

Guía para Mejora Lean en Servicios



Juan Gabardini



Las oportunidades de mejora en Lean

Si participas en **equipos** que aplican **Kanban** o metodologías **Lean**, es frecuente que te enfoques en la mejora continua.

En su origen, Lean se aplicó a manufactura. Muchos de los ejemplos de mejora continua surgen en este contexto. Esos ejemplos suelen tener dos caras:

- Cómo detectar oportunidades de mejora (Desperdicios)
- Cuáles son las mejoras habituales. (Prácticas Lean)

Cuando trabajamos en Servicios o Desarrollo de software los ejemplos en la bibliografía son difíciles de aplicar a nuestro contexto.

Por ejemplo "Inventario" es un desperdicio en el ámbito de manufactura, y si lo detectamos es probable que podamos mejorar nuestro proceso aplicando "Just in time". Pero ¿Cómo aplica el concepto de "Inventario" a Servicios? ¿Qué significa "Just in time" en ese contexto?

En este mini-libro presentaremos cómo detectar oportunidades de mejora y cómo **mejorar servicios sin** que necesites **entender manufactura**.

Este mini-libro es parte del material utilizado en el curso [Lean y Kanban en Acción](#), de Kleer., Este curso lo hacemos entre Leo Agudelo, Agus Leudesdorf y yo. ¡Gracias Leo y Agus por todas las revisiones y sugerencias!

Además, puede interesarte los siguientes cursos relacionados de nuestra plataforma de e-Learning:

- [Fundamentos de Lean y Kanban](#)
- [Mejora Continua con Toyota Kata](#)
- [Kata de Coaching](#)



Tabla de contenidos

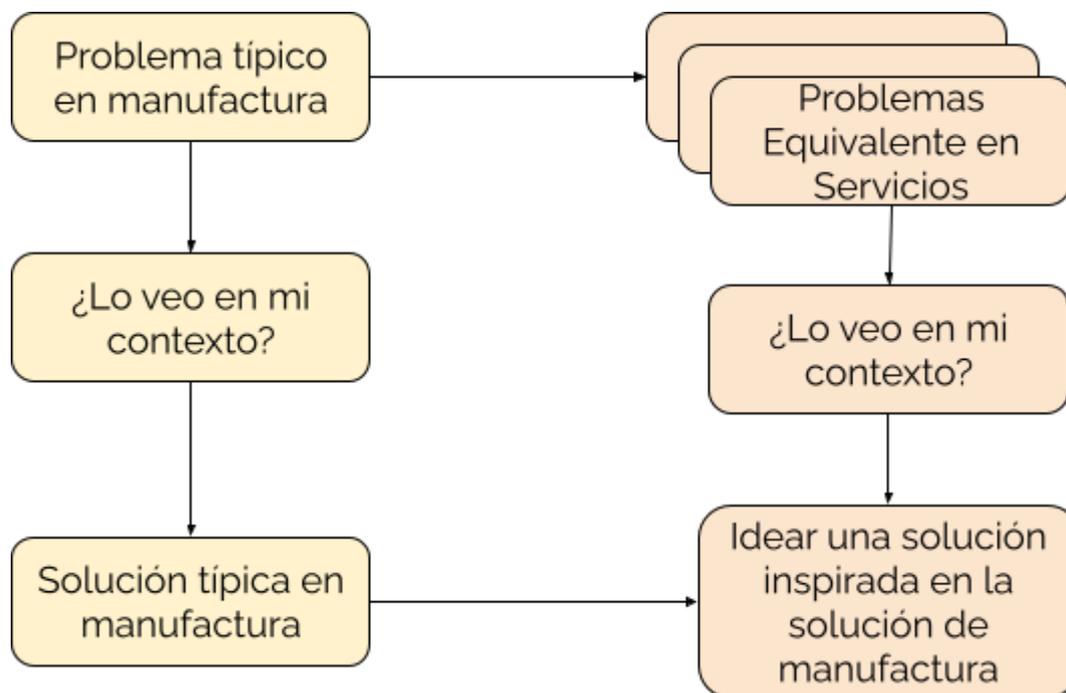
Las oportunidades de mejora en Lean	2
Mejora en Servicios: más allá de la manufactura	4
Sobre problemas y oportunidades de mejora, patrones y olores	4
¿Cómo usamos esta guía?	5
Patrones de oportunidades de mejora en Lean para servicios y software	6
Trabajo pendiente	6
Trasposos / Burocracia	6
Cambio de contexto	7
Esperas	7
Perfeccionismo	8
Sobreproducción	8
Problemas de Calidad	9
Reaprendizaje	9
Patrones de solución Lean / Agile para servicios y software	11
Limitar el Trabajo pendiente	11
Entregas incrementales (Rebanado, slicing o small batches)	12
Incidentes y Problemas	13
Diseño simple	14
A prueba de errores	16
Tareas pequeñas eficientes	18
Automatización Inteligente	20
Conexiones entre Oportunidades de Mejora y Soluciones	22
Lean y Manufactura	24
Por qué los llamados Desperdicios	24
7+1 Tipos de Desperdicios	24
Patrones de mejora en Manufactura	26
Resumen y Conclusiones	28
Próximos pasos	29
Lecturas para profundizar	29

Mejora en Servicios: más allá de la manufactura

La presentación tradicional de la mejora en Lean parte del concepto de desperdicios y, a partir de allí, las mejoras habituales en manufactura.

Pero cuando lo aplicamos a servicios, estos ejemplos solo nos sirven como inspiración. En nuestra situación, trabajando en servicios, nuestro camino es complejo:

1. Entender las problemáticas en manufactura, de cada uno de los tipos de desperdicios. Para nosotros es un ámbito poco conocido.
2. Llevar los tipos de desperdicios al ámbito de servicios, El mapeo, desde manufactura a servicio no es directo, lleva a metáforas forzadas.
3. Entender las soluciones típicas de manufactura. Nuevamente, puede ser difícil tanto entender la solución como imaginar la manera de aplicar la solución a nuestro contexto.
4. Además, hay problemas habituales en servicios que se encuentran en manufactura.



