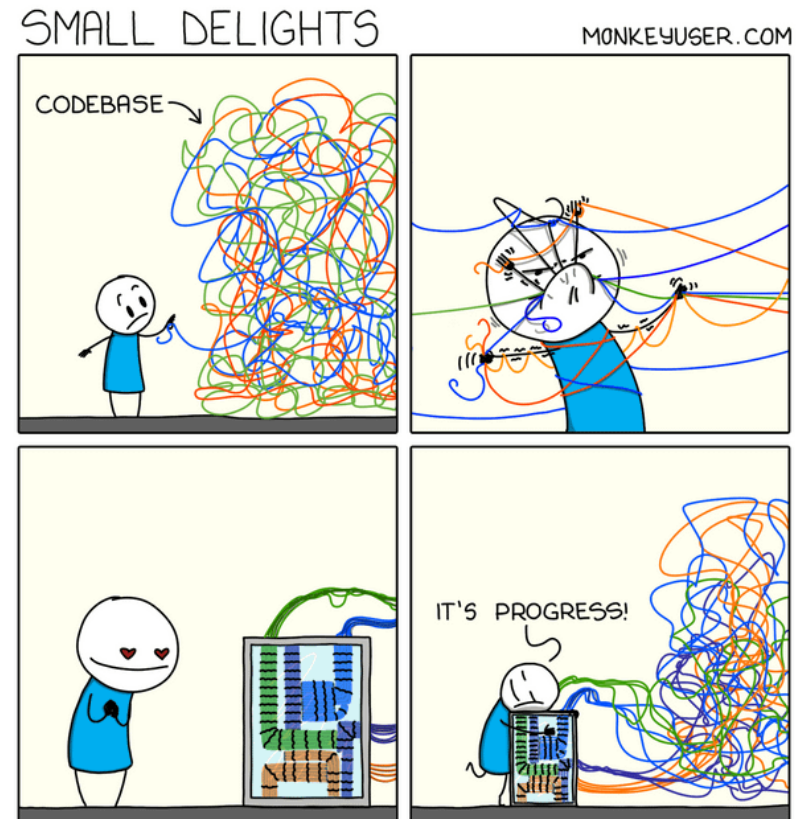
- Estar mejor preparados para el cambio, porque:
 - El ritmo del cambio en la tecnología y en la demanda de inmediatez de los usuarios crece **exponencialmente**
- Desarrollar la capacidad de cambiar el software de manera **rápida y sustentable**
 - consciente, evitando el desperdicio de recursos, reconociendo oportunidades, responsable, con calidad, buscando la mejora continua



Qué aprendimos en Objetos 2

- Reconocer las fuerzas que llevan al deterioro de la arquitectura (**SMELLS**) y aprender a reconocer oportunidades para mejorarla (**REFACTORINGS**)
- Con el objetivo de lograr código legible y extensible (**MEJORAR CALIDAD INTERNA**)
- En cada iteración (**RESOLVIENDO LA DEUDA TECNICA**)

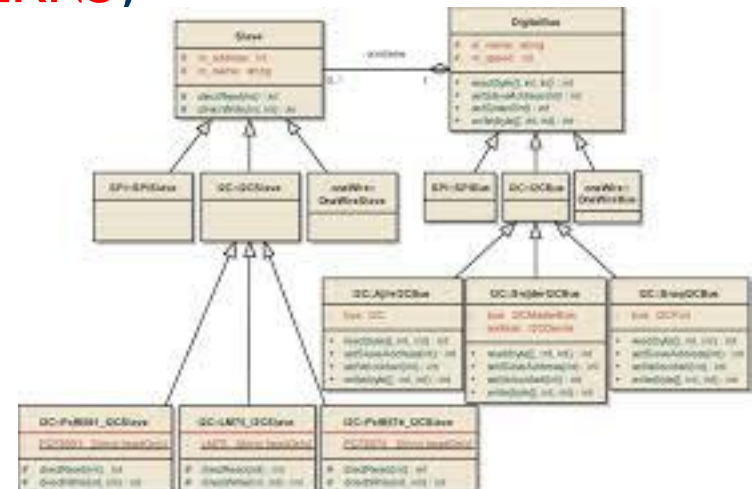
REFACTORING.



