Una Experiencia con Arquitecturas de Software para Roadmapping en Procesos de Transformación Digital

J. Andrés Díaz Pace[1] and Alejandro Bianchi[2]

[1] ISISTAN, UNICEN University, CONICET
Campus Universitario, Tandil, Bs. As., Argentina

[2] LIVEWARE IS, Buenos Aires, Argentina

adiaz@exa.unicen.edu.ar - alejandro.bianchi@liveware.com.ar

Resumen Entender y modernizar un sistema de software requiere un foco en el diseño arquitectural del sistema, pero también requiere que su implementación se lleve a cabo de una manera ordenada, a fin de mantener alineados aspectos de negocio y tecnología y asegurar el éxito del proceso de transformación en el tiempo. Un enfoque práctico y poco explorado en este sentido es la combinación de técnicas de roadmapping y arquitecturas de software. El presente trabajo reporta una experiencia de interacción universidad-empresa, en el marco de un proyecto PDTS, que propone un enfoque (framework) para abordar procesos de transformación digital guiados por arquitectura.

1. Introducción

En los últimos años han cobrado relevancia en el mercado del software la denominada *transformación digital* (TD) de sistemas [1,2], apalancada por tecnologías como BPM, Cloud Computing, Big Data, Internet of Things, movilidad, e Internet industrial. En este contexto, una organización debe entender qué significa para ella un proceso de TD, cuál es el impacto en su cultura organizacional, y cómo debería prepararse para los desafíos que implica. Por ejemplo, una TD no afecta de la misma manera a una gran corporación que a una PYME. La búsqueda de respuestas a estas preguntas debe estar en la agenda de todo dueño o CEO de compañía, a fin de trazar una hoja de ruta que indique cuándo y cómo llevar adelante un proceso de TD.

Para comprender los desafíos es importante clarificar la definición de TD. Según [2], transformación digital es el *uso de las tecnologías de la información (IT) para producir cambios disruptivos que produzcan mejoras radicales en la performance integral de una organización*. Notar que en esta definición la palabra clave es "transformación" y no "digital", de ahí que un proyecto de estas características debe ser liderado desde el negocio y no desde la Gerencia/Área de IT. Una transformación implica tanto aspectos técnicos como no técnicos. Independientemente del tamaño y/o tipo de negocio. La TD implica trabajar en transformar tres bloques concretos [2,1], a saber:

- **Experiencia del usuario**: Se refiere a ofrecer mejores experiencias a los usuarios a través de omnicanalidad, marketing directo, venta personalizada, y análisis del comportamiento del cliente, para entender sus necesidades y/o deseos.
- **Optimización de los procesos operativos**: Se refiere a innovar y automatizar los procesos de fabricación, de gestión, distribución y logística. También puede implicar proveer a los empleados de la posibilidad de trabajar cuando y donde lo necesiten, optimizar el diseño de productos a través de realidad virtual, integrar en tiempo real a los proveedores de componentes y/o de prestación de servicios, entre otras posibilidades.
- **Nuevos modelos de negocios:** Se refiere a aplicar la tecnología para crear nuevos negocios complementarios al core de la compañía, de manera de extender productos y servicios a nuevos mercados.

Adicionalmente, cabe mencionar que existen ciertas se deben desarrollar un conjunto de *capacidades organizacionales* que deben ser desarrolladas para ofrecer unaapalanquen una solución robusta que provea reales beneficios al negocio, tales como: integración de datos y procesos, alineamiento entre IT y el negocio, trabajo colaborativo entre IT y el negocio, y entrega continua de soluciones.

Un proceso de TD conlleva una dimensión temporal que implica partir del sistema o plataforma inicial (o as-is), y en base a un sistema o plataforma objetivo (o to-be), plantea una t*ransición gestionada* normalmente mediante una secuencia de sistemas intermedios, hasta llegar al sistema objetivo. Una técnica interesante para abordar dicha transición es el *roadmapping*, una técnica que fue originalmente desarrollada por Motorola en los 70 con miras a lograr una mejor alineación entre la tecnología y el desarrollo de productos, mediante una representación visual estructurada de la estrategia que se desea desarrollar [3]. Un *roadmap* (u hoja de ruta) identifica y selecciona alternativas estratégicas para lograr determinados objetivos, y comunica en forma resumida las decisiones claves. El roadmapping ha sido ampliamente adoptado por muchas organizaciones de diferentes sectores, ya que el concepto subyacente es muy flexible.