En este material intentamos simplificar la mejora en contexto de servicios, presentando los problemas u oportunidades de mejora directamente en el contexto de servicios

Sobre problemas y oportunidades de mejora, patrones y olores

Un equipo, área u organización tiene un **problema** en su entrega de servicios cuando no logramos alguna cualidad acordada. Por ejemplo, la calidad de nuestro servicio es percibida como mala por nuestros clientes.

Un problema puede ocurrir en forma aislada, por motivos no recurrentes. Pero si vemos que se repite es una **oportunidad de mejora**, ya que, resolviendo la causa raíz de esos problemas haremos que dejen de ocurrir.

Luego de aplicar esta dinámica varias veces, descubrimos **patrones**, oportunidades de mejora que se repiten, causas raíces habituales a los problemas en servicios. Los patrones de oportunidades de mejora pueden ser difíciles de observar directamente. Los servicios y el desarrollo de software son actividades complejas, sociales y relacionadas con el trabajo del conocimiento. Sin embargo, ciertos **indicios** u **"olores"** (Algo huele a podrido en Dinamarca.) pueden servir como señales tempranas de oportunidades de mejora.

¿Cómo usamos esta guía?

Tu proceso de mejora puede ser:

- **Reactivo:** Tenemos un problema que nos afecta mucho, necesitamos resolverlo. Un ejemplo de esta práctica es Genchi Genbutsu (Ir y Ver): Significa "ir al lugar y entender la situación". Esta práctica implica que el equipo y los líderes vayan directamente al lugar de trabajo para entender los problemas de primera mano y tomar decisiones más informadas.
Esta guía ayudará a identificar la causa raíz y posibles soluciones.
- **Proactivo:** Periódicamente dedicamos tiempo a mejorar como equipo. Un ejemplo es aplicando Value Stream Mapping (VSM): Es una herramienta visual que detalla todos los pasos en un proceso y muestra el flujo de materiales e información. Se utiliza para identificar oportunidades de mejora en los procesos.
Esta guía ayudará a identificar varias oportunidades de mejora, evaluar sus impactos y priorizar cuál es prioritaria para ser mejorada.
- **Continuo:** Todas las personas del equipo están atentas a oportunidades de mejora; Un ejemplo es a través de Gemba Walks: La práctica de caminar por el lugar de trabajo (el "gemba", donde ocurren las cosas) para observar los procesos en acción, identificar ineficiencias, y hablar con los que realizan las actividades para obtener feedback y sugerencias.
Esta guía ayudará a tener presente los patrones de oportunidades de mejora identificar.

Patrones de oportunidades de mejora en Lean para servicios y software

Trabajo pendiente

En manufactura: Inventario y trabajo en proceso (*WIP: work in progress*)

Inglés: *Backlog*

Descripción: Cualquier lista de tareas o pedidos, independientemente del grado de avance de las mismas.

Desde Soluciones no puestas a disposición del cliente, pasando por trabajo parcialmente hecho y hasta pedidos sin atender aún.

Problema que causa:

- Se retrasa la obtención de valor (costo de oportunidad)
- Impacto financiero negativo: invertimos dinero en desarrollar parcialmente.
- Nos exponemos a cambios que vuelvan obsoletos o incorrectos el trabajo realizado.
- Costo de mantenimiento: Cada vez que revisamos una lista de pendientes, para entenderlos, evaluar próximos pasos, estimar, priorizar, estamos dedicando tiempo a tareas que no aportan valor.

Ejemplos de situaciones:

- Todas las listas dentro del equipo
 - En servicio: lista de pedidos hechos a otras áreas, listas de tareas.
 - En software: Ítems a desarrollar que ya se ha decidido construir, bugs a corregir, diseños a implementar, producto a probar.
 - Documentación a la espera de ser revisada.
 - Producto no instalado en producción. Soluciones no habilitadas o informadas a los clientes.
 - Solicitudes o reclamos de clientes no resueltos.
-

Traspasos / Burocracia

En manufactura: Transporte

Inglés: *Handoffs*

Descripción: Mover información o decisiones a tomar entre personas u áreas. Separación entre los que tienen la información, los que deciden y los que hacen

Problema que causa:

- Requiere comunicación adicional para compensar por la pérdida de la información y para resolver ambigüedades.
- Sobrecarga la comunicación entre personas y áreas, lo que aumenta la cantidad de temas simultáneos, y lleva a perder el foco.
- Disminuye la autonomía, lo que disminuye la velocidad de respuesta a cambios y dificulta el escalamiento

Ejemplos de situaciones:

- Los jefes toman las decisiones con la información de las personas que están en la operación.

- Silos de conocimiento: personas expertas o áreas funcionales. Cada paso entre silos tiene pérdida de información que requiere esfuerzo adicional para compensar.
 - Estilos de comunicación basados texto (documentos).
 - Requerimiento de múltiples aprobaciones. Suele ocurrir en cultura enfocadas en procedimientos y aversión al riesgo.
-

Cambio de contexto

En manufactura: Movimiento innecesario

Inglés: *Context switch*

Descripción: Trabajar en varios temas en poco tiempo. Puede ser una persona, o equipo, Puede ocasionarse por el objetivo de ocupar todo el tiempo disponible, o maximizar el conocimiento o especialidad. También se da por espacios de trabajo que no tienen en cuenta la ergonomía y usabilidad (física o virtualmente).

Problema que causa: Pérdida de eficiencia, mayores tiempos de para la misma tarea por tiempo dedicado a recuperar la información de contexto, reaprendizajes, mayor complejidad en la gestión del trabajo, estrés, más cantidad de errores. Puede llegar al punto, sobre todo en actividades de alta complejidad, que la actividad nunca se termina.

