**UNIVERSIDAD TECNOLÓGICA DE PANAMÁ**

**FACULTAD DE INGENIERÍA DE SISTEMAS COMPUTACIONALES**

**LICENCIATURA EN INGENIERÍA DE SISTEMAS DE INFORMACIÓN**

**INGENIERÍA DE SOFTWARE**

**TEMA: MODELO DE DESARROLLO Y CICLOS DE VIDA, SISTEMAS Y SUS SOFTWARE**

**LABORATORIO #1**

**Profesora: Ana Gloria Cordero de Hernández MSc.**

**Alumno:**

**Cutire, Fernando 8-972-906**

**Grupo: 1IF121**

**II semestre 2020**

1. Describa a través de una presentación escrita, el nombre del paradigma o metodología, autores, breve historia, año de surgimiento y una imagen representativa de su ciclo de vida, para uno de las siguientes clasificaciones. (No debe presentar los mostrados en la clase) 10 p.

Paradigma de desarrollo tradicional.

Paradigma de desarrollo orientado a objetos.

Paradigmas de desarrollo ágil de aplicaciones.

Otros paradigmas de desarrollo.

En esta sección se resolverá según el paradigma de desarrollo ágil de aplicaciones usando la metodología **desarrollo de software *lean.***



Ilustración : Ciclo del LSE

El término de desarrollo de software *lean* se utilizó por primera vez como título de una conferencia organizada por la iniciativa ESPRIT de la Unión Europea.[1](https://es.wikipedia.org/wiki/Lean_software_development#cite_note-1)​

Sin mantener relación alguna, Robert “Bob” Charette, planteó un año después el concepto de desarrollo de software *lean* como parte de su trabajo de investigación sobre mejores formas para administrar los riesgos en proyectos de software.

Más tarde, en mayo de 2003, Mary Poppendieck y Tom Poppendieck presentan su libro *"Desarrollo de software Lean*".[2](https://es.wikipedia.org/wiki/Lean_software_development#cite_note-2)​ El libro presenta los tradicionales principios Lean en forma modificada, así como un conjunto de 22 instrumentos y herramientas y las comparaciones con otras prácticas ágiles. La participación de Mary y Tom en la comunidad del desarrollo ágil de software, incluyendo charlas en varias conferencias, ha dado lugar a dichos conceptos, que son más ampliamente aceptados en la comunidad de desarrollo ágil. Ejemplos de ello sería la utilización del término "*lean-agile*" por empresas de consultoría como NetObjectives Pace y CC, así como la inclusión de algunos de estos conceptos.

2. Exponer sistemas y software identificando: Definición, Aplicaciones o software y Ejemplos, para la siguiente clasificación de sistemas (coloque ilustraciones):

En esta sección se desarrollará usando la metodología presentada anteriormente.

**Definición**

Tenemos muchas otras metodologías en Agile como Scrum y Extreme Programming, entonces ¿por qué Lean? ¿Qué lo diferencia de otros métodos?

Aquí está la respuesta: la metodología de desarrollo ajustado reduce los costos, los esfuerzos y el desperdicio. El desarrollo de software ajustado es una metodología de iteración desarrollada inicialmente para la industria manufacturera para optimizar la producción y evitar el desperdicio. Muchos de los principios y prácticas en el desarrollo de software ajustado provienen del movimiento empresarial ajustado y fueron utilizados inicialmente por grandes empresas como Toyota.

Mary y Tom Poppendieck llevaron Lean al mundo del desarrollo de software. Convirtieron todos los valores, prácticas y principios de acuerdo con la industria del software, los documentaron todos en un libro y los pusieron en práctica.

**Aplicaciones o software**

Lean Manufacturing y Toyota Production Systems (TPS) fueron los primeros en desarrollar Lean Software Development. En TPS, dividieron los desechos en siete secciones principales.

Mary y Tom Poppendeick, en su libro "Lean Software Development - An Agile Toolkit", interpretaron estos siete desperdicios de fabricación en siete desperdicios de desarrollo de software. Estos siete desperdicios son sin valor agregado y actúan como oponentes. Nos hacen perder nuestros esfuerzos, tiempo y costos.

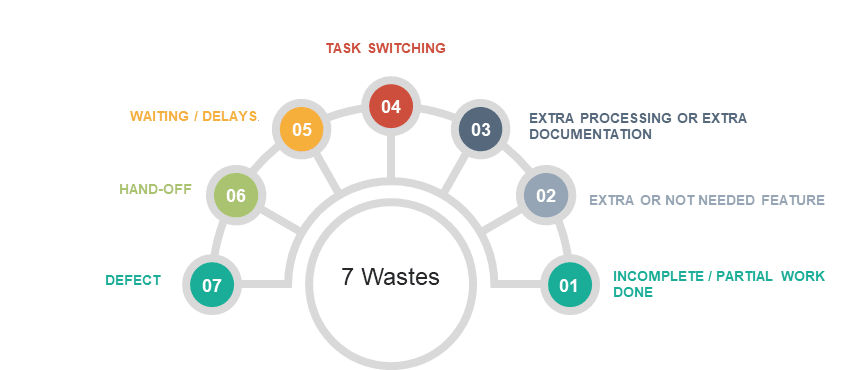


Ilustración : Desperdicios

Una vez que comprendamos los siete desechos, deberíamos poder identificar los desechos en el proceso, independientemente de la industria. Una vez que pueda identificarlo, puede eliminarlo y, por lo tanto, aumentar su eficiencia, productividad e ingresos.

**Ejemplos**

|  |  |
| --- | --- |
| Residuos de la industria de fabricación | Residuos de la industria de desarrollo de software |
| Inventario | incompleto / trabajo parcial realizado |
| Procesamiento adicional | Característica adicional o no necesaria |
| Sobreproducción | Procesamiento adicional de sobreproducción o documentación adicional |
| Transporte | Cambio de tarea |
| retraso | Retraso al entregar |
| Movimiento | manos fuera de la producción |
| Defecto | Errores o software sin pruebas |

**Referencias web**

Wikipedia. (2018). Lean software development. agosto 27, 2020, de Wikipedia Sitio web: https://es.wikipedia.org/wiki/Lean\_software\_development

toolsqa. (2019). Lean Software Development – Comprehensive Guide [2019]. agosto 27, 2020, de toolsqa Sitio web: https://www.toolsqa.com/agile/lean-software-development/