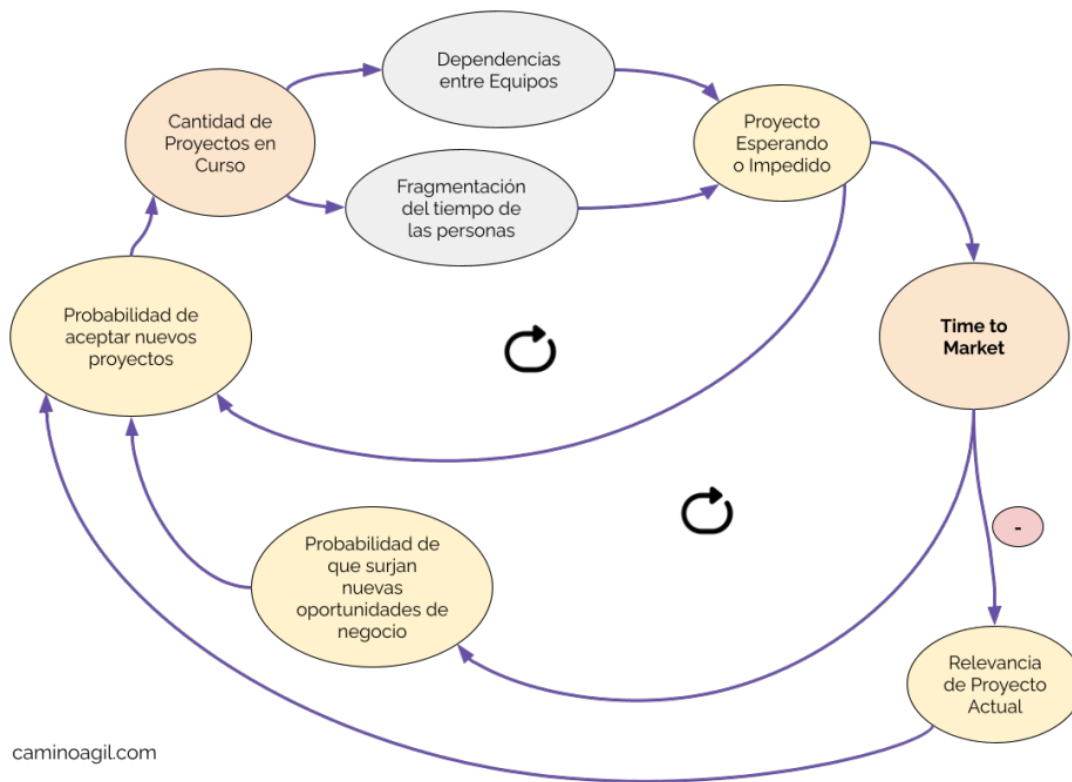


Por qué los proyectos se demoran y cómo resolverlo en 4 pasos

[in linkedin.com/pulse/por-qué-los-proyectos-se-demoran-y-cómo-resolverlo-en-buonamico](https://www.linkedin.com/pulse/por-qué-los-proyectos-se-demoran-y-cómo-resolverlo-en-buonamico)



Damián Buonamico

Agile Coach - 12 years of experience

May 27, 2021

¿Percibes que los proyectos en tu organización se demoran cada vez más y necesitas entregar resultados? No estás solo. Conoce las posibles causas y cómo resolver este problema de manera sistémica en cuatro pasos.

Hace unos años trabajé para un importante banco de Buenos Aires. Estaban desarrollando un nuevo sistema para la red de cajeros automáticos que reemplazaría al actual (aún operativo, pero ya obsoleto y sin soporte del proveedor)

Cuando me convocaron, el proyecto tenía dos años de demoras y renegociaciones de plazos. Como el nuevo sistema no llegaba, lo que inicialmente era una migración tecnológica derivó en nuevos proyectos: no podían esperar más, se hacía necesario

implementar nuevas funcionalidades en el sistema vigente, más el doble esfuerzo de replicarlo en el nuevo.

En esos años habían probado prácticamente todo: pagar horas extras, reemplazar a la mayoría de las personas, ampliar la capacidad con más equipos, capacitar a las nuevas personas, cambiar al gerente a cargo, ajustar el alcance y una pseudo-implementación de Scrum.

La desconfianza y las relaciones dañadas generaban más reuniones de status y control que quitaban tiempo de desarrollo. Los equipos se sentían muy presionados y estresados. Sin embargo, el proyecto no parecía avanzar. ¿Qué estaba fallando?

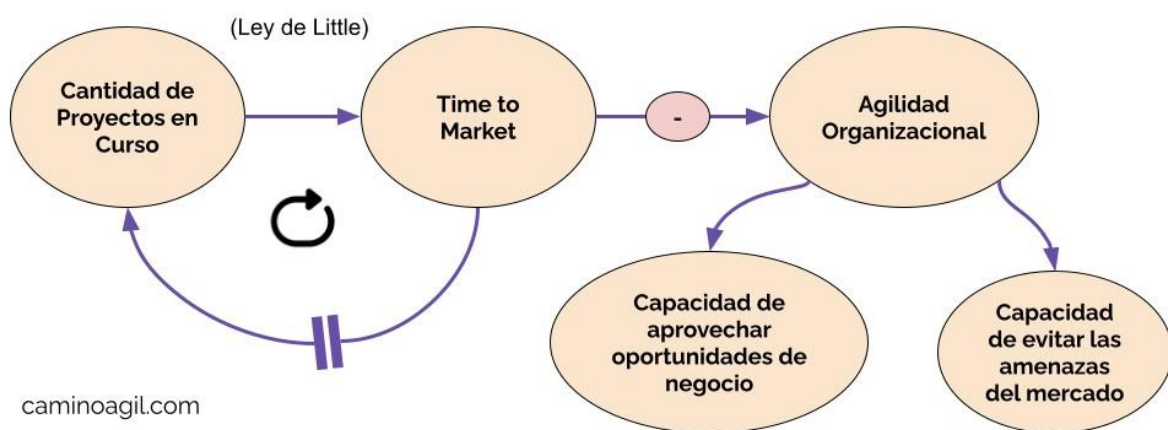
En este artículo te ofrezco una mirada integral de la situación, aplicando modelado sistémico para entender el problema y divisar las soluciones.