Ejemplos de situaciones:

- Miembros del equipo iniciando nuevas actividades sin completar la primera debido a dedicación parcial de las personas, esperas a tareas de terceros, múltiples compromisos con clientes atendidos simultáneamente (funcionalidades de un producto de SW, campañas de marketing, etc.)
 - Una persona atendiendo trabajo de creación e interrupciones: Un trabajo de una hs que realiza una persona que es interrumpida cada 15 min no se termina de hacer nunca. <https://hr.berkeley.edu/impact-interruptions>
 - Complejidad adicional en la resolución de la tarea, como cuando se requiere información de diversas fuentes por falta de un reporte que tenga toda la información, o cuando los datos deben ser convertidos para ser útiles, ya sea mentalmente o con aplicaciones adicionales.
 - Movimientos físicos innecesarios, como levantarse del escritorio en el que atiende al cliente para escanear un documento o para buscar una impresión de documentos.
 - En desarrollo de software, tener ramas de versionado de de larga duración en el repositorio de código, que obligan a revisar y aplicar los cambios ya hechos a otras ramas.
-

Esperas

En manufactura: Esperas

Descripción: Colas de espera antes de iniciar una actividad o proceso. Pueden ser una forma de optimizar la ocupación de la capacidad del proceso, de manejar que siempre tenga algo para hacer. También puede aparecer como sincronización con el trabajo de

otras áreas, cuando se requiere la presencia simultánea de dos o más elementos para continuar el proceso.

Problema que causa: Incrementan el tiempo total de entrega y aumenta el esfuerzo dedicado a gestión y seguimiento. Genera optimización local y sobre adaptación (en vez de mejorar el proceso, nos acomodamos al proceso existente)

Ejemplos de situaciones:

- Esperas por aprobaciones externas al equipo o persona
Relacionado con Traspasos / Burocracia
 - Iniciamos dos trabajos necesarios en paralelo y el más rápido debe esperar al más lento.
Relacionado con Trabajo pendiente.
 - Gestión de demanda: Un área recibe pedidos de distintos orígenes, para optimizar el uso de la capacidad del área, se realiza una lista de pedidos que es revisada periódicamente (2 semanas, un mes, un trimestre). Los pedidos deben esperar, en promedio, la mitad del período (1 semana, medio mes o medio trimestre) en el mejor de los casos (cuando todos los pedidos son realizados)
 - Esperas dentro de una actividad: En una actividad en la que colaboran distintas personas o áreas, realizamos comunicación asíncrona. Pasamos un documento a revisar, esperamos la revisión, incorporamos las revisiones o hacemos consultas sobre las mismas. Cada ida y vuelta incluye una espera.
-

Perfeccionismo

En manufactura: Sobre procesamiento

Descripción: Realizamos un trabajo con calidad o detalle mayor del necesario para el uso que le dará el cliente y que no aporta al valor percibido.

Problema que causa: Disminuye la eficiencia, por dedicar trabajo a temas que no dan valor. La pérdida de foco nos puede llevar a no cumplir con lo que realmente hace diferencia. Al aumentar el tiempo dedicado, aumenta el trabajo en proceso (Trabajo no entregado o parcial). Aumenta la complejidad potencialmente aumentando Defectos y por ende, el retrabajo.

Ejemplos de situaciones:

- Generar información o precisión excesiva que no son necesarios para la toma de decisiones. Por ejemplo en reportes o en estimaciones.
 - En software: preocupación por atributos de calidad no relevantes en el contexto actual, como performance en sistemas de bajo uso, usabilidad para volumen (filtros, paginados) de datos cuando los datos son pocos, generalizar innecesariamente (ver los conceptos de YAGNI o KISS)
 - Documentar o registrar información de uso interno que nunca es consultada.
-

Sobreproducción

En manufactura: Sobreproducción

Descripción: Realizar un servicio que no es necesario. Agregar funcionalidades o características en software o servicios sobre lo que no validamos interés o nos

anticipamos su necesidad

En general, producir (diseñar, construir) a partir de supuestos sobre el comportamiento futuro de los clientes (En términos Lean, *push* en vez de *pull*)

Problema que causa: Agregan complejidad que encarece el desarrollo, uso y mantenimiento. Esfuerzo innecesario sin recuperar el valor, es el peor caso de Trabajo pendiente ya que hicimos todo el trabajo, pero no obtenemos el valor.

Ejemplos de situaciones:

- Iniciamos un proceso de selección de personal sobre una planificación anual de roles a cubrir, pero que aún no son necesarios. Los datos de candidatos pierden valor con el paso del tiempo.
- Diseñamos y ponemos a la venta cierto servicio completo sin validar interés. Luego se vende poco.
- Diseñar servicios manejando casos que se requieren poco en la práctica. Creamos funcionalidad en aplicaciones, manuales, material, alternativas de producto, entrenamos a las personas,

Problemas de Calidad

En manufactura: Defectos

Descripción: Problemas durante el proceso de construcción o entrega que impiden que el servicio o producto sea entregado o que genera reclamos del cliente.

Problema que causa: Retrabajo y retrasos para tareas correctivas y preventivas. Pérdida de foco cuando el equipo debe dejar de brindar servicio para corregir los problemas. Pérdida de confianza y satisfacción del cliente.

Ejemplos de situaciones:

- Discrepancias entre las expectativas del cliente y la definición del servicio. Por ejemplo, la descripción comercial del servicio no corresponde con el servicio realizado.
- Alta variabilidad. El servicio no es repetible en forma confiable. Varía por las personas que lo brindan, por el lugar, por el momento, por el tipo de cliente.
- Alta complejidad en la solución, exige trabajar bajo estrés, coordinando entre muchas personas, o con alta carga cognitiva.
- Pérdida de conocimiento de la persona o equipo sobre cómo resolver las solicitudes o cómo construir el producto.

Reaprendizaje

En manufactura: Personal subutilizado

Descripción: Algo muy valioso para las organizaciones son las capacidades y conocimiento de las personas. Las mejores organizaciones aprovechan, mantienen y difunden y mejoran esas capacidades y habilidades.

Problema que causa: Retrasos e ineficiencias para reaprender lo que ya sabíamos hacer. Problemas de Calidad por no aprovechar las experiencias pasadas. Desmotivación, por tener que resolver problemas básicos, no poder construir sobre el saber de otros.

Ejemplos de situaciones:

- Paso del tiempo: Hemos resuelto un problema poco habitual. Las personas que lo hicieron no recuerdan lo aprendido
- Distintas áreas: Otra área o personas de la organización resolvió el problema, no hay mecanismos o incentivos para aprovechar ese aprendizaje.
- Rotación: Las personas que resolvieron el problema ya no están en el área o en la organización.
- No inventado acá (*NIH: not invented here*): Aunque sabemos de que un problema similar fue resuelto, no creemos que apliquen los aprendizajes fuera de su equipo, área o incluso fuera de la organización.