Qué aprendimos en Objetos 2

- Construir diseños reusables, más sensibles a la evolución (**CATALOGO DE PATRONES**)
- Reconocer las consecuencias positivas y negativas de un diseño (fundamentar y criticar diseños)
- Hacer evolucionar diseños, aplicando mejoras paso a paso (**REFACTORING TO PATTERNS**)

PATRONES DE DISEÑO



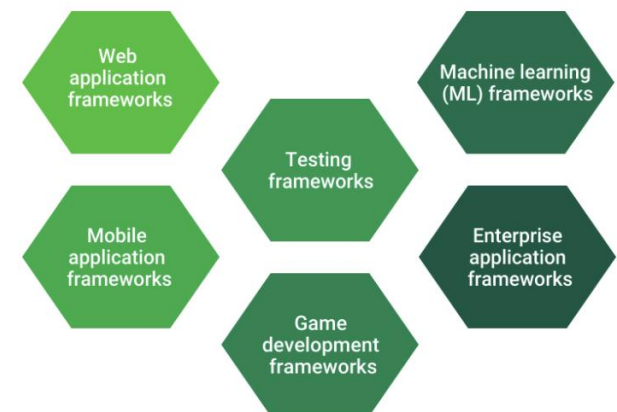
¿Qué cosa es importante recordar de patrones?

- Propósito
- Estructura:
 - clases que componen el patrón (roles),
 - cómo se relacionan (jerarquías, clases abstractas/interfaces, métodos abstractos - protocolo de interfaces, conocimiento/composición)
- Variantes de implementación
- Consecuencias positivas y negativas
- Relación con otros patrones (RECORDEMOS LAS RELACIONES QUE FUIMOS VIENDO ENTRE LOS PATRONES)

Qué aprendimos en Objetos 2

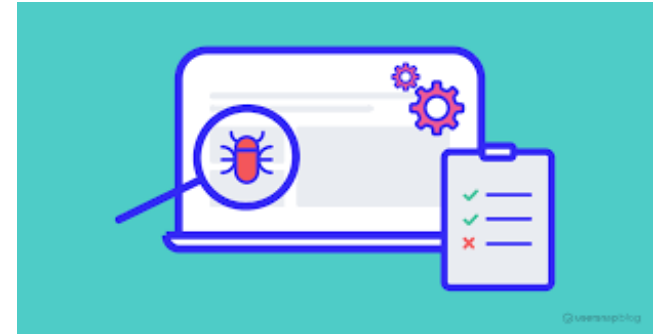
- Instanciar arquitecturas reusables para una familia de aplicaciones, reconociendo puntos de variabilidad y extensión de estas arquitecturas (**FRAMEWORKS, HOT SPOTS Y FROZEN SPOTS**)
- Entender las implicancias de utilizar estas arquitecturas (**CAJA BLANCA Y CAJA NEGRA**)
- Diseñar y programar estas arquitecturas (**USO DE PATRONES EN FRAMEWORKS**)

FRAMEWORKS.

