

Plan de desarrollo de Proyecto

Resumen

El siguiente plan refleja las principales sugerencias en torno a la implementación del proyecto Tienda. (versión digital aquí¹)

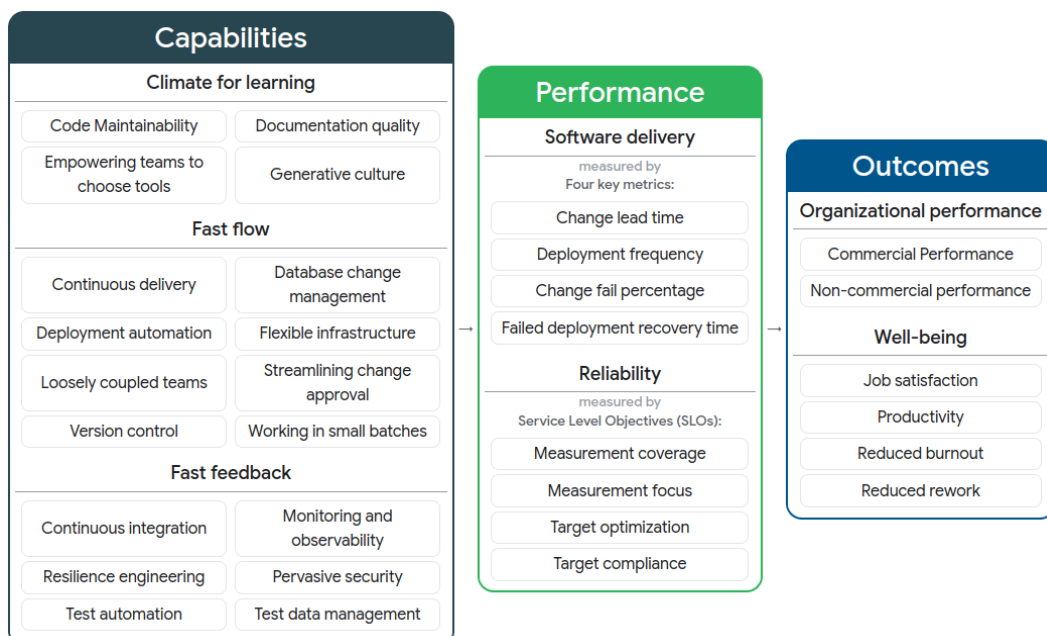
En pocos puntos se presenta una guía con las principales propuestas para un desarrollo ágil y práctico.

Capacidades y buenas prácticas

DORA - DevOps Research & Assessment

Es relevante utilizar o seguir un modelo de trabajo validado. Para el desarrollo de software, no tiene sentido “reinventar la rueda”, si ya existe un modelo sumamente analizado, que identifica los elementos y procesos necesarios para un desarrollo de software exitoso. Esencialmente del modelo DORA señala²:

- Las capacidades o buenas prácticas (*Capabilites*) **es un predictor de desempeño** de la velocidad y confiabilidad del Software (*Performance*).
- La velocidad y confiabilidad del Software (*Performance*) **es un predictor del desempeño** de la empresa (*Outcomes*)



¹ https://docs.google.com/document/d/1Vx_HkcrNwaKL-atwG7f2dDeNJVec0OhPLwEtcKH3kC8/edit

² <https://dora.dev/research/?view=detail>

El estudio DORA³ encontró una serie de características en común en las empresas líderes de desarrollo de software. Este reporte, patrocinado por Google, analizó más de 2000 empresas y 30.000 usuarios para analizar los patrones comunes de las empresas de alto desempeño.

Las empresas exitosas en la construcción de sus productos de tecnología, —según el reporte DORA—, son aquellas que aportan valor a los usuarios finales a través de mejoras frecuentes y estables, lo que permite alcanzar el cumplimiento de sus propios objetivos.

Este “alto rendimiento” según el estudio, se obtiene principalmente por la velocidad y estabilidad del desarrollo de software. A medida que los equipos “liberan” nuevas funcionalidades frecuentemente, y con bajas tasas de error, el valor real a los usuarios es tremendo. Y dado que despliegan nuevas características a cada rato, los cambios del código entre una y otra son más pequeños, y, por tanto, más fáciles de corregir en caso de errores.

