**Metodologías Agiles – Ciclo unificado – Fundamentos Agiles**

**Metáforas**

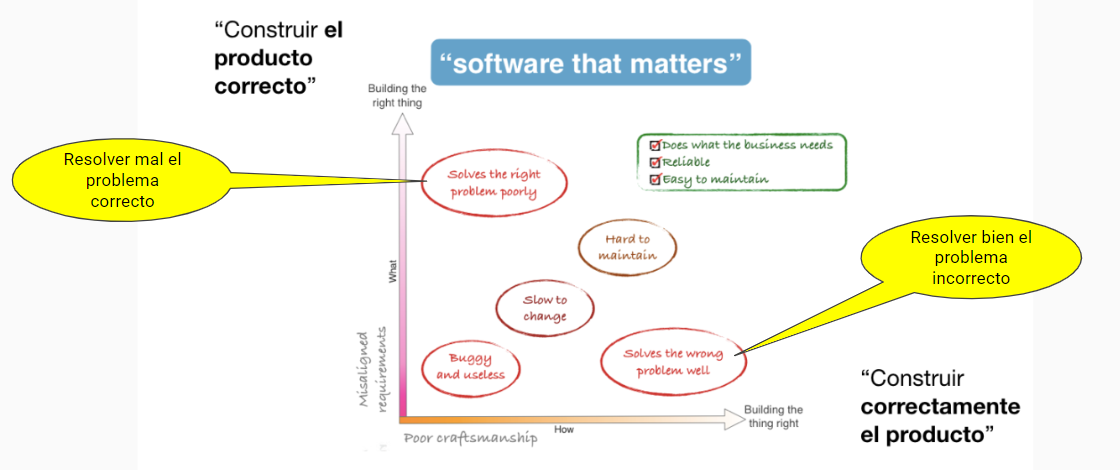
Son muy útiles para comunicar y establecer ideas complejas y actitudes en una organización.

**Mejorar la vida – Sacar el dolor**

**El factor humano – Comunicación y colaboración**

* En el desarrollo de software es fundamental el “**factor humano”.**
* La ingeniería es el arte de resolver problemas y mejorar la vida de las personas.
* Para construir un sistema es necesario entender como las personas van a utilizarlo.
* Una gran parte del trabajo de desarrollo de software es comunicación y colaboración.
* Gran parte de las herramientas de las met. agiles no son sino formas de gestionar la comunicación y la colaboración.

**Las dos caras del desarrollo de software**



**Puesta en funcionamiento**

**Software como producto a distribuir y desplegar**

* Modelos de distribución y despliegue
* Aplicación web en internet
* App móvil
* Aplicación de escritorio
* En cualquier modelo es necesario que la empresa de desarrollo **mantenga todas las versiones del software entregadas** y usadas por los usuarios. Es **imprescindible un sistema de control de versiones.**

**La importancia de lo funcional del soft**

**Software como producto a usar**

* El objetivo del software es ser usado y **proporcionar valor** en ese uso.
* La flexibilidad del software hace complicado diseñar el producto final.
* Muchas veces hay que **descubrir** el producto final.ee
* Ciclo de mejora continua: IDEA 🡪 PRODUCTO 🡪 EVALUACION

**Calidad en el software**

* Nuestra responsabilidad es desarrollar software de calidad, mejorando los procesos de desarrollo y el resultado final.
* El producto no funciona cuando:
* La aplicación solo cumple parcialmente lo que necesita el usuario.
* La aplicación no entiende el contexto del usuario.
* El software está mal desarrollado cuando:
* Es lento, poco eficiente, poco robusto.
* El código es poco modificable, poco entendible, con **deuda técnica.**

**Las 3 variables de un desarrollo software**

* El **alcance:** cuantos requisitos o tareas hay que realizar.
* El **tiempo:** cuánto durará el desarrollo.
* El **coste:** cuánto dinero se dedicará al proyecto.

Se denomina el **“triángulo de hierro”** y muestra la relación entre las 3 variables. No podemos mejorar una de ellas sin tocar alguna de las otras o afectar la calidad.

**Las 4 dimensiones del desarrollo software**

El resultado final depende de cuatro dimensiones principales:

* **Personas**
* **Proceso**
* **Producto**
* **Tecnología**

**Metodologías de desarrollo**

**¿Qué es una metodología de desarrollo?**

* Define pasos a seguir, modelo del ciclo de vida del software.
* Roles y responsabilidades de los distintos actores que intervienen en el proceso.

Una metodología es más que un conjunto de reglas, también se compone por valores, principios, venas prácticas y herramientas.