Patrones de solución Lean / Agile para servicios y software

Limitar el Trabajo pendiente

Descripción: Acordamos un límite al trabajo que aceptamos hacer y solo hacemos seguimiento a lo aceptado.

Acordamos que el grupo tiene una capacidad limitada, por lo tanto aceptar pedidos que excedan esa capacidad generará problemas.

Se puede implementar inicialmente de una manera abrupta (todo lo que está fuera lo aceptable sale de nuestro trabajo pendiente) o incremental (los nuevos acuerdos cumplen con la nueva regla).

Beneficios:

- El grupo mejorar la capacidad gracias a estar más enfocado y dedicar menos tiempo a actividades que no aportan valor (gestión de Trabajo pendiente)
- Visibiliza diferencias entre la capacidad real del equipo y las expectativas y necesidades de los cliente

Problemas secundarios:

- Requiere visibilizar los límites (la capacidad del equipo) que llevan a decisiones (aceptar menos o ampliar la capacidad). En algunas organizaciones estas decisiones requieren un cambio cultural y de liderazgo.
- En ámbito de servicios y desarrollo de software, la capacidad no es fácil de medir y en muchos casos no es conocida.

Ejemplos de situaciones:

- Todas las listas dentro del equipo
 - En servicio: lista de pedidos hechos a otras áreas, listas de tareas.
 - En software: Ítems a desarrollar que ya se ha decidido construir, bugs a corregir, diseños a implementar, producto a probar.
- Documentación a la espera de ser revisada.
- Producto no instalado en producción. Soluciones no habilitadas o informadas a los clientes.
- Solicitudes o reclamos de clientes no resueltos.

Entregas incrementales (Rebanado, *slicing* o *small batches*)

Descripción: Consiste en dividir el trabajo en partes pequeñas y manejables de tamaño similar. Este enfoque permite que el proceso fluya de manera más eficiente, reduce la necesidad de estimaciones complejas y facilita un aprendizaje más frecuente.

Beneficios:

- Mejora el flujo de trabajo y reduce los cuellos de botella.
- Facilita la identificación temprana de problemas y la implementación de mejoras.
- Entrega de valor al cliente en forma temprana y frecuente.
 - Reduce el riesgo asociado con grandes entregas.
 - Mejora la respuesta a cambios en las necesidades o prioridades.

Problemas secundarios:

- Requiere un cambio en equipos acostumbrados a realizar entregas "completas"
- La división excesiva puede complicar la gestión si no se maneja adecuadamente.
- Algunos stakeholders pueden tener dificultades para visualizar el valor de entregas muy pequeñas.

Ejemplos de situaciones:

- En desarrollo de software:
 - Dividir una funcionalidad grande en múltiples historias de usuario más pequeñas.
 - Implementar integración continua y despliegue continuo (CI/CD) para facilitar las entregas frecuentes.
 - Realizar refactorizaciones pequeñas y continuas en lugar de grandes reescrituras.
- En servicios:
 - Procesar solicitudes de clientes en pequeños lotes a lo largo del día en lugar de acumularlas para un procesamiento masivo.
 - Realizar actualizaciones incrementales de documentación en lugar de revisiones completas y extensas.

Recursos

- Cartas de slicing
<https://www.kleer.la/es/recursos#slicing-cards>

Incidentes y Problemas

Descripción: Identificar tempranamente los defectos (en el producto y en el proceso).

Dividir la respuesta en dos partes:

- **Incidente:** Resolver los defectos en forma inmediata para el caso detectado.
- **Problema:** Además, dedicar tiempo a resolver el origen del incidente, la causa raíz, para que no vuelva a pasar.

Beneficios:

- Disminuye el impacto del defecto en los usuarios o clientes.
- Resolver el incidente tempranamente permite obtener mejor información para identificar y corregir la causa raíz.
- Dividir la respuesta permite tener un tiempo de respuesta rápido para el caso puntual y gestión planificada y priorizada para el análisis de causa raíz y corrección.

Problemas secundarios:

- Es necesario simplificar y automatizar la detección de defectos (monitoreo).
- El monitoreo debe ser relevante, las alertas toman tiempo y si hay muchos falsos positivos se pierde la confianza.
- La responsabilidad de atender incidentes y problemas debe ser clara y acordada, tanto internamente (quién del equipo lo atiende), como externamente (tiempo de respuesta y distinción entre respuesta al incidente o corrección del problema)

Ejemplos de situaciones:

- En desarrollo de software:
 - Integración continua para detectar inmediatamente problemas de integración.
 - Monitoreos y Alertas sobre ambiente productivo, por ejemplo para detectar servicios no disponibles, memoria o capacidad de procesamiento insuficiente.
 - Validación de datos: en Tableros construidos con datos de múltiples sistemas, detección de errores en datos que no cumplen con los supuestos.
 - En cuanto a la corrección de problemas, cambiar el diseño de manera que no vuelva a ocurrir ese problema (Design-out failure modes).
- En servicios:
 - Gestión de espera de los pedidos de servicios: monitoreo y alertas de tiempos de espera, cantidad de pedidos en espera.

Recursos

- Agile testing en desarrollo de Datawarehouse - Calidad de datos
<https://kleer.la/es/blog/agile-testing-en-desarrollo-de-data-warehouse-calidad-de-datos>

Diseño simple

Descripción: Diseñar el producto, servicio o proceso que **resuelva la necesidad actual**. Tratar de hacer contemplar necesidades futuras tiene múltiples impactos negativos que no estamos seguros que generen valor.

Adicionalmente, aún atendiendo necesidades actuales, un diseño simple es valioso. La mejor pieza es no tener la pieza. El mejor paso del proceso es no tener el paso. (*Best part is no part. Best process is no process*)

También se lo describe como:

- YAGNI (you ain't gonna need it): No lo vas a necesitar. Se puede cambiar el diseño (*) para resolver una nueva necesidad, pero no podemos recuperar el tiempo dedicado a hacer algo que no se usa.
"Prediction is very difficult, especially if it's about the future" (La predicción es muy difícil, especialmente si se trata del futuro), Niels Bohr
- KISS (Keep it simple, stupid/smarty): Manténlo sencillo. Con el deseo de lograr un diseño que nos llene de orgullo, podemos agregar complejidad innecesaria.
En la imagen se ve un huevo Fabergé, un ejemplo de complejidad y detalle solo para demostrar la habilidad del orfebre.