Time to Market y Agilidad

El “Time to Market” (TTM) o (“Lead Time”, en terminología Lean) es el tiempo que transcurre desde el compromiso de iniciar un proyecto hasta que se completa.

La conocida “Ley de Little” explica la relación sistémica entre la cantidad de proyectos en curso y el TTM: **A mayor cantidad de proyectos en curso, mayor será el tiempo promedio de entrega de proyectos.**

El siguiente gráfico muestra un análisis sistémico a alto nivel, donde vemos un ciclo vicioso: a mayor TTM, eventualmente habrá más proyectos en curso. Como vimos en el ejemplo del Banco.



¿Cuál es la relación con la agilidad? Hoy las organizaciones necesitan adaptarse rápidamente para aprovechar oportunidades con ventanas de tiempos cada vez más cortas y para evitar amenazas de competidores. Para desarrollar esta capacidad de adaptación es fundamental lograr tiempos de entrega reducidos.

Si los proyectos se demoran, la organización pierde su capacidad de entregar valor y adaptarse.

¿Cuáles son los causantes de las demoras?

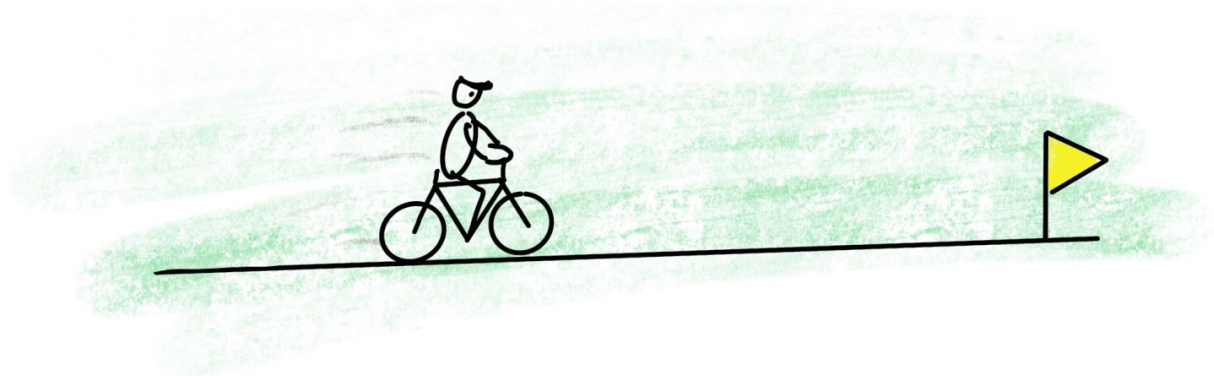
Si bien de forma aislada cada causa hace su aporte en demorar los proyectos, lo realmente alarmante está en los efectos sistémicos que refuerzan el problema con ciclos viciosos. Haciendo que los proyectos se demoren cada vez más.

Causa 1: Compromisos poco realistas

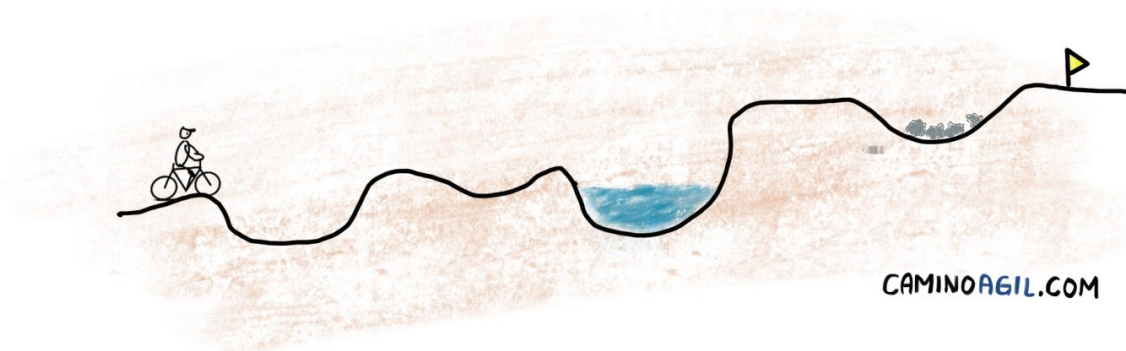
En los proyectos donde hay incertidumbre (como en el desarrollo de software), **intentar predecir desde el comienzo el alcance y la duración es tan arriesgado como pronosticar el clima del mes que viene.**

En ocasiones los proyectos comienzan con el pie izquierdo: cuando los equipos se comprometen a cumplir plazos y alcances demasiado optimistas. Ya sea por presión externa o por el fenómeno conocido como “wishful-thinking” -un sesgo cognitivo de pensamiento optimista que nos hace asumir que no habrá imprevistos.

PLAN



REALIDAD



Si asumimos las estimaciones como un hecho seguro, tomamos ciertas decisiones y compromisos en base a esto. Cuando el plan no se cumple las consecuencias impactan en el proyecto: se vuelve a planificar con nuevas estimaciones, esta vez con más nivel de detalle. La confianza se reduce y se agregan mecanismos de control: reuniones de estado, actualización de reportes, control de horas. Actividades que insumen tiempo valioso que podría dedicarse al desarrollo del proyecto. En el gráfico del análisis sistémico se representa como “tiempo del desarrollador fragmentado”.