**Modelo de desarrollo de cascada (Waterfall)**

* Fuerte énfasis en la planificación detallada y predecible
* El análisis y diseño se hace en gran parte antes de la implementación, prueba y puesta en producción.
* Cada fase debe producir un **artefacto** que es analizado, corregido y que se da como entrega a la próxima fase.
* Pruebas
* Se realizan al final del desarrollo.
* Cada fase tiene sus propias validaciones.
* Cada artefacto se valida antes de la siguiente fase.
* Las pruebas de código se plantean como una verificación.
* No se contempla el cambio, evolución ni automatización de pruebas.

**Brooks: El cuello de botella del desarrollo de software**

Hay cuatro propiedades de los elementos esenciales de un proyecto de software:

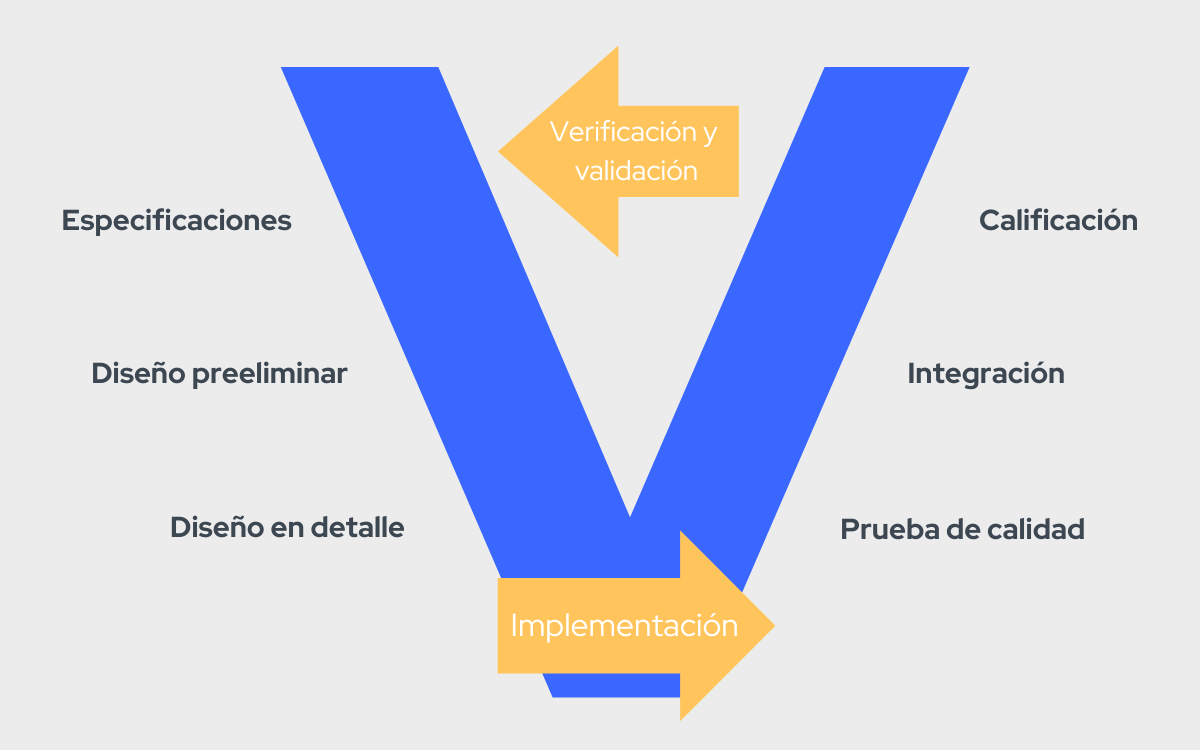
* Complejidad
* Conformidad
* Cambiabilidad
* Invisibilidad

Algunas ideas de como seria posible optimizar el desarrollo de los elementos esenciales del software:

* **Comparar frente a construir:** utilizar software ya existente.
* **Prototipado rápido y refinamiento de los requisitos.**
* **Desarrollo incremental – crecer no construir el software.**

**Modelo V**

* Cada fase del ciclo define un conjunto de pruebas que hay que realizar sobre el producto final.
* El eje horizontal representa el tiempo o el grado de terminación.
* El eje vertical representa el nivel de abstracción.
* Evolución a mejor del modelo cascada:
* Énfasis en pruebas.
* Distintos tipos de pruebas.
* Relación entre fases iniciales.