Beneficios:

Disminuye la complejidad, lo que, a su vez, tiene estos beneficios:

- Menor costo cognitivo para el equipo de diseño y mantenimiento, puede ser más chico o tomar más temas.
- Más rápido de hacer, menos costoso de mantener.
- Menos defectos.

Problemas secundarios:

- (*) No siempre se puede modificar el diseño para resolver una nueva necesidad. Cuando aparecen nuevas necesidades, el diseño deberá cambiar para tenerlas en cuenta. Para ello, debe ser sencillo y no muy costoso modificar el diseño existente.
- A lo largo del ciclo de vida del producto o servicio, y luego de extender el diseño varias veces, puede ser necesario revisar en forma holística para encontrar un nuevo diseño simple y elegante que resuelva la necesidad inicial más las que surgieron luego.

Ejemplos de situaciones:

- En desarrollo de software:

- Las prácticas de XP relacionadas (Programación de a pares, TDD, Diseño simple y Refactorio) habilitan a que apliquemos Diseño simple cuidando que no surjan los problemas secundarios descriptos.

- En servicios y procesos:
 - Antes de optimizar un paso, evaluar si es necesario.
Por ejemplo, un proceso de validación, evaluar cuánto cuesta la validación, cuánto cuesta la situación "invalida", cuán frecuente es la situación "inválida".
 - Eliminar variantes del proceso o servicio.
Por ejemplo, simplificando la oferta de platos en un restaurante, enfocándose en los más populares y rentables. Esto simplifica la cocina, reduce el inventario y mejora la eficiencia del servicio.
Esto es particularmente útil cuando estamos en una situación de limitación de capacidad (tenemos demanda no cubierta). En productos, es el ejemplo del Ford T (solo negro) y de iPhone (pocas versiones disponibles)

A prueba de errores

Descripción: Mecanismos que prevengan errores antes de que ocurran o se detecten inmediatamente para evitar que se propaguen.

El objetivo es hacer difícil o imposible cometer errores, reduciendo así la necesidad de inspecciones posteriores y mejorando la calidad desde el origen.

También se lo describe como:

- Prevención de errores: Diseñar sistemas y procesos que eliminen la posibilidad de cometer errores comunes.
- Detección temprana (también *shift-left*): Implementar controles que identifiquen los errores cuando ocurren, permitiendo una corrección inmediata.

Beneficios:

- Reducción de errores y defectos en el producto o servicio final. Evita el retrabajo del usuario y de las personas que dan el servicio
- Aumento de la eficiencia al disminuir la necesidad de revisiones y correcciones posteriores.
- Disminución del estrés y la frustración del equipo al reducir errores repetitivos.

Problemas secundarios:

- La implementación inicial de los mecanismos a prueba de error puede requerir tiempo y recursos adicionales.
- Riesgo de crear sistemas demasiado rígidos que dificulten la innovación.
- Puede generar una falsa sensación de seguridad si no se mantienen y actualizan regularmente los mecanismos de prevención de errores.

Ejemplos de situaciones:

- En desarrollo de software:
 - Desarrollo de a pares, lo que implica revisiones de código en el momento de la construcción.
 - Análisis estático de código y tipado estático en lenguajes de programación para prevenir errores tempranamente.
 - Especificaciones ejecutables (TDD - *Test Driven Development* y BDD = *Behavior Driven Development*) permite definir el comportamiento esperado de la aplicación antes de construirla.
 - Usar pruebas automatizadas (unit tests, integration tests) para verificar el comportamiento correcto del código.
- En servicios y procesos:
 - Utilizar listas de verificación (checklists) para asegurar que todos los pasos críticos se completen en un proceso.

- Diseñar experiencias de usuario que solo permitan las acciones correctas y prevengan errores comunes. Por ejemplo, formularios con validación de campos para prevenir la entrada de datos incorrectos.

Tareas pequeñas eficientes

Descripción: Lograr realización eficiente de tareas pequeñas, lo que permite a una organización ser al mismo tiempo adaptativa y sostenible.

Para ello se requiere tanto empezar (*setup*) como finalizar (*clean up*) una tarea en poco tiempo y bajo costo.

En manufactura, esto se logra con SMED (*Single-minute Exchange Die*) que permite a una línea de producción pasar de un producto a otro en poco tiempo, habilitando lotes pequeños de producción.



Beneficios:

- Para la organización:
 - Mayor flexibilidad para responder a cambios en las demandas del mercado o del cliente.
 - Mayor capacidad de innovación al poder experimentar con nuevas ideas de forma ágil
- Para el cliente:
 - Reducción de tiempos de espera y mejora en los tiempos de entrega
- Para el equipo
 - Incremento en la motivación del equipo al ver resultados más frecuentemente
 - Reducción del estrés, ya que el foco en un entregable pequeño disminuye el multitasking y los picos de esfuerzo asociado a grandes tareas o proyectos.

Problemas secundarios:

- Complejidad de la planificación y coordinación de múltiples tareas pequeñas
- Perder la visión global y largo plazo por el foco en iniciativas de corta duración.
- Balance entre
 - Esfuerzo y costo adicional para adaptar procesos y herramientas
 - Riesgo de perder la economía de escala.
- Requiere personas generalistas, lo que puede generar:
 - Resistencia al cambio por parte de empleados.
 - Potencial pérdida de profundidad y especialidad en las personas y en la organización.

Ejemplos de situaciones:

- En desarrollo de software:
 - Setup de ambiente: un desarrollador puede empezar a aportar en un producto en menos de 5 min, por medio de Docker o instalación con un solo comando.