La ley de Brooks (Brooks F, 1995) señala que cuando un proyecto de software se encuentra demorado, **asignar más personas lo retrasa aún más**. Esto es debido a que la nueva incorporación debe aprender sobre el contexto del proyecto, los requerimientos, las herramientas que se emplean y adaptarse a las dinámicas del equipo. Situación que insume tiempo de las personas que ya están ocupadas en el desarrollo.

También he encontrado empresas que contratan personas que no tienen la experiencia requerida para la tarea, y con la expectativa de que sean productivos en el corto plazo. La falta de reconocerlo conduce a compromisos poco realistas.

Mi recomendación en este caso: reconocer la naturaleza impredecible de los proyectos complejos. El foco debe estar en mejorar el contexto organizacional para cuidar a la productividad del equipo y reducir los impedimentos. Al final encontrarás más detalle sobre estas recomendaciones.

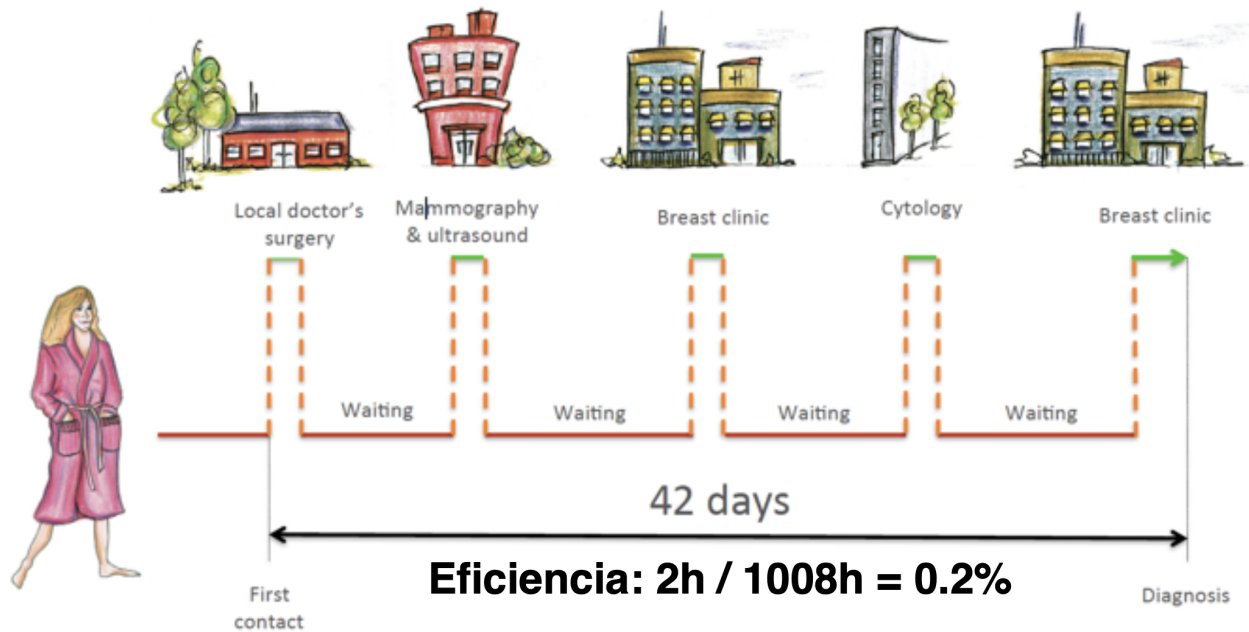
Causa 2: Fragmentación del tiempo de las personas

La cultura organizacional tradicional tiene su management diseñado para optimizar la eficiencia operacional de recursos. Reducir costos operativos y maximizar la utilización. Este concepto, dentro de la gestión de proyectos complejos implica decisiones que castigan el Time To Market.

La maximización de la ocupación de las personas lleva a asignarlas a varios proyectos al mismo tiempo, lo que implica que el proyecto debe esperar siempre que se necesite del aporte de una persona que está atendiendo otro proyecto. Este concepto se conoce como "eficiencia de recursos" (Modig 2011).

Por el contrario, con la "eficiencia de flujo" los proyectos están siempre avanzando, porque las personas están disponibles para atenderlos. Como dijo una vez mi colega Carlos Peix: **"siempre algo debe esperar: o espera el trabajo o espera la persona."**

Modig (Modig 2011) lo ejemplifica con un estudio realizado en un centro de salud. Cuando los médicos están todo el tiempo ocupados atendiendo a un paciente tras otro se los está aprovechando al máximo, sin embargo los pacientes tienen que esperar mucho tiempo su turno para recibir unos pocos minutos de atención, y luego volver a esperar. En el estudio realizado, se calcula la eficiencia de flujo, como el tiempo total de atención recibida (2 horas) por sobre el tiempo total transcurrido (42 días) la eficiencia da un alarmante: 0.2 %.



Existen consecuencias adicionales: las personas reconocen ser parte -principalmente- de un solo equipo. Así, los equipos dedicados que se vuelven cohesivos, desarrollan habilidades de colaboración que les permiten ser mucho más productivos que los pseudo-equipos compuestos por personas con asignación parcial. Asignar a los individuos distintos equipos o cambiar su estructura es perder el capital máspreciado.

El capital más importante de una organización son sus equipos cohesivos.

Hay un costo adicional asociado a trabajar en varios proyectos en simultáneo. Este costo se conoce como “cambio de contexto” (del inglés, context-switching). Es el desperdicio del tiempo invertido en recuperar el contexto del proyecto anterior: preparar las herramientas y recordar el estado anterior.