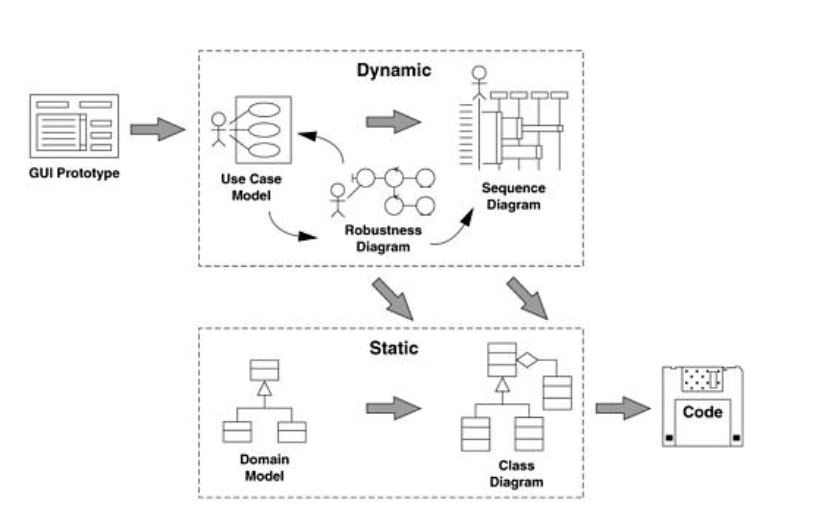


**Definición del proceso ICONIX**

* Metodología de desarrollo de software.
* Solución de compromiso entre RUP y XP.
* Basado en UML.
* Guiado por casos de uso.
* Usa sólo 4 diagramas UML.

**Etapas**

* **Análisis de requerimientos.**
* Identificar Casos de Uso.
* Producir un Modelo de Dominio.
* Hacer prototipos UI.
* **Análisis y Diseño preliminar.**
* Describir textualmente los Casos de Uso.
* Efectuar un Análisis de Robustez.
* **Diseño detallado**
* Construir un Diagrama de Clases a partir del MD.
* Construir Diagramas de Secuencia a partir de los CU.
* **Implementación**
* Escribir Casos de Prueba.
* Codificación a partir de los DCD y DSD.



**ICONIX logra:**

* Uso racionalizado de UML.
* Alto grado de trazabilidad.
* Iterativo e incremental.

**El Proceso Unificado**

**¿Por qué implementarlo?**

* Inefectividad, ineficacia y/o ineficiencia.
* Porque tiene malas variables en su economía.
* Porque tiene mala calidad de software.
* Porque tiene mala mantenibilidad.
* Porque tiene complejidad arbitraria.

**¿Qué es?**

* Proceso de desarrollo de software basado en componentes, construido por componentes interconectados y bien definidos.
* Dirigido por Casos de Uso.
* Centrado en la Arquitectura.
* Iterativo e incremental.

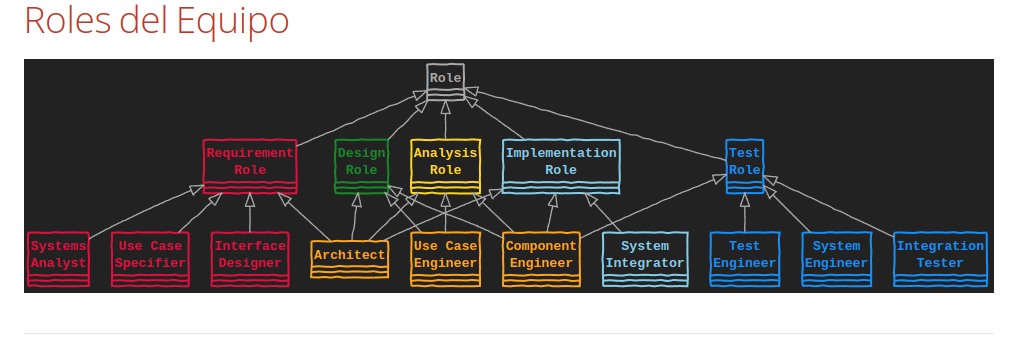
**¿Para qué implementarlo?**

* Efectividad, eficacia y eficiencia.

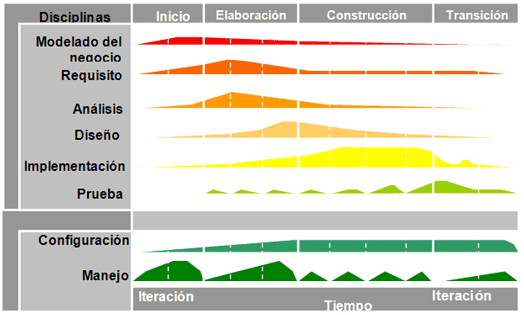
**Beneficios**

* Construir el sistema en el tiempo de forma incremental.
* Manejar mejor los requisitos cambiantes.
* Disponer de una estructura para guiar el desarrollo.
* Gestión temprana del riesgo.

**Roles**



**Fases**



**Fundamentos Ágiles**

**Modelo espiral**

* Se aprovecha de que el software es fácilmente modificable.
* Sistema mínimo funcionando desde el principio.
* Cada iteración añade un conjunto de funcionalidades y si es necesario refina lo existente.
* Al final de cada iteración se obtiene un prototipo.