Bueno, pero para lograr iterar rápidamente y de forma estable, el estudio DORA encontró una serie de capacidades y buenas prácticas que tenían todas las empresas de alto desempeño. Si bien dicho reporte recoge 29 capacidades⁴, aquí sólo se propone la búsqueda de 11 de ellas, que son más factibles de implementar en el contexto de la institución.

Gestión de Proyecto

En este punto se sugieren buenas prácticas en torno a la gestión del proyecto.

- **Visibilidad del proyecto.** Para que exista una comunicación efectiva entre los miembros del equipo, es esencial que exista una visibilidad completa del proyecto. ¿Qué tareas tengo pendiente? ¿Qué debo realizar primero? ¿Qué tareas están realizando mis compañeros?
- **Documentación de calidad.** Hoy en día la gestión del conocimiento debe ser parte integral del flujo de desarrollo para evitar la dependencia de personas y proveedores. Es decir, en las distintas etapas de construcción de los productos debe incorporarse el proceso de documentación técnica, tanto al inicio como al final del requerimiento. Buena documentación aumenta significativamente la eficiencia de los equipos, haciendo que “pierdan” menos tiempo buscando información.
- **Limitar la carga de trabajo.** Se ha visto que cuando los equipos de desarrollo tienen “infinito” trabajo o tareas por delante, con plazos de entrega irreales, tienden a buscar “atajos” o resolver los problemas de forma rápida y provisoria. Esto genera código difícil de mantener, y, la mayoría de las veces, problemas futuros (deuda técnica). Adicionalmente la sobrecarga de tareas genera estrés y baja satisfacción laboral. Por el contrario, cuando se define un máximo de tareas por profesional, se obtiene mejor software.

³ <https://dora.dev/>

⁴ <https://dora.dev/capabilities/>

Gestión de Producto

Brevemente se mencionan los principales puntos respecto del levantamiento de información, la construcción de los requerimientos, y la recepción de los productos desarrollados.

- **Requerimientos pequeños.** Se debe descomponer una tarea en subtarear más pequeñas cuando sea posible. Esta división permite a los desarrolladores poder iterar en lotes más pequeños y desplegar al ambiente de producción antes. Por el contrario, cuando la tarea es muy grande, pueden pasar meses antes de pasar a producción, y después de obtener los comentarios de los usuarios, puede que ciertas funcionalidades no son necesarias y no entregan valor. Cuando existen funcionalidades pequeñas, es más fácil “corregir” el rumbo.
- **Feedback de usuarios.** Es clave que el equipo de proyecto tenga acceso a los comentarios y sugerencias de los usuarios lo antes posible. Ello permitirá resolver oportunamente los problemas. Para ello, es necesario incorporar un proceso de captura de información frecuente.
- **Visibilidad de los procesos de negocio.** Es sumamente importante que el equipo técnico tenga acceso a información detallada de los procesos de negocio, así como de la visión del negocio. Esta información provee de una mejor perspectiva para el diseño de las soluciones.

Desarrollo de Tienda

En términos generales, las buenas prácticas de desarrollo se resumen en las siguientes 5:

- **Políticas y estándares.** El proveedor y los equipos deben contar con una guía de desarrollo que establezca la forma de trabajo. Debe estar definido el flujo de desarrollo, la forma de medir, como se validará el código, qué pruebas construir, etc. Esta política permitirá aclarar y ordenar el desarrollo.
- **Integración continua.** Se debe dar el espacio (tiempo) para que los desarrolladores puedan construir los conjuntos de pruebas necesarios para probar el funcionamiento de las nuevas funcionalidades. Esta automatización de pruebas es clave para validar el óptimo funcionamiento de la Tienda, pero requiere un trabajo de mantención continuo. Luego, este tipo de tareas deberán considerarse siempre como importantes y necesarias, y no deben ser descartadas.
- **Despliegue continuo.** Implica que los cambios realizados en el repositorio de código puedan ser habilitados inmediatamente en el ambiente productivo a través de acciones automatizadas. El despliegue continuo permite la actualización de aplicaciones en horario hábil, sin afectar a los usuarios.
- **Arquitectura desacoplada.** Implica evitar construir módulos “parches” o reescribir el core de las aplicaciones. Por el contrario, se debe diseñar el conjunto de soluciones de forma modularizada, en cada componente es un servicio independiente.
- **Flexibilidad.** Otorgar al equipo la flexibilidad para que puedan seleccionar la forma de trabajo, lenguaje de programación y diseño de arquitectura.