No obstante, un problema común cuando se aplica roadmapping a proyectos de IT es que muchas veces la información presentada está fragmentada, o no incluye una perspectiva técnica (de software) que permita vincular tecnologías con objetivos de negocio, por ejemplo en lo referido a modelos de software y procesos de desarrollo. Debido a esto, las hojas de ruta no siempre ayudan a los responsables del proyecto a planificar y monitorizar un proceso de TD. Para mitigar estos problemas, se propone integrar roadmapping con técnicas de arquitectura de software [4], buscando de desarrollar mejores hojas de ruta para proyectos de desarrollo centrado en arquitecturas. Esta propuesta se enmarca en un proyecto PDTS[3], desarrollado en conjunto por investigadores de UNICEN y UTN-FRCórdoba y la empresa Liveware IS como principal adoptante. Para el desarrollo de la propuesta se siguió un enfoque iterativo y basado en abstracción

[3] Proyecto de Desarrollo Tecnológico y Social. PDTS No. 217 "Un Método centrado en Arquitecturas de Software para Líneas de Producto con Soporte de Herramientas", financiado por MINCYT y CONICET.

a partir de casos de estudio provistos por Liveware IS. Como resultados técnicos de este proyecto se apunta a proveer guías, modelos y herramientas para llevar a cabo proyectos de software que involucren TD (en un horizonte de tiempo) y que deban atendercon aspectos atributos de calidad críticoas, de una manera más predecible, repetible e ingenieril.

El presente artículo constituye un reporte de experiencia del proyecto mencionado, y discute el enfoque de integración entre roadmapping y arquitecturas de software. El resto del artículo está organizado en 3 secciones. La Sección 2 describe brevemente los casos de estudio reales que motivaron nuestro enfoque. La Sección 3 introduce los conceptos básicos de roadmapping y arquitectura de software, para luego presentar el enfoque propuesto. Finalmente, la Sección 4 discute el estado actual del proyecto y perspectivas de trabajo futuro.

2. Experiencias Iniciales

La necesidad de contar con hojas de ruta que combinen objetivos de negocio, tecnologías, y modelos de diseño de software se originó a partir de una serie de proyectos de Liveware IS relacionados con modernización de sistemas legados y TD en dominios de radares misión crítica y financieros[4]. A continuación, se presentan dos de estos proyectos como casos de estudio, referidos como Proyecto A y Proyecto B, respectivamente.

Proyecto A. .En un dominio de misión crítica, la organización (cliente) requería la re-conversión de una solución de software legada para procesamiento de telemetría y gestión de alertas de equipo, con el objetivo de obtener una arquitectura de referencia [5] para una familia de productos específicos. Los principales desafíos se resumen en el Cuadro 1.

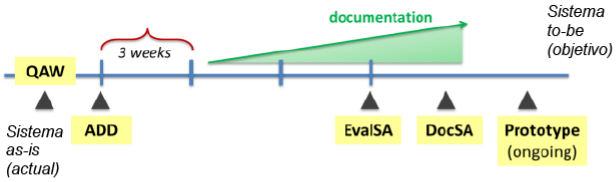
Desafío 1.A: Migración de un sistema legado con tecnología propietaria a un nuevo sistema con mayor facilidad de evolución y nuevos atributos de calidad (por ej., performance, modificabilidad de reglas de negocio).
Desafío 2.A: Horizonte de tiempo acotado (1 año), sin versiones intermedias entre el sistema as-is y el sistema to-be (sin necesidad de hoja de ruta).
Desafío 3.A: Definición de una arquitectura de referencia.
Desafío 4.A: Consolidación de prácticas de Ingeniería de Software en la organización.

Cuadro 1. Resumen de Proyecto A (modernización).