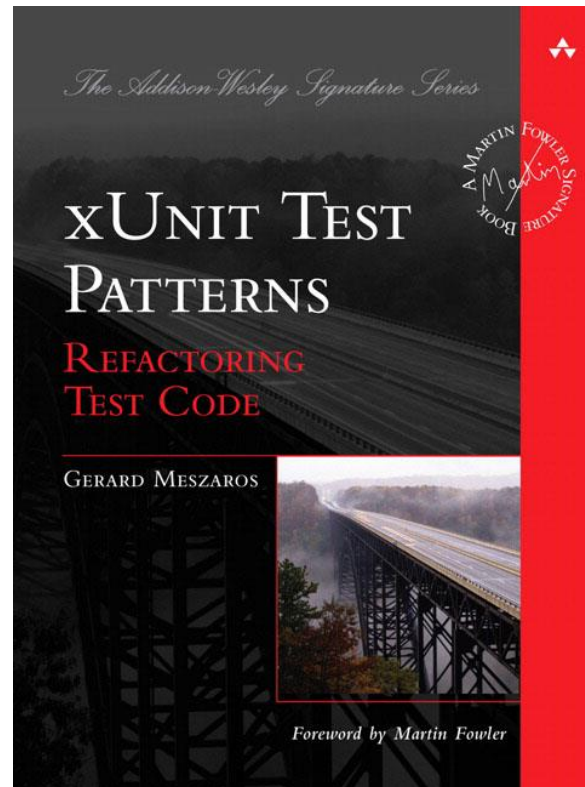


Buscando calidad en lo que hacemos

- Importancia del testing
(**ANTES Y DESPUÉS DEL REFACTORING**)
- Tipos de Tests
- Estrategias de testing con test doubles
- Test Driven Development (**TESTING+REFACTORING**)
- JUNIT es un FRAMEWORK
- XUNIT PATTERNS: Estrategias para escribir tests de unidad documentadas como patrones



XUnit Test Patterns



Cómo organizamos los test methods en testcase classes?

- **Testcase class per class**
 - Poner todos los test methods de un SUT en una única testcase class
 - **How to use it?** Creamos una clase separada por cada clase que queremos testear
 - **When to use it?** Es un buen punto de partida cuando no hay muchos test methods
- **Testcase class per feature (por cada método)**
 - Agrupamos test methods basados en cada feature de la clase
- **Testcase class per fixture**
 - Agrupamos los test methods que comparten un mismo fixture

Fixture Setup Patterns

- In-line Setup
 - Cada test method crea su propio fixture nuevo
- Delegated Setup
 - Cada test method crea su propio fixture llamando a Métodos de Creación
 - `createAnonymousFlight();`
 - `createAnonymousCancelledFlight();`
- Implicit Setup
 - Se construye un test fixture común a varios tests en un método `setUp()`