Mi recomendación simple pero poderosa, es: las personas deben estar asignadas a un solo equipo. Y un equipo debe trabajar en un único proyecto por vez.

Causa 3: Las interdependencias entre equipos.

Las organizaciones ágiles promueven equipos autónomos para que puedan avanzar en los proyectos sin depender de otros.

Cuando un proyecto que está a cargo de un equipo requiere participación de otro, normalmente éste no se encontrará disponible, sino ocupado con otras prioridades. Como resultado, el proyecto nuevamente cae en tiempos muertos y los equipos están impedidos para progresar.

Las organizaciones que están en su camino de ser más ágiles implementan equipos autónomos, pero muchas veces no completamente, y tarde o temprano terminan impedidos por estas dependencias. Esto sale a la luz cuando se requieren permisos de

seguridad, equipamiento de infraestructura o implementación en producción.

En una organización donde colaboré, un equipo contó en la Daily Scrum que tenía un impedimento: necesitaba que el área de Seguridad habilite un permiso de acceso. Éste equipo requiere que se “cree un ticket” con el pedido en el sistema de gestión Jira. Sólo la persona que tomó el ticket es quien da curso al pedido y esta persona trabajaba en horario nocturno, con lo cual, cada respuesta se demoraba al menos un día hábil. Luego de varios días sin novedades, volvió a consultar. La respuesta fue que “el ticket estaba mal redactado y por eso se lo cerró como inválido.” En este equipo implementar un desarrollo en producción puede implicar la gestión de 30 tickets con diferentes áreas involucradas. Este es el resultado de una organización fragmentada en silos y optimizada para reducir costos operativos.

Algo similar puede ocurrir entre equipos supuestamente colaborativos: dentro de la misma organización, el equipo A debía utilizar un servicio desarrollado y mantenido por el equipo B. Sin embargo el sistema presenta un defecto de calidad, que para el equipo B no era para nada relevante y no lo iban a solucionar. Sin embargo, para el equipo A resultaba bloqueante para un proyecto prioritario.

Lo que se espera como colaboración entre áreas termina siendo burocracia que busca optimizar otras variables distintas que las de reducir el Time to Market.

Mis recomendaciones para este caso son:

- Organizar equipos que sean realmente autónomos y que puedan avanzar sin depender de otros equipos.
- Aprender: los integrantes deben aprender a usar las herramientas y procesos para realizar las tareas por ellos mismos. O bien...
- Incorporar al equipo especialistas en las tareas que el equipo necesita pero no carece: En desarrollo de software ejemplos típicos son las prácticas de DevOps y DevSecOps.
- Mejoras conjuntas de procesos: los equipos que requieren interactuar deben hacer retrospectivas conjuntas para diseñar mejoras en la colaboración. En una retrospectiva del caso mencionado, el equipo asignó a una persona con prioridad exclusiva en atender los pedidos evitando así los tiempos muertos. La solución fue efectiva para el caso, pero tal vez no sea escalable para toda la organización.
- En desarrollo de software, automatizar los procesos rutinarios: pruebas de regresión, paso de ambientes y puesta en producción. No solo porque acelera el trabajo sino que también elimina las dependencias con otras áreas, sin que éstas pierdan el control del proceso..

Un ejemplo personal: cuando trabajaba como desarrollador de software y necesitaba cambios en un componente administrado por otro equipo, podía descargar el código fuente, modificarlo siguiendo los estándares definidos y enviarles el código nuevamente para su implementación. Mi código debía pasar las pruebas de calidad y la revisión de

una persona de ese equipo. Pero realmente me permitía avanzar en el proyecto, evitando tiempos muertos. Esta es una práctica habitual en desarrollo de software, conocida como “Collective Code Ownership” y es realmente efectiva.

Las consecuencias sistémicas.

Aquí es donde se viene lo interesante: cuando conjugamos estas causas aisladas y hacemos el análisis sistémico encontramos que deriva en una situación muy compleja.

Por una parte, el equipo no puede avanzar en un proyecto que está “en espera” por un impedimento que está fuera de su área de control. Y por otro lado, ciertos proyectos que se demoran pueden perder relevancia y hasta quedar obsoletos, mientras que nuevas oportunidades aparecen y los stakeholders no pueden permitirse esperar más para comenzar a trabajar en ellas.