En este contexto, el equipo de Liveware propuso una estrategia basada en los métodos de arquitectura QAW, ADD y ATAM del Software Engineering Institute (SEI) [4] (ver Sección 3), a partir de un análisis del sistema actual (as-is), de la definición del sistema objetivo (to-be), y de la identificación de diferencias (gaps) entre el sistema as-is y el sistema to-be. La hoja de ruta utilizada aquí fue

[4] Por razones de confidencialidad, se omiten detalles de los casos de estudio, y se ha anonimizado u alterado algunas partes, aunque sin afectar aspectos relevantes de la problemática.

simple, ya que este proyecto no planteaba la necesidad de arquitecturas intermedias para alcanzar el sistema objetivo. No obstante, la estrategia sentó las bases para determinar (en este caso, mediante técnicas de reconstrucción) los activos disponibles en el sistema as-is y la factibilidad de reutilizarlos (con algunas modificaciones) en el sistema to-be. Este ejercicio representa un análisis básico de capacidades estructurado en función de escenarios de calidad. La Fig. 1 ilustra el modelo de ciclo de vida planteado, y también muestra uno de los artefactos (tabla con semáforo) utilizado para identificar gaps en el sistema actual como extensión del método QAW.



Escenarios de calidad y prioridades, respecto a grupos de funcionalidades existentes en el sistema as-si

Scenario	Priority	GUIs	Core	Repository	Drivers
C2	H	N/A	N/A		
P1	H				
U1	H				N/A
D1	H				N/A
C1	M			N/A	
C3	M				
M1	M				N/A
M2	M				
A2	M				N/A
DI1	M				
A1	L	N/A			N/A

Figura 1. Utilización de métodos de arquitectura en el ciclo de vida del Proyecto A, adaptados a un contexto de TD – con un roadmap simple.

La estrategia desarrollada proveyó un beneficio de visibilidad y convergencia de los intereses de los stakeholders respecto a la arquitectura, y permitió contar con indicadores para medir progreso en las actividades de diseño y codificación. Esta experiencia constituye un roadmap rudimentario.

Proyecto B. En un dominio financiero, la organización enfrentaba un problema de obsolescencia tecnológica de un core bancario y adicionalmente buscaba proveer una mayor automatización de procesos de negocio hacia sus clientes. Los principales desafíos se resumen en el Cuadro 2.

Desafío 1.B: Modernización de un sistema legado con tecnología propietaria (obsolescencia tecnológica) a un nuevo sistema con nuevos atributos de calidad (por ej, performance, modificabilidad, testeabilidad) y algunos productos de terceros.
Desafío 2.B: Horizonte de tiempo moderado (2 años y medio), con algunas versiones intermedias entre el sistema as-is y el sistema to-be
Desafío 3.B: Definición de una arquitectura de referencia común para productos customizados en varios clientes.
Desafío 4.B: Utilización del proyecto de modernización como parte de la estrategia de marketing estratégico de la organización.

Cuadro 2. Resumen de Proyecto B (transformación digital).

En este contexto, se aprovechó el desarrollo metodológico anterior sobre una arquitectura de referencia, pero instanciado (es decir, planificado) sobre un horizonte de tiempo anual que necesariamente precisaba de arquitecturas intermedias para lograr el objetivo. Una característica distintiva de este proyecto fue el hecho que la arquitectura de referencia debía integrar productos de terceros (por ej., un motor de BPM) como activos corporativos de la organización. Respecto a las arquitecturas intermedias y las capacidades a desarrollar en el tiempo, estas debían apuntar a reducir paulatinamente la *deuda técnica* del sistema [6]. Claramente, este proyecto necesitó de un roadmap más elaborado que el proyecto anterior. La Fig. 2 esquematiza el rol del roadmap como enlace entre la arquitectura actual y la arquitectura objetivo, en función de un alineamiento constante con los aspectos funcionales y del negocio. En este sentido, el roadmapping ayuda a una evolución gestionada de la TD [7].

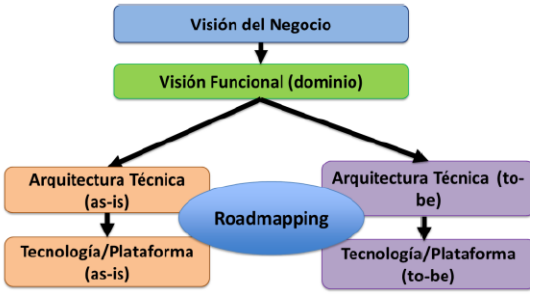


Figura 2. Rol del roadmapping como enlace entre los sistemas as-is y to-be, y el alineamiento respecto a la visión de negocio.

