



Universidad de Buenos Aires
Facultad de Ciencias Económicas



Software Factories

AR Tema extractado del libro "**Análisis Funcional de Sistemas y Tecnologías de la Información**" de Aníbal M. Mazza Fraquelli - ISBN 978-987-26981-3-3

Presentación del Tema

El concepto de *Software Factory* (Fábrica de Software) refiere a un modelo organizativo y operativo orientado a la producción sistemática, estandarizada y escalable de soluciones de software. Desde la perspectiva de las Tecnologías de la Información (TI), no se trata simplemente de un equipo de desarrolladores, sino de una estructura integral que articula procesos, metodologías, herramientas, métricas, estándares de calidad y gobierno tecnológico con el objetivo de entregar productos de software de manera previsible, controlada y alineada con la estrategia organizacional.

Para estudiantes de licenciatura en administración, comprender la lógica de una Software Factory implica analizar cómo la producción de sistemas de información puede organizarse bajo principios similares a los de una fábrica industrial: división del trabajo, estandarización, control de calidad, automatización y mejora continua. Sin embargo, a diferencia de la manufactura tradicional, el "producto" es intangible y su principal activo es el conocimiento.

Desde la mirada de TI, la Software Factory constituye un componente clave dentro de la arquitectura de gobierno tecnológico, ya que impacta directamente en la capacidad de la organización para desarrollar, mantener e integrar sistemas de información críticos como ERP (*Enterprise Resource Planning* – Planificación de Recursos Empresariales), CRM (*Customer*

Relationship Management – Gestión de Relaciones con Clientes), BI (*Business Intelligence* – Inteligencia de Negocios) y plataformas digitales propias.

Desarrollo

1. Concepto y fundamentos de la Software Factory

Una Software Factory es un modelo estructurado para la producción de software basado en procesos repetibles, medibles y optimizables. Se apoya en principios de ingeniería de software y gestión de proyectos para lograr eficiencia, calidad y reducción de riesgos.

Entre sus fundamentos se destacan:

- Estandarización de procesos.
- Uso de metodologías formales.
- Definición clara de roles y responsabilidades.
- Gestión de la configuración y versiones.
- Métricas de desempeño.
- Automatización del ciclo de desarrollo.

Desde la administración, esto se traduce en una unidad organizativa que puede operar como:

1. Área interna de desarrollo.
2. Unidad compartida de servicios tecnológicos.
3. Proveedor externo tercerizado (outsourcing).

En todos los casos, la clave está en la gobernanza del servicio y en la definición clara de acuerdos de nivel de servicio (SLA – *Service Level Agreement*, Acuerdo de Nivel de Servicio).

2. Componentes estructurales

Una Software Factory eficiente integra diversos componentes:

a) Procesos

Se basan en marcos metodológicos como:

- Agile (Ágil): enfoque iterativo e incremental.
- Scrum: marco ágil con roles definidos (Product Owner, Scrum Master, Development Team).
- Waterfall (Cascada): modelo secuencial tradicional.
- DevOps (Development + Operations – Desarrollo + Operaciones): integración continua entre desarrollo y operaciones.

Desde la gestión administrativa, la elección del modelo impacta en costos, tiempos, gestión de riesgos y control presupuestario.

b) Ciclo de Vida del Desarrollo de Software

Se organiza bajo el SDLC (*Software Development Life Cycle* – Ciclo de Vida del Desarrollo de Software), que incluye:

1. Relevamiento de requerimientos.
2. Análisis funcional.
3. Diseño técnico.
4. Desarrollo.
5. Pruebas (testing).
6. Implementación.
7. Mantenimiento evolutivo y correctivo.

Para el administrador, cada fase implica recursos, presupuesto y métricas de control.

c) Herramientas Tecnológicas

Una Software Factory moderna utiliza:

- Repositorios de código (Git).
- Integración Continua (CI – *Continuous Integration*).
- Entrega Continua (CD – *Continuous Delivery*).

- Herramientas de seguimiento (JIRA, Azure DevOps, etc.).
- Automatización de pruebas.

La automatización reduce errores humanos y aumenta la trazabilidad.

d) Roles

Incluye perfiles como:

- Business Analyst (Analista de Negocio).
- Software Developer (Desarrollador).
- QA (*Quality Assurance* – Aseguramiento de Calidad).
- DevOps Engineer.
- Project Manager (PM).
- Product Owner.

Desde la perspectiva administrativa, estos roles deben estar alineados con la estructura organizacional y el presupuesto de TI.

3. Modelo de Madurez y Calidad

Una Software Factory profesional implementa modelos de calidad como:

- CMMI (*Capability Maturity Model Integration* – Modelo de Madurez de Capacidades Integrado).
- ISO/IEC 12207 (procesos de ciclo de vida del software).
- ITIL (*Information Technology Infrastructure Library* – Biblioteca de Infraestructura de TI).

Estos modelos permiten medir la madurez organizacional y garantizar estandarización.

4. Métricas y Control de Gestión

En términos administrativos, una Software Factory debe ser medible. Algunas métricas relevantes:

- Velocidad de desarrollo (en metodologías ágiles).