- El tiempo entre el cambio confirmado (subido al repositorio de código) hasta el cambio disponible para los usuarios es de menos de 10 min (*Continuous Delivery*).
- El tiempo de resolución de un bug o incremento de funcionalidad es de menos de un día, gracias a Desarrollo de a pares (*Pair programming*) o En ensamble (*mob programming*).
- El inicio de una versión o proyecto se acelera gracias a la colaboración sincrónica de responsables de producto, sponsors, stakeholders y equipo de desarrollo, por ejemplo en un taller de Incepción.
- En servicios y procesos:
 - Optimización del inicio administrativo de proyectos. Por ejemplo, en una empresa petrolera se requiere que todo el proceso de perforación de un pozo de fracking sea menor a dos meses, dado la vida útil del pozo. Y en general, si los procesos de contratación a proveedores son largos (un par de meses), no tiene sentido pensar en proyectos de menos de tres meses que incluyan proveedores.
 - En procesos de recursos humanos, por ejemplo, capacitar a una persona para una nueva tarea o tomar un nuevo rol. Si logramos que sea en poco tiempo, fomentaremos a personas que se desarrollen por fuera de su espacio actual de habilidades y capacidades.
 - Para que un oficial de cuenta bancario pueda atender a un cliente, lo ayudaría tener en forma ordenada y fácilmente accesible toda la información del mismo: situación actual, historia, problemas anteriores, posibles productos que pueden ser de su interés.

Automatización Inteligente

Descripción: Implementar sistemas automatizados para tareas repetitivas y procesos estandarizados, mientras se mantiene la intervención humana en puntos críticos para reflexión, mejora continua y toma de decisiones estratégicas.

Esta práctica busca combinar lo mejor de la eficiencia de las máquinas con la creatividad y adaptabilidad humana. El objetivo es liberar a las personas de tareas monótonas para que puedan enfocarse en actividades de mayor valor añadido.

También se lo llama *Jidoka* y *Autonomation*, en la bibliografía de Toyota Production System y Lean.

Beneficios:

- Mayor consistencia en los procesos y reducción de errores en tareas repetitivas
- Incremento de la productividad y eficiencia
- Liberación del tiempo del personal para tareas de mayor valor, por ejemplo mejora continua facilitada por el análisis de datos automatizados

Problemas secundarios:

- La automatización y gran número de monitores puede llevar a las personas a confiarse en exceso y perder la habilidad de entender el proceso como para mejorarlo o para resolver condiciones de excepción.
- Si la automatización es realizada por personas distintas a las que hacen el monitoreo, es difícil que evolucione y mejore.

Ejemplos de situaciones:

- En desarrollo de software:
 - Entrega continua (CD - *Continuous delivery*): Tener automatizado todo el proceso desde que se sube un cambio al repositorio hasta que queda en manos de los usuarios, incluyendo pruebas automatizadas. En caso de falla en el CD, el equipo responsable del producto debe dedicarse con prioridad a resolver el problema.
 - Monitoreo y alertas en producción, Por ejemplo, validación continua de condiciones problemáticas en producción. Por ejemplo, si generamos indicadores de gestión que requieren que los datos estén actualizados, generar una alarma si esto no se cumple. Ver con más detalle en [Agile Testing en el Desarrollo de Data Warehouse: Calidad de datos](#)
- En servicios y procesos:
 - Automatización de flujos de trabajo y procesos, por ejemplo automatizar un proceso de abastecimiento que interactúa con sistemas internos y envía automáticamente pedidos de cotización a los proveedores, procesa las respuestas y realiza la compra automáticamente en las situaciones normales, requiriendo intervención cuando alguna condición no se

cumple (no hay suficientes proveedores calificados o alguno responde con una consulta)

- Algunas formas de automatizar procesos con herramientas de software:
 - Generales: Zapier, Microsoft Power Automate, IFTTT, Google Docs,
Por ejemplo, una persona responde a un formulario Google (un evento) y respondemos enviando un mail personalizado con información (respuesta)
 - BPM (*Business Process Management*): Appian, Pega Platform, Camunda, IBM Business Automation Workflow
Por ejemplo, un cliente pide un préstamo y el proceso va indicando cada paso que debe realizarse avisando al sector o persona responsable de hacerlo (información del cliente, análisis de riesgo, aprobación, alta del préstamo), mientras el estado completo del proceso es visible.
 - RPA (*Robotic Process Automation*): UiPath, Automation Anywhere
Por ejemplo, un bot de RPA extrae datos de facturas PDF, los ingresa en un sistema de contabilidad y genera un informe de gastos, todo de forma automática sin intervención humana.
 - No code: Airtable, Bubble
Por ejemplo, un equipo de marketing crea una aplicación de gestión de campañas usando Bubble, con un formulario para ingresar detalles de la campaña, un dashboard para visualizar métricas y notificaciones automáticas, todo sin escribir código.

Conexiones entre Oportunidades de Mejora y Soluciones

Oportunidad de mejora	Solución	¿Por qué?
Trabajo pendiente	Limitar el Trabajo pendiente	Establece un límite claro en la cantidad de trabajo en proceso, evitando la acumulación excesiva de tareas y mejorando el flujo.
	Entregas incrementales	Dividir el trabajo en partes más pequeñas y manejables, reduce la cantidad de trabajo pendiente y facilita entregas más frecuentes.
Trasposos / Burocracia	Diseño simple	Simplifica los procesos y reduce la necesidad de trasposos innecesarios entre personas o departamentos.
	Automatización Inteligente	Automatiza procesos repetitivos y estandarizados, reduciendo la necesidad de intervención manual y los trasposos asociados.
Cambio de contexto	Tareas pequeñas eficientes	Permite enfocarse en tareas específicas y completarlas rápidamente, reduciendo la necesidad de cambiar constantemente entre diferentes contextos.
	Limitar el Trabajo pendiente	Al reducir la cantidad de trabajo en proceso, disminuye la presión de cambiar frecuentemente entre tareas.
Esperas	Automatización Inteligente	Reduce los tiempos de espera al automatizar procesos y flujos de trabajo, permitiendo una ejecución más rápida y continua.
	Entregas incrementales	Permite avanzar con partes más pequeñas del trabajo, reduciendo las esperas asociadas a entregas más grandes y complejas.
Perfeccionismo	Diseño simple	Fomenta la creación de soluciones que cumplen con los requisitos sin agregar complejidad innecesaria.
	Incidentes y problemas	Responder rápidamente a los errores y luego tomarnos tiempo para mejorar en lo que se necesita quita la presión de hacerlo "bien" desde el inicio.