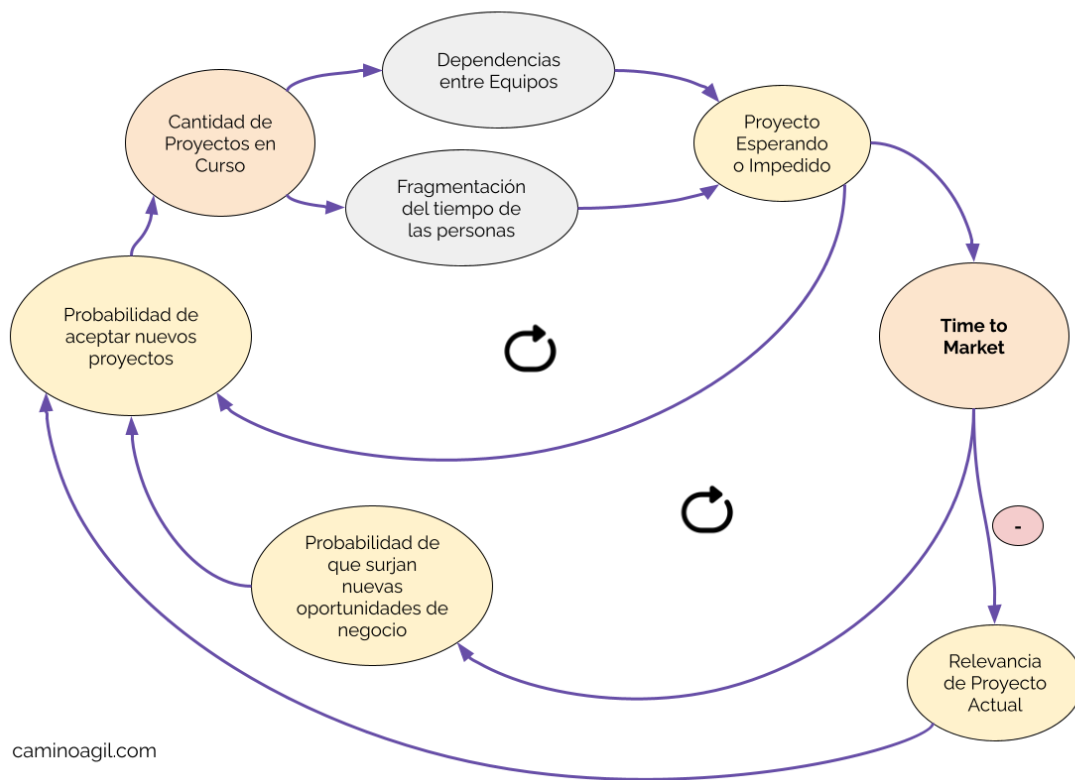
Las personas comienzan trabajando asignados a 2 proyectos y luego, se encuentran “haciendo contribuciones” (porque ya no pueden decir que están dedicados) en 10 proyectos a la vez.

La presión por comenzar nuevos proyectos es mayor que el esfuerzo que se requiere para destrabar los proyectos en curso.

Lamentablemente pocos son los proyectos que se cancelan, debido a que hay compromisos que cumplir y a la “falacia del costo hundido”: como ya hemos invertido mucho tiempo y dinero, la tendencia es pensar que conviene seguir invirtiendo para evitar así considerar la inversión como una pérdida absoluta.

La situación deriva en que las personas deben distribuir su tiempo en cada vez más proyectos (fragmentación del tiempo), con la falta de foco, el costo asociado al “cambio de contexto” y los tiempos muertos de los proyectos.

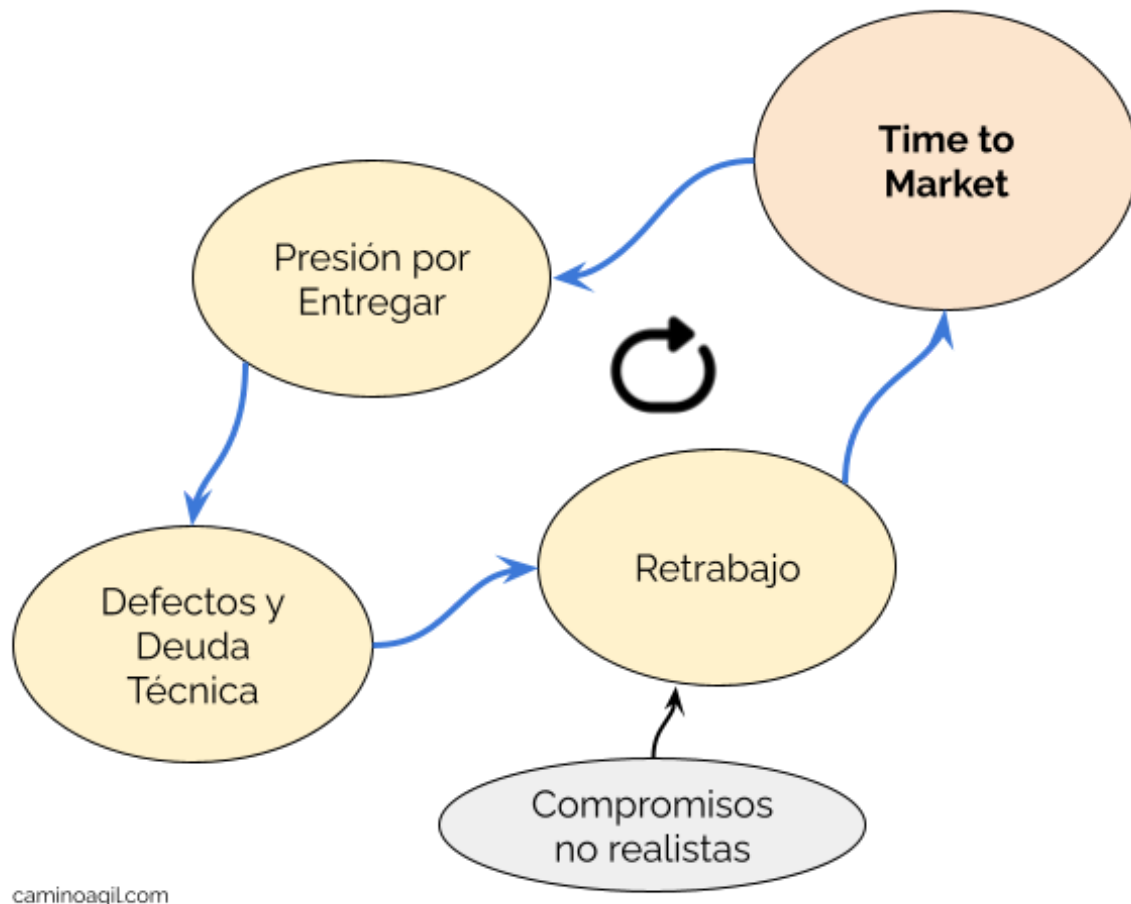
Se suma además el costo secundario de gestionar múltiples proyectos. Se requieren más Project Managers, más coordinación, más gestión, más reuniones de avance. Paradójicamente, las personas con mayor expertise y que más podrían aportar al avance real de los proyectos son quienes más ocupan su día de reunión en reunión.