Objetos 2 - Aprobación de la materia

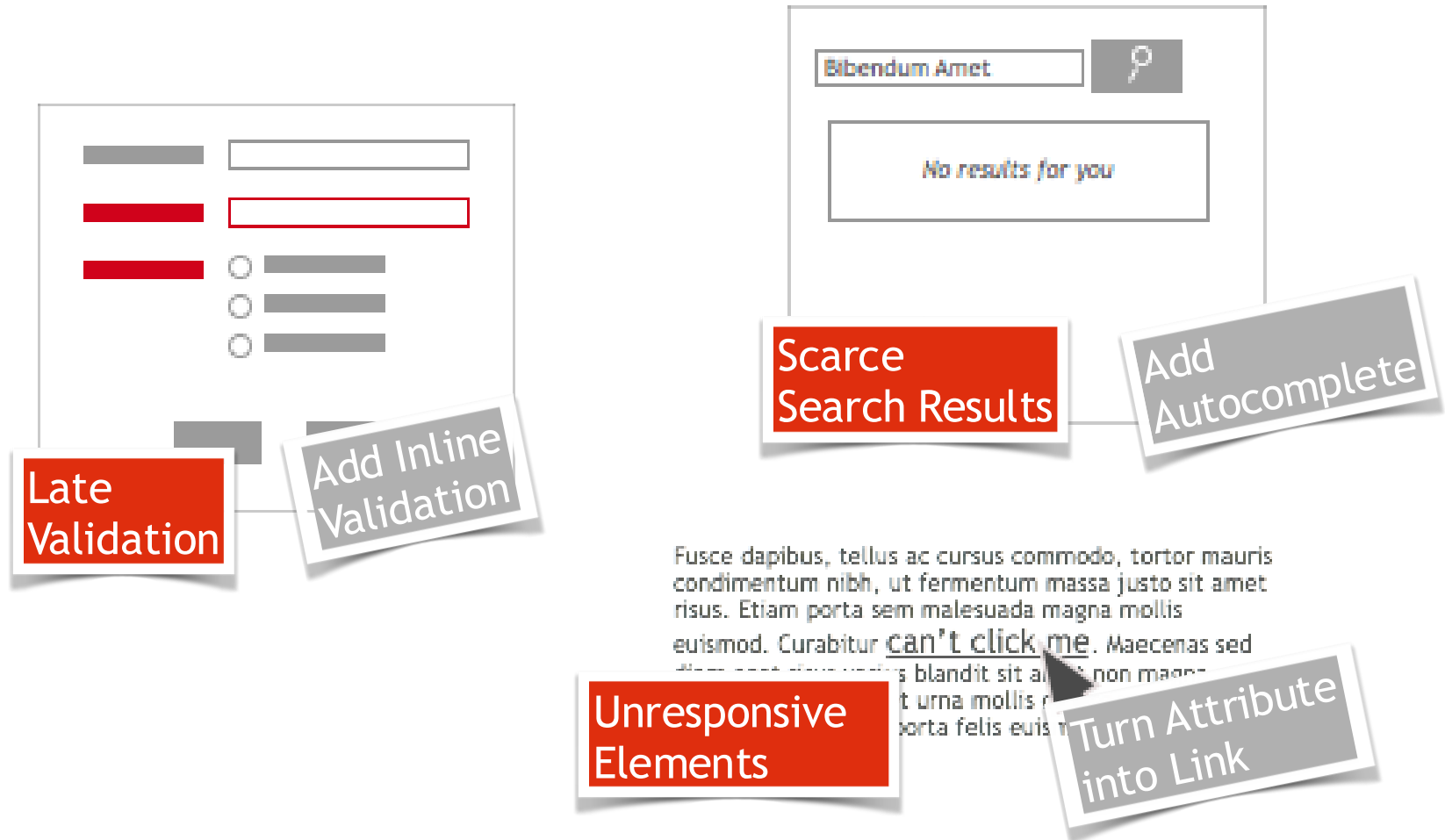
- Cursada:
 - se aprueba con un examen parcial, que tiene 2 recuperatorios. El parcial se aprueba por temas.
- Fechas de parcial:
 - 7 de junio
 - 28 de junio
 - 12 de julio
- Final de la materia:
 - Examen final
 - Promoción: se rinde el 12 de julio. Única fecha

Objetos 2 - Promoción

- Para promocionar la materia se requiere:
 - aprobar la cursada en 1ra o 2da fecha de parcial;
 - responder correctamente las preguntas extra que se agregan al parcial para acceder a la promoción;
 - aprobar un examen integrador de promoción que se rinde por única vez el 12/7/25 con calificación 6 o más.

- Proyecto de investigación: UX-Driven Development: Monitorización continua de la experiencia del usuario durante el desarrollo y mantenimiento ágil de aplicaciones interactivas
- Información del proyecto: [link](#)
- UX Refactoring
- UX-Debt
- Framework de detección y corrección de UX Smells: Kobold
- UX Refactoring Tool: UX-Painter
- UX Evaluation Tool: UX-Analyzer
- Dark patterns

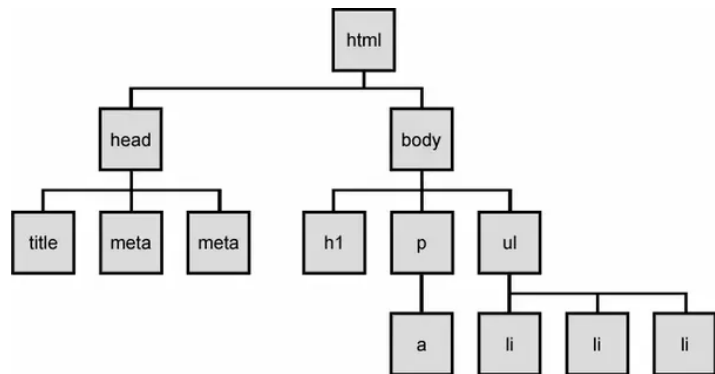
UX Smells and UX Refactorings



•Client-Side Web Refactorings (CSWR):