KPIs y métricas

Para evaluar cómo es la velocidad y estabilidad del equipo de desarrollo, es recomendable medir las siguientes variables (recomendación del estudio DORA):

1. Tiempo de entrega de cambios (**Lead Time for Changes**). Mide el tiempo desde que se confirma un cambio de código hasta que está disponible en producción. Esta métrica refleja la velocidad de entrega y la eficiencia del proceso. Es crucial porque indica cuán rápidamente un equipo puede entregar nuevas funcionalidades, mejoras o correcciones a los usuarios finales.
2. Frecuencia de despliegue (**Deployment Frequency**). Indica cuán frecuentemente se despliegan cambios en producción. Esta métrica refleja la agilidad del equipo y su capacidad para adaptarse rápidamente. Es importante porque permite entregar valor a los usuarios de manera más frecuente y con menos retrasos.
3. Tiempo medio de recuperación (**Mean Time to Restore, MTTR**). Mide el tiempo que tarda en recuperarse de una interrupción del servicio o incidente en producción. Esta métrica evalúa la capacidad del equipo para responder y recuperarse rápidamente de fallos, minimizando el impacto en los usuarios. Es crucial porque indica la resiliencia del sistema y la capacidad del equipo para mantener la continuidad del servicio.
4. Tasa de falla en cambios (**Change Failure Rate**). Mide el porcentaje de cambios en producción que resultan en incidentes, errores o necesitan ser revertidos. Esta métrica destaca la estabilidad y calidad del software desplegado. Es importante porque una tasa de fallos baja indica una alta calidad del código y del proceso de pruebas, con menos errores que llegan a producción. Refleja la estabilidad y la confiabilidad del sistema de producción.

Posteriormente, en la medida de lo posible, se deben definir ciertos objetivos que se deseen alcanzar. El cumplimiento de ciertos objetivos para estas métricas eventualmente deberían hacerse exigibles al proveedor adjudicado.

Estrategia

Visión

Actualmente no existe una visión o misión del producto (y proyecto), y, por lo tanto, no están claros los objetivos de mediano y largo plazo. Es bueno definir una visión del producto Tienda, que si bien por el momento no está definida, podría ser la siguiente:

“Facilitar la compra y distribución de insumos médicos, promoviendo una completa visibilidad del flujo transaccional, tanto para compradores como proveedores”

Contexto institucional

Es bueno recordar algunas características del sistema público que “moldean” la forma de trabajo de los proyectos de tecnología. Con este contexto, es posible definir o buscar conscientemente ciertos lineamientos que permitan asegurar el funcionamiento futuro del proyecto. Estas características son:

- Ley de compras. Al finalizar el contrato actual con el proveedor adjudicado, habrá que levantar un nuevo proceso de contratación, que posiblemente implique un cambio de proveedor. Desde esa perspectiva, no se puede depender del proveedor del momento, y, por lo tanto, es crítico y clave que **el proyecto siempre mantenga una documentación de calidad**, así como controles de validación para las entregas de código.
- Mecanismo de contratación. Típicamente existen 2 mecanismos de contratación para el desarrollo de software: (1) suma alzada y (2) por horas. Para el contexto de la institución lo recomendable será siempre buscar una estrategia de contratación por horas, principalmente porque dicho mecanismo permite abordar el desarrollo con una mirada en la calidad de las entregas. Por el contrario, cuando el mecanismo es a suma alzada, no hay control en el diseño en el código, y esto típicamente conlleva una calidad cuestionable.
- Evitar cambios profundos seguidos. Es frecuente observar el cambio de dirección de ciertos proyectos cuando asume un nuevo director o existe un cambio de jefatura. A veces se solicita reemplazar y migrar la plataforma, o bien iniciar un camino totalmente distinto al definido. Entonces se pierde todo lo avanzado. Por lo mismo, se debe buscar un compromiso de largo plazo a nivel institucional para mantener y respaldar el proyecto como fue concebido, y evitar cambiar su origen.