Esto no termina aquí. La presión por entregar influye negativamente en la calidad, lo que a su vez genera consecuencias que impactan en el Time to Market de otros proyectos: el tiempo es usado en la gestión de reclamos, retrabajo para solucionar los problemas, mantenimientos y mayor esfuerzo para hacer cambios sobre un producto mal construido.

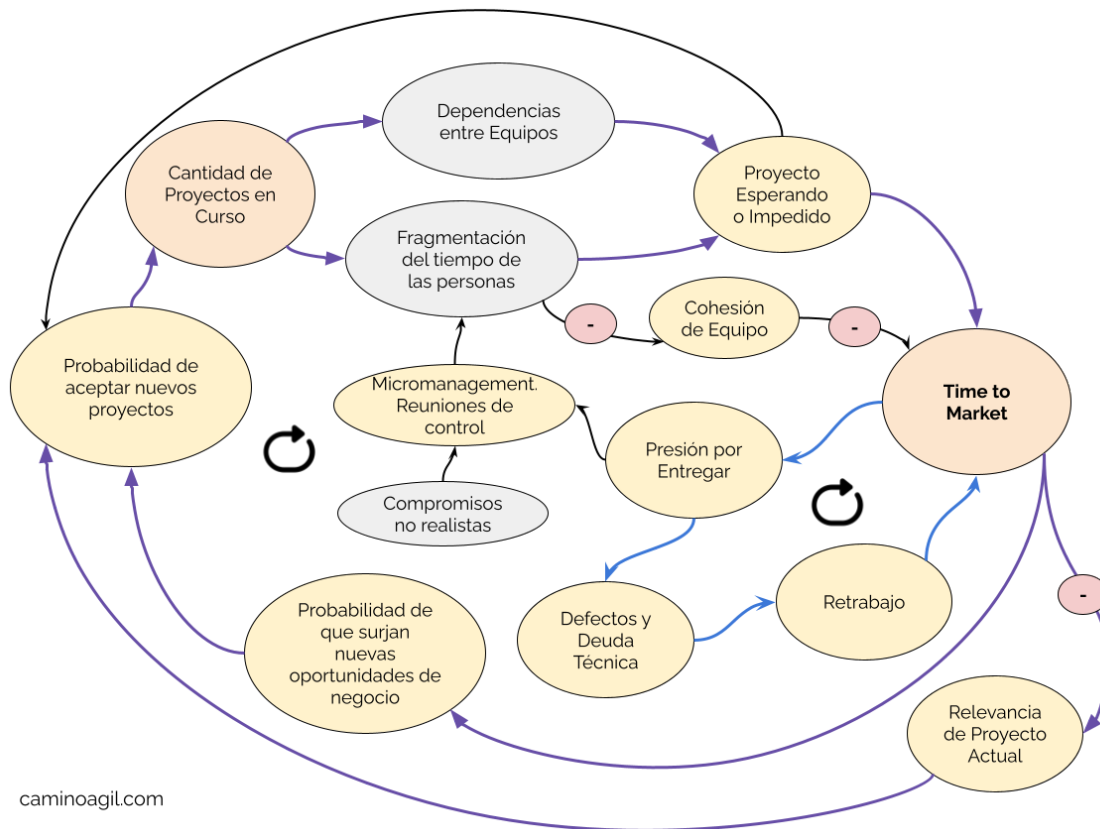
La “deuda técnica” y los defectos son grandes ocasionantes de demoras. En [este post](#), explico todos los motivos por lo que los defectos son tan nocivos para la productividad.

Como se puede ver en el siguiente gráfico, aquí también se genera otro ciclo vicioso.



Así, los nuevos proyectos quedarán bloqueados cada vez más rápido. Los proyectos pasan más tiempo en “espera” que teniendo un progreso real, a pesar de que todas las personas se encuentren realmente ocupadas trabajando en su máxima capacidad o incluso estresadas con horas extras. Los tiempos muertos y los cuellos de botellas hacen que la organización se estanque.

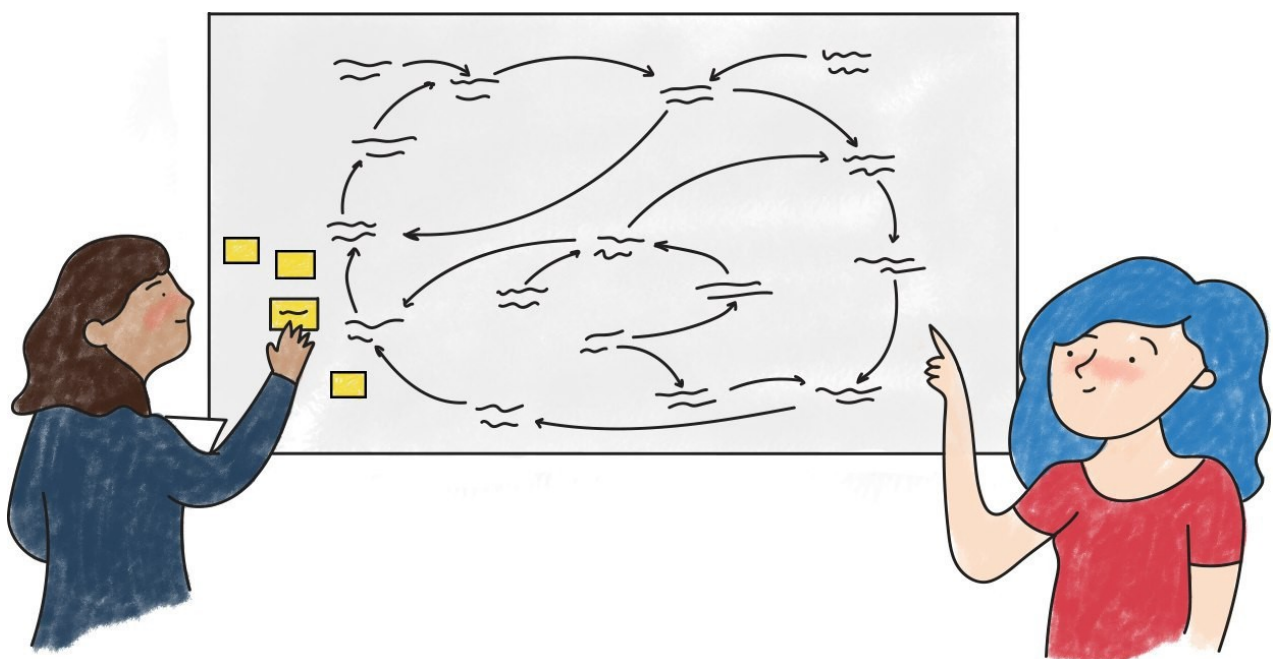
Entonces, el análisis sistémico se complejiza así:



La solución en cuatro pasos: también sistémica

Las recomendaciones que describí para cada caso son necesarias pero no suficientes. Se requiere una estrategia de liderazgo de la organización con una mirada sistémica. Los equipos por sí solos no pueden resolver el problema. Los siguientes cuatro pasos pueden llevar a la organización a un mejor contexto.

1) El primer paso es identificar esta situación para hacerla visible. El análisis sistémico nos sirve para esto. Si no vemos el problema no podremos actuar sobre él.



2) El liderazgo de la organización debe promover objetivos e implementar métricas que persigan la optimización global por sobre la optimización local.

- La optimización local se refiere a la eficiencia recursos y la productividad de lo que produce cada equipo (o peor aún, cada individuo). Métricas relacionadas son: horas trabajadas, story points, cantidad de software producido, nivel de ocupación.
- La optimización global se refiere a lo que la organización logra entregar al cliente. Algunas métricas relacionadas son: cantidad de proyectos completados, tiempo total de entrega al cliente, y métricas que reflejan la satisfacción del usuario o del cliente y los resultados del negocio logrados.

Como vimos en el ejemplo de Modig:

La optimización del trabajo de un equipo, puede resultar en una sub-optimización de los resultados de la empresa

3) Lograr flujo en la organización.

La estrategia de liderazgo debe priorizar que los proyectos fluyan, minimizando los tiempos de espera. Siendo esto más importante que maximizar la ocupación de las personas.

Henri Kniberg lo llama “escapar a la trampa de la utilización de recursos” y lo demuestra con una simple metáfora en este video:

Una de las frases famosas de Kanban es “**stop starting, start finishing**” (“dejar de comenzar, comenzar a terminar”). Para esto se propone como práctica central limitar la cantidad de trabajo en curso para gestionar la capacidad limitada y preservar el flujo

Un dicho popular dice: “si estás en un pozo, deja de cavar”. En una organización que se encuentra con demoras en los proyectos, lo mejor es dejar de asumir nuevos compromisos para dar lugar a que los proyectos en curso se completen.

Al limitar el trabajo en curso, los impedimentos se hacen visibles y eso es el primer paso para resolverlos. Sin este límite es más fácil trabajar en “lo que se puede”, en lugar de “en lo que realmente se debe” y los problemas se esconden dentro de la alta ocupación.

En Lean este concepto se ilustra con una metáfora, en la que el agua (la cantidad de trabajo) tapa los problemas.