Oportunidad de mejora	Solución	¿Por qué?
Sobre-producción	Entregas incrementales	Permite ajustar la producción basándose en feedback más frecuente, evitando la sobreproducción de características no necesarias.
	Limitar el Trabajo pendiente	Ayuda a controlar la cantidad de trabajo en proceso, evitando la producción excesiva de elementos no inmediatamente necesarios.
Problemas de Calidad	A prueba de errores	Implementa mecanismos para prevenir errores, mejorando la calidad del producto o servicio desde el inicio.
	Incidentes y Problemas	Permite una respuesta rápida a los problemas de calidad y facilita el análisis de causa raíz para prevenir futuros problemas.
Reaprendizaje	Automatización Inteligente	Captura y codifica el conocimiento en sistemas automatizados, reduciendo la necesidad de reaprender procesos.
	Tareas pequeñas eficientes	Facilita la documentación y transferencia de conocimiento al trabajar en tareas más pequeñas y manejables.

Lean y Manufactura

En este mini-libro, hemos presentado las oportunidades de mejora Lean sin referencias a manufactura.

Pero puede ser útil conocer el origen en manufactura, ya que agregan una dimensión que ayuda a la comprensión y para encontrar material adicional en bibliografía.

Por qué los llamados Desperdicios

En manufactura se llaman Desperdicios (*waste* en inglés) a las oportunidades de mejora de los procesos. Toma el nombre de la pérdida de material que se produce en los cortes o maquinado. En la foto vemos los recortes de tela, no utilizables, que se producen por el corte. Pagamos por toda la tela pero los recortes no van al producto. Es un desecho o desperdicio (*waste*).



<https://onlineclothingstudy.com>

7+1 Tipos de Desperdicios

La descripción de de Lean en manufactura incluye los siguientes Desperdicios

Sobreproducción: Producir más de lo necesario o antes de que sea necesario. El peor caso es que deba descartarse el producto por obsolescencia o falta de mercado. También causa mayor inventario, y puede ocultar problemas como defectos y retrasos en la producción.

Transporte: Los materiales, productos o incluso la información se transportan más de lo necesario. El transporte excesivo provoca tiempos y costos que no agregan valor al producto. También causa mayor inventario con riesgo de daños (Defectos) y retrasos (Esperas).

Espera: El tiempo en el que los recursos (empleados, máquinas, materiales) están inactivos esperando el siguiente paso en el proceso. Esto disminuye la eficiencia y puede aumentar los costos operativos. También causa mayor inventario.

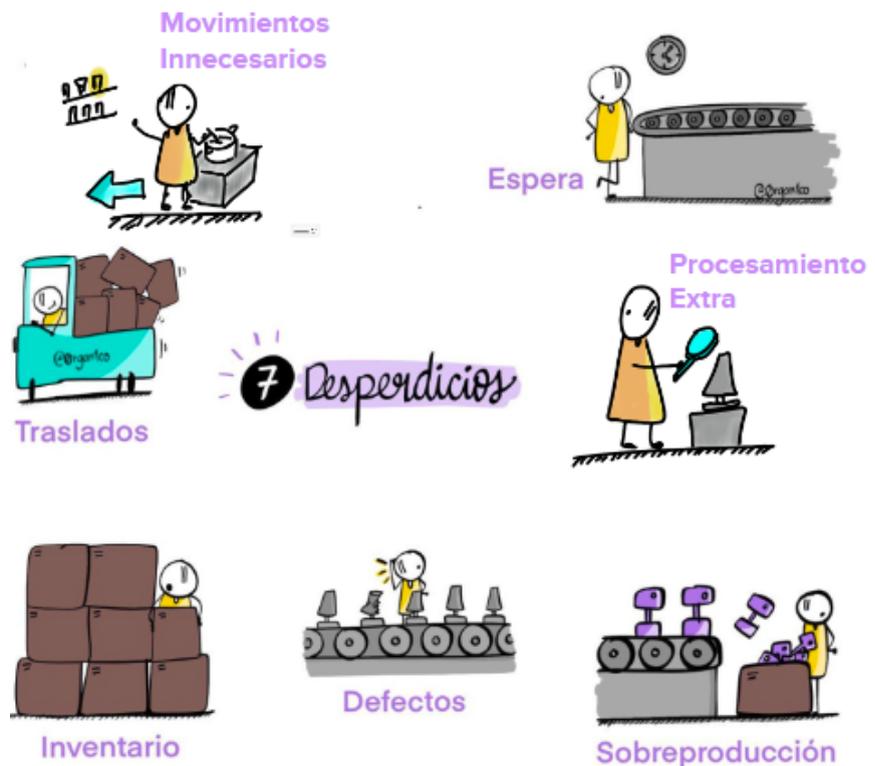
Proceso extras: Se refiere a realizar más trabajos o procesos más complejos de lo necesario para satisfacer los requisitos del cliente. Esto puede incluir pasos innecesarios en la fabricación o en la prestación de servicios.

Inventario: Mantener más inventario del necesario, ya sean materias primas, productos en proceso o productos terminados. El inventario ocasiona costos adicionales de almacenamiento, costos financieros y aumenta el riesgo de obsolescencia y deterioro.

Movimientos innecesarios: Cualquier movimiento de personas o máquinas que no agregue valor al producto. Esto incluye moverse para alcanzar una herramienta o puesto de trabajo, actividades para calibrar una máquina, caminar, o moverse de manera ineficiente, lo que puede llevar a una mayor fatiga y a una menor productividad.

Defectos en el producto: Cualquier error o defecto en los productos que requiera retrabajo o desecho. Esto representa un desperdicio de materiales y tiempo, y puede afectar negativamente la satisfacción del cliente.

Personal subutilizado: Las habilidades, talentos y conocimientos de los empleados son muy valiosos. Si no los aprovechamos logramos una menor innovación, moral baja, eficiencia reducida y rotación con pérdida del conocimiento. Este desperdicio es un agregado más reciente y tiene mucho en común con lo visto antes en Servicios



Dibujado por Jorge Rolan (organ1co)

Patrones de mejora en Manufactura

Varias prácticas y herramientas habituales en la mejora surgieron a partir de la aplicación de mejora continua en manufactura respondiendo a los desperdicios comentados en la sección anterior.