Este caso de estudio permitió a Liveware IS una mejor definición del artefacto de hoja de ruta, relacionando en un solo diagrama información de producto (software) y capacidades técnicas (modelo de ciclo de vida, entrenamiento) y de negocio (por ej., estrategia comercial). Como beneficios obtenidos en el Proyecto B, puede mencionarse que el diseño arquitectural fue el conductor para gestionar los cambios, resolver conflictos, y mantener a los equipos focalizados en objetivos concretos. En forma similar al Proyecto A, la arquitectura sirvió para identificar las partes del sistema a mejorar (aunque no fue necesaria una reconstrucción), la priorización de estas partes, y las actividades a desarrollar en el equipo para lograr dichas capacidades. Una versión de alto nivel del roadmap resultante se da en la Fig. 4.

3. Enfoque propuesto

El enfoque inicial de integración de arquitectura de software con roadmapping fue sufriendo ajustes, en base a experiencias prácticas como las anteriores, hasta converger a su versión actual, que se discute en esta sección. Antes de describir el enfoque, se introduce brevemente los conceptos básicos de roadmapping (tradicional) y de técnicas de arquitectura del SEI, como pilares del trabajo.

3.1. Roadmapping

Bob Galvin, ex-CEO de Motorola, dió la siguiente definición de roadmap: "*Una hoja de ruta es una mirada proyectada hacia el futuro sobre un campo de investigación elegido (producto, servicio, etc.), a partir del conocimiento colectivo y la imaginación de los innovadores más brillantes en ese campo*". Esta definición hace hincapié en la importancia que el conocimiento y la experiencia desempeñan en el proceso, la naturaleza prospectiva del enfoque, y su flexibilidad para adaptarlo a nuevas realidades. Un roadmap estándar es un gráfico organizado con base en una dimensión temporal, que incluye múltiples perspectivas o capas que abarcan aspectos interrelacionados de negocios y tecnológicos [3]. Un roadmap muestra cómo la tecnología puede ser alineada al desarrollo de productos y servicios, a la estrategia de negocios y a las oportunidades de mercado. El formato básico para un roadmap posee tres perspectivas, a saber: Mercados, Productos y Servicios, y Tecnologías, según se esquematiza en la Fig. 3. Pueden agregarse perspectivas adicionales al diagrama, como es el caso de recursos humanos.

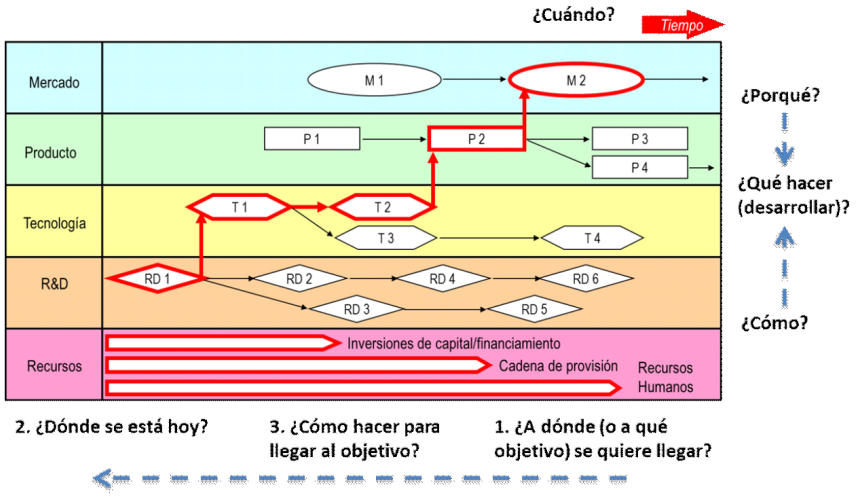


Figura 3. Diagrama tradicional de roadmapping – con preguntas típicas.

Mediante su dimensión temporal, un roadmap permite a los stakeholders determinar: i) el estado actual del sistema (as-is), ii) el estado objetivo o futuro (to-be) del sistema, y iii) las acciones a realizar en las diferentes perspectivas para tender hacia dicho objetivo. Por un lado, este aspecto está representado por las tres preguntas debajo