- Tasa de defectos.
- Lead Time (tiempo total desde requerimiento hasta entrega).
- Burn Down Chart (gráfico de avance).
- Costo por funcionalidad.

Estas métricas permiten análisis de eficiencia y retorno sobre la inversión (ROI – *Return on Investment*).

5. Ventajas Estratégicas

Desde la mirada de TI, una Software Factory permite:

- Escalabilidad.
- Predictibilidad presupuestaria.
- Reducción de dependencia de individuos.
- Mejora continua.
- Alineación estratégica.

Ejemplo: una empresa financiera que desarrolla internamente su plataforma digital puede acelerar el lanzamiento de nuevos productos si su Software Factory está madura y automatizada.

6. Riesgos y Desafíos

No todo modelo de Software Factory es exitoso. Existen riesgos:

- Excesiva burocratización.
- Desalineación con el negocio.
- Falta de gestión del cambio.
- Sobrecostos ocultos.
- Dependencia tecnológica.

Desde la administración, el error más frecuente es considerar la Software Factory como un centro de costos y no como un generador de valor estratégico.

7. Software Factory y Transformación Digital

En el contexto de transformación digital, la Software Factory es un habilitador clave.

Permite:

- Implementar automatización de procesos (RPA – *Robotic Process Automation*).
- Desarrollar plataformas omnicanal.
- Integrar sistemas heredados (*legacy systems*).
- Incorporar inteligencia artificial y analítica avanzada.

Sin una Software Factory eficiente, la transformación digital queda limitada a iniciativas aisladas.

8. Modelo Interno vs Outsourcing

Las organizaciones pueden optar por:

- Software Factory interna.
- Tercerización.
- Modelo híbrido.

Desde la administración estratégica, la decisión depende de:

- Competencias centrales.
 - Costos.
 - Seguridad de la información.
 - Regulación.
 - Tiempo de salida al mercado (*Time to Market*).
-

9. Impacto en los Sistemas de Información

Una Software Factory impacta directamente en:

- Arquitectura empresarial.
- Integración de datos.

- Gobernanza de datos.
- Seguridad informática.
- Continuidad operativa.

No es simplemente “programar”, sino construir infraestructura digital sostenible.

Conclusión

La Software Factory constituye un modelo organizacional fundamental para la producción sistemática y controlada de software dentro de las organizaciones modernas. Desde la perspectiva de las Tecnologías de la Información, su relevancia trasciende lo operativo y se inserta en el plano estratégico, dado que condiciona la capacidad de innovación, adaptación y competitividad de la empresa.

Para los estudiantes de administración, comprender la lógica de una Software Factory implica reconocer que el software no es un gasto técnico aislado, sino un activo estratégico cuya producción debe gestionarse con criterios de eficiencia, calidad, control y alineación con los objetivos corporativos.

Una Software Factory madura combina procesos estandarizados, métricas claras, automatización, modelos de calidad y una estructura organizativa definida. Cuando está correctamente implementada, se convierte en un motor de transformación digital, permitiendo escalar soluciones, reducir riesgos y generar valor sostenible.

En síntesis, la Software Factory no es un equipo de programación, sino un sistema organizacional de producción de sistemas de información alineado con la estrategia empresarial.

Preguntas de autoevaluación

1. ¿Cuáles son las principales diferencias entre un equipo de desarrollo tradicional y una Software Factory estructurada?
2. ¿Cómo impacta la elección de una metodología (Agile, Waterfall, DevOps) en la gestión administrativa del proyecto?
3. ¿Qué métricas permiten evaluar la eficiencia de una Software Factory?

4. ¿Cuáles son los riesgos de tercerizar una Software Factory y cómo pueden mitigarse desde la gobernanza de TI?
 5. ¿Por qué la Software Factory es un habilitador clave de la transformación digital en las organizaciones?
-
-

Material de Clases

Compilado por **Aníbal M. Mazza Fraquelli** Doctor de la Universidad de Buenos Aires para el uso de sus clases en la Facultad de Ciencias Económicas de la Universidad de Buenos Aires.

Contenidos de esta página

Los contenidos **aquí incluidos integran desarrollos y escritos propios del autor, así como materiales de terceros (documentos, textos, fragmentos, conceptos, imágenes, esquemas, definiciones u otros recursos)**, los cuales son utilizados a título ilustrativo, explicativo o formativo, respetando la normativa vigente en materia de derechos de autor y citando las fuentes cuando corresponde.

La selección, organización, adaptación pedagógica y contextualización de los contenidos constituye un trabajo original del autor, orientado a facilitar los procesos de enseñanza y aprendizaje.

Este material no persigue fines comerciales y su reproducción, total o parcial, queda limitada al ámbito educativo, debiendo preservarse siempre la mención de la autoría y las fuentes originales.

Autorización de uso

Se permite la reproducción, comunicación pública, distribución y utilización total o parcial de los contenidos de su material, en formato físico o digital, con fines exclusivamente educativos, académicos o de divulgación, siempre que se respete la integridad del contenido y se incluya la correspondiente referencia a la fuente y a la autoría.

Las ideas, opiniones e interpretaciones contenidas en este material corresponden exclusivamente al autor.

Queda expresamente excluido cualquier uso con fines comerciales.