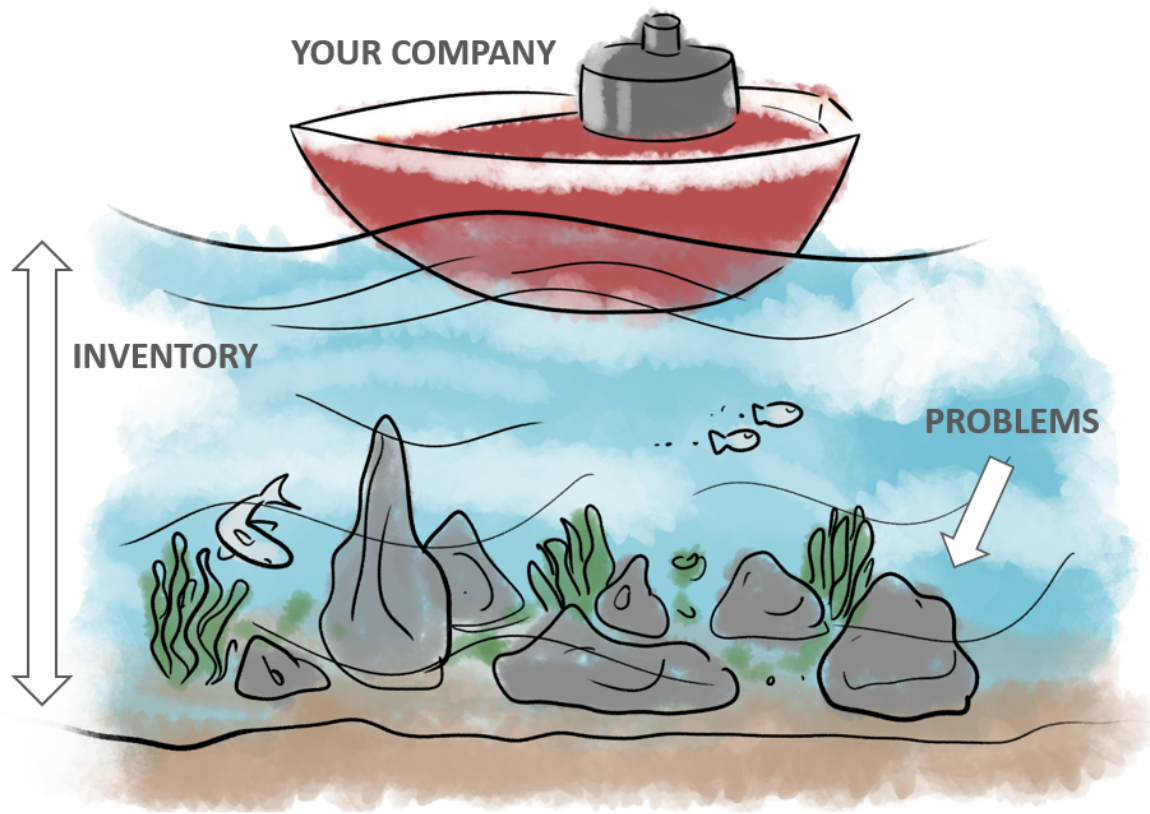


Imagen de: <https://medium.com/drill/lean-production-the-method-that-made-toyota-the-most-valuable-car-brand-in-the-world-13279db0b2>

Esta regla nos lleva a priorizar a conciencia los proyectos que más valor tienen para el negocio y ayuda a que los equipos mantengan foco en completarlos.

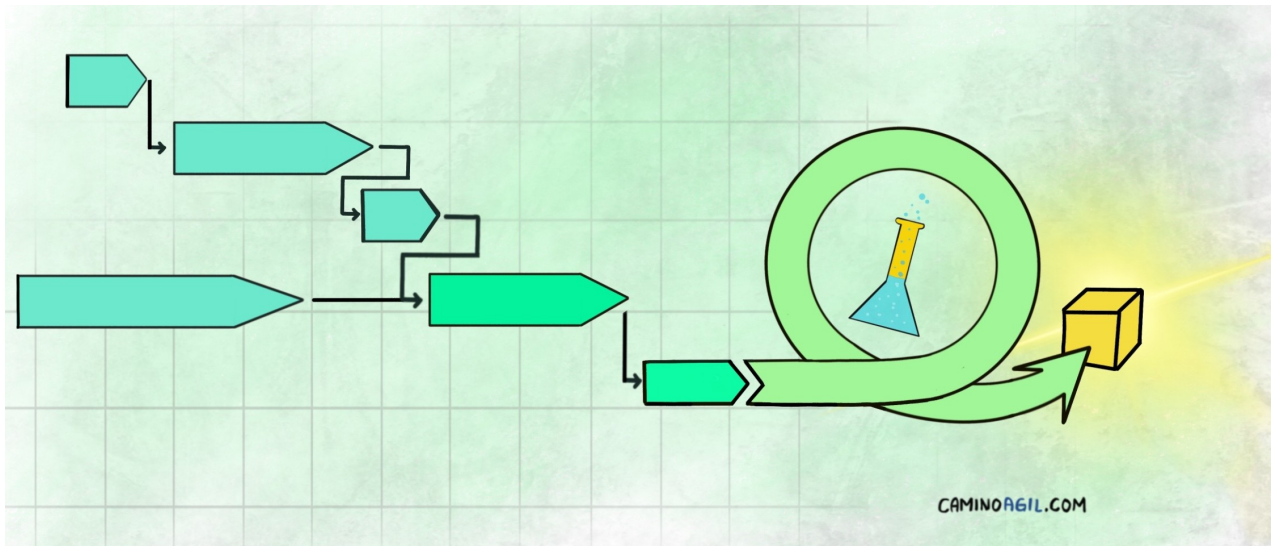
4) Buscar el punto óptimo entre trabajo en curso y tiempo de entrega.

Una vez logrado el flujo de trabajo, estaremos en condiciones de ajustar la cantidad de proyectos en curso. El objetivo será buscar cuántos proyectos puede gestionar la organización sin perjudicar los tiempos de entrega.

Sin tiempos de espera, los proyectos se completan tanto más rápido que será más fácil evitar que los proyectos se acumulen.

Estos cuatro pasos, no son simples de implementar, pero al hacerlo estaremos generando una transformación importante en la organización, llevándola a una forma más lean y agile de gestionar el trabajo.

Tal vez un paso más pequeño para comenzar, puedes utilizar estas **10 estrategias para agilizar cualquier proyecto.**



¡Muchas gracias por leer hasta aquí y muchos éxitos!

Referencias:

Brooks F, 1995. The Mythical Man-Month, Frederick P. Brooks Jr.

Modig, 2011. This is Lean, Resolving Efficiency Paradox. Niklas Modig.

<https://www.amazon.com/This-Lean-Resolving-Efficiency-Paradox-ebook/dp/B00JZZS7Q0>.

Senge 1990. Peter Senge. The Fifth Discipline, 1990

System Thinking. <https://less.works/less/principles/systems-thinking.html>

10 estrategias para agilizar cualquier proyecto

<http://www.caminoagil.com/2019/11/27/como-agilizar-cualquier-proyecto-10-estrategias-clave/>

Los bugs en Scrum

<http://www.caminoagil.com/2013/01/02/los-bugs-en-scrum/>

Imagen de: <https://medium.com/drill/lean-production-the-method-that-made-toyota-the-most-valuable-car-brand-in-the-world-13279db0b224>

Reconocimientos y Agradecimientos:

Ilustraciones de Ingrith Rojas y Damián Buonamico.

Gracias Nicolás Falcioni por las revisiones.