Estrategia de desarrollo

La incorporación de las buenas prácticas y/o capacidades mencionadas previamente, ayudarán a que el desarrollo del software sea óptimo. Evidentemente que más adelante se podrá evaluar la incorporación de las restantes capacidades que predicen un mejor desempeño.

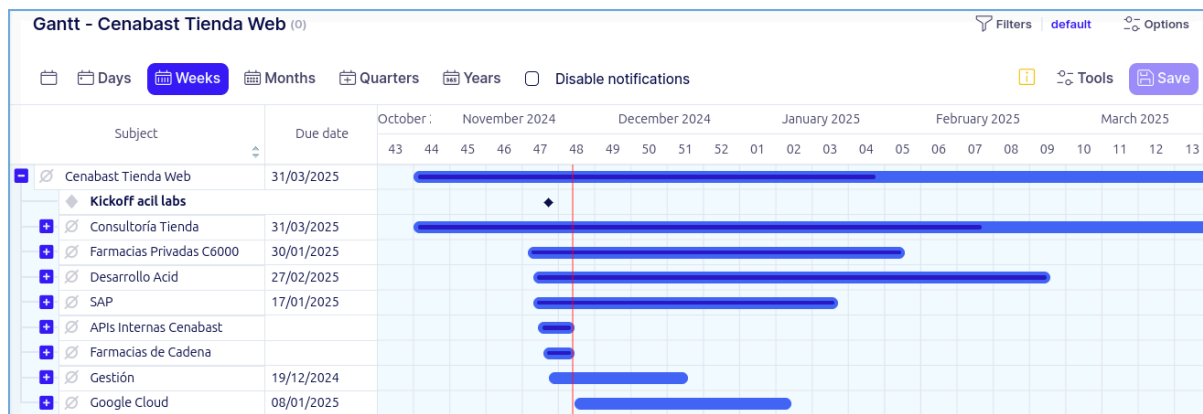
Corto plazo

Para el corto plazo, uno de los objetivos definidos de la fase actual es la habilitación del canal de Farmacias de Cadena. No están claros los objetivos siguientes, ya que a medida que el producto vaya siendo conocido por la organización, recién surgirán ideas o propuestas para mejorar la Tienda.

Sin embargo, y dada la complejidad de los procesos de Negocio, la construcción y diseño de la Tienda se está realizando con una separación de los canales de Tienda, pensando que en un futuro se puedan incorporar nuevos canales.

Así, se creó la siguiente carta Gantt⁵:

⁵ https://proyectos.cenabast.cl/projects/1/easy_gantt



Como muestra la imagen, el proyecto se dividió en 8 subproyectos, cada uno asociado a un equipo de trabajo distinto. Esto permitirá controlar y gestionar mejor cada una de las actividades y tareas.

En toda esta primera etapa (2-3 meses), es importantísimo que se pueda ir completando y construyendo la documentación respecto de los procesos de negocios y lógicas de funcionamiento. Esto permitirá al equipo de desarrollo tener mejor visibilidad y contexto del proyecto.

Respecto de la metodología de trabajo, se sugiere utilizar metodologías ágiles, lo que no será problema ya que el proveedor es experto en esta área.

Precauciones

- Cuidado con aceptar siempre las recomendaciones del proveedor. Se requiere un entendimiento y proceso de evaluación interno para tomar la mejor decisión para la empresa. Mucho problemas en proyecto de gobierno se arrastran por decisiones que “tomó el proveedor”.
- Evitar flujos de aprobación o autorización. Está validado que es mejor cuando los equipos de desarrollo tienen las facultades y permisos para subir cambios a los ambientes de producción sin necesidad de una autorización de un superior jerárquico. Siempre que las responsabilidades sean claras y exista un flujo de testing automatizado, será más eficiente que los desarrolladores puedan “empujar” cambios sin autorizaciones previas.