Jidoka (Automatización con control de personas): Este principio implica detectar tempranamente y corregir problemas. Al menos la detección de la anomalía es automatizada. La corrección puede ser parcialmente automatizada, involucrando a los operarios solo cuando es necesario. Disminuye los Defectos.

Andon (Pausa en la producción): Es una parada de la producción activada por cualquier operario de la línea de producción. El operario que detecta un problema de calidad (creado o encontrado), un problema de seguridad o algún otro problema en el proceso (falta de partes, equipo defectuoso) activa el andon y recibe inmediatamente ayuda para resolver el problema de la manera más rápida que permita reiniciar la producción. Inmediatamente después, el líder del equipo que detectó el problema analiza el problema para detectar la causa raíz. Relacionado con Jidoka. Disminuye Defectos y Personal subutilizado.

Estandarizar el proceso: Documentar las mejores prácticas actuales y ser la base para las mejoras futuras. Esto ayuda a reducir la variabilidad, mejorar la calidad y facilitar la formación y el trabajo en equipo. Disminuye el Personal subutilizado y los Defectos.

Poka-Yoke (Sistemas anti-error): Prevenimos errores humanos gracias al diseño en el producto, las herramientas o el proceso. Disminuye los Defectos.

TPM (Total Productive Maintenance): Es mantenimiento proactivo para maximizar la fiabilidad y eficiencia de las máquinas mediante el diseño, operación y mantenimiento de las mismas. Participan desde la alta dirección hasta los operarios. Disminuye la Espera, los Movimientos innecesarios y los Defectos.

Heijunka (Nivelación de la Producción): Es la práctica de suavizar la producción al nivelar el tipo y la cantidad de producción durante un período específico. Esto ayuda a reducir la carga de trabajo y los inventarios, y facilita un flujo de trabajo más predecible.

Producción en flujo constante: Implica la organización de la producción de manera que los productos se muevan continuamente a través del proceso, reduciendo el tiempo de espera, los inventarios y los retrasos, y mejorando la eficiencia.

Control visual - Kanban: Kanban es un sistema de señalización visual que controla la cantidad de producción y mantiene el flujo de trabajo, asegurando que solo se produce lo que se necesita y cuando se necesita.

5S: Es una metodología que promueve un entorno de trabajo organizado, limpio y eficiente. Las 5S son: Seiri (clasificar), Seiton (ordenar), Seiso (limpiar), Seiketsu (estandarizar) y Shitsuke (sostener).

Mejora del entorno de trabajo: Implica realizar cambios en el entorno laboral para mejorar la seguridad, la eficiencia y el bienestar de los trabajadores. Esto puede incluir mejoras ergonómicas, reducción de ruidos y otros ajustes para crear un entorno más productivo y agradable.

Muda, Mura, Muri (Eliminación de Desperdicios, Irregularidades y Sobrecarga): Muda se refiere a eliminar los desperdicios, Mura a eliminar las irregularidades en los procesos y Muri a evitar la sobrecarga de trabajadores y equipos. Juntas, estas prácticas buscan crear procesos más eficientes y sostenibles.

SMED (Single-Minute Exchange of Dies): Se refiere a reducir el tiempo y el esfuerzo necesarios para cambiar de una tarea o producto a otro. Esta técnica es particularmente útil en entornos de fabricación para reducir tiempos de inactividad y aumentar la flexibilidad.

Resumen y Conclusiones

Este mini libro ha explorado la **aplicación de los principios Lean en el contexto de servicios y desarrollo de software**, adaptando conceptos originalmente desarrollados para la manufactura. No intenta ser una descripción exhaustiva, sino centrada en los Desperdicios.

Hemos cubierto:

- Patrones de en Lean para servicios y software.
Te dejamos una tabla de **oportunidades de mejora** con los Desperfectos en Manufactura más cercanos

En Servicios y software	En manufactura
Trabajo pendiente	Inventario
Trasposos / Burocracia	Transporte
Cambio de contexto	Movimientos innecesarios
Esperas	Esperas
Perfeccionismo	Proceso extra
Sobreproducción	Sobreproducción
Problemas de Calidad	Defectos en el producto
Reaprendizaje	Personal subutilizado

- Patrones de **solución** Lean / Agile para servicios y software:

En Servicios y software	En manufactura
Limitar el Trabajo pendiente	Just-in-Time (JIT)
Entregas incrementales	Producción en pequeños lotes, Heijunka
Incidentes y Problemas	Andon (sistema de notificación de problemas)
Diseño simple	
A prueba de errores	Poka-yoke
Tareas pequeñas eficientes	SMED (Single-Minute Exchange of Die)

En Servicios y software	En manufactura
Automatización Inteligente	Jidoka (automatización con un toque humano), Estandarizar el proceso

- Conexiones con los conceptos originales de Lean en manufactura

Puntos clave a recordar:

- La **mejora continua** es fundamental en la filosofía Lean, independientemente del sector.
- Los principios Lean pueden **adaptarse exitosamente a servicios y desarrollo de software**.
- La identificación de oportunidades de mejora y la aplicación de soluciones adecuadas son **procesos continuos**.
- La participación activa de **todo el equipo** es crucial para el éxito de las iniciativas Lean.

Próximos pasos

1. Reflexiona sobre cómo estos conceptos se aplican a tu contexto específico.
2. Identifica una o dos oportunidades en tu trabajo o proyecto actual.
3. Inicia un pequeño experimento aplicando una de las soluciones propuestas.
4. Mide los resultados y ajusta según sea necesario.

Recuerda que la mejora Lean es un viaje, no un destino. Cada paso que des hacia la eficiencia te acercará a un servicio o producto de mayor calidad y a un equipo más satisfecho y productivo.

Lecturas para profundizar

Implementing Lean Software Development, Mary and tom Poppendieck, 2006, Addison Wesley

Leading Lean Software Development, Mary and tom Poppendieck, 2010, Addison Wesley