**La propuesta ágil: ciclos cortos**

* Cada ciclo incorpora todas las fases del ciclo de vida.
* Retroalimentación temprana.
* Cada iteración tiene un nuevo producto al que se le han añadido características nuevas.
* El producto evoluciona y adopta los cambios sugeridos por el cliente.

**¿Qué es ágil?**

**Desarrollo incremental**

* En todas se adopta un desarrollo de iteraciones cortas.
* Al final de cada una debe obtenerse un incremento.
* Debe permitir una integración rápida de las nuevas características.

**Principios Ágiles**

* Satisfacer al cliente mediante la entrega rápida de software útil.
* Se entrega frecuentemente software que funciona.
* La simplicidad del código es esencial.
* Las iteraciones cortas entregan valor al cliente y permiten retroalimentación para el equipo.
* Kaizen, mejora continua.

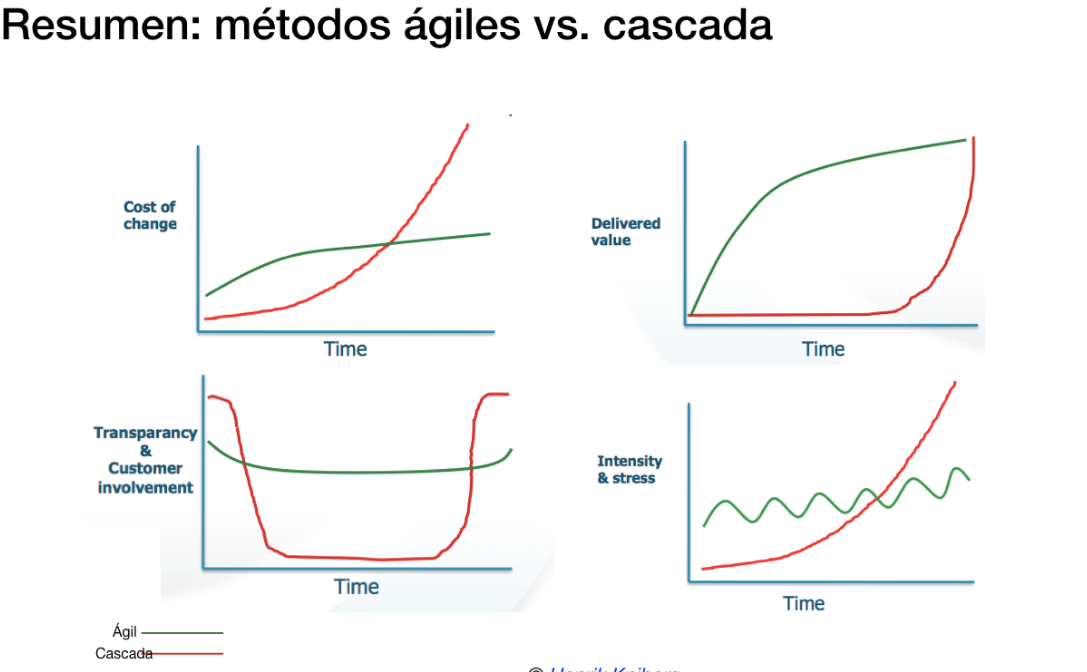
**Principios del Manifiesto Ágil**

1. **Satisfacer al cliente mediante entrega temprana y continua de software con valor.**
2. **Aceptar que los requisitos cambien, incluso en etapas tardías.**
3. **Entregar software funcional frecuentemente.**
4. **Los responsables de negocio y desarrolladores trabajan en conjunto.**
5. **Los proyectos se desarrollan en torno a individuos motivados.**
6. **El método más efectivo de comunicación es el cara a cara.**
7. **El software funcionando es la medida principal de progreso.**
8. **Promueven el desarrollo sostenible.**
9. **La atención continua a la excelencia técnica y al buen diseño mejora la agilidad.**
10. **La simplicidad es esencial.**
11. **Las mejores arquitecturas, requisitos y diseños emergen de equipos auto-organizados.**
12. **A intervalos regulares el equipo reflexiona sobre cómo ser más efectivo.**

**Producto mínimo viable.**

* MVP
* Idea para construir startups aplicable al desarrollo de software.
* Hacer primero lo que más incertidumbre genera.

**Agile vs Cascada**



**Valores, principios y practicas**

* Las practicas tiene sentido en el marco de un conjunto de valores y principios.
* Las met. Agiles responden a principios y valores.
* Si se intenta copiar las practicas sin aplicar los principios subyacentes se caerá en la repetición vacía.
* Deben ser compartidos por toda la organización.
* Son generales y se pueden aplicar a múltiples situaciones.

**Los 4 valores del manifiesto ágil**

1. **Individuos e interacciones sobre proceso y herramientas.**
2. **Software que funciona sobre documentación exhaustiva.**
3. **Colaboración con el cliente frente a negociación exhaustiva.**
4. **Responder al cambio frente a seguir un plan.**