- Refactorings que solucionan malos olores de UX aplicando cambios del lado del cliente, sobre el DOM de las páginas web

*(“Personalized Web Accessibility Using Client-Side Refactoring”,
Garrido et al. IEEE Internet Comp 17(4), 2013)*



~ = code refactoring: tree rewrite !

The image shows a web browser window displaying the Amazon shipping address selection page. The browser's address bar shows the URL: `amazon.com/gp/buy/addressselect/handlers/display.html?hasWorkingJavascript=1`. The page header includes the Amazon logo and navigation links: **SIGN IN**, **SHIPPING & PAYMENT**, **GIFT OPTIONS**, and **PLACE ORDER**. The main heading is "Select a shipping address", followed by the instruction "Enter a new shipping address." and a note: "When finished, click the 'Continue' button."

The form contains the following fields:

- Full name:** A text input field.
- Address line 1:** A text input field with placeholder text: "Street address, P.O. box, company name, c/o".
- Address line 2:** A text input field with placeholder text: "Apartment, suite, unit, building, floor, etc."
- City:** A text input field.
- State/Province/Region:** A text input field.
- ZIP:** A text input field.
- Country/Region:** A dropdown menu currently showing "United States".
- Phone number:** A text input field with a link "(Learn more)" next to the label.

On the right side of the browser window, there is a "Versions" panel. It displays "Original" with a lock icon, a cursor icon, and a copy icon. Below this is a yellow button labeled "New version" with a plus icon.

UX-Painter: check-out de Amazon refactorizado

The image shows a screenshot of the Amazon shipping address selection page. The form is titled "Select a shipping address" and includes a sub-header "Enter a new shipping address." with instructions: "When finished, click the 'Continue' button." The form fields are: Full name, Address line 1, Address line 2, City, State/Province/Region, Postal Code, and Country/Region. A dropdown menu is open for the Country/Region field, showing options like "Un", "Burundi", "Hungary", "Romania", and "Tunisia".

On the right side, there is a panel titled "Custom form" with a section "Refactorings Applied". The refactorings listed are: Add Inline Form Validation, Turn Select into Autocomplete, Rename Element, Resize Input, and Turn Attribute into Link. Below this list are buttons for "Add Refactoring +", "Back", and "Save".

Colored arrows point from the refactorings to the form elements: a red arrow from "Turn Attribute into Link" to the "SHIPPING & PAYMENT" link; a grey arrow from "Add Inline Form Validation" to the "Full name" input; an orange arrow from "Turn Select into Autocomplete" to the "Country/Region" dropdown; a blue arrow from "Rename Element" to the "Country/Region" dropdown; and a green arrow from "Resize Input" to the "Postal Code" input.

UX_Analyzer

Evaluations > Air somewhere

Air somewhere

New Version

Deceptive

Interaction Effort

1.4

Users: 86



Normal

Interaction Effort

1.2

Users: 63



Evaluations > Air somewhere > Don't select seat link

Don't select seat link

Stats User Sessions Widgets Setup

Label	Type	Interaction Effort	Weight
<input type="checkbox"/> Self Check In	Anchor	1.2	<div><div></div></div>
<input type="checkbox"/> Find Reservation	Anchor	1.3	<div><div></div></div>
<input type="checkbox"/> I don't want to select a seat	Anchor	1.5	<div><div></div></div>
<input type="checkbox"/> Continue	Anchor	1.3	<div><div></div></div>
<input type="checkbox"/> Finish	Anchor	1	<div><div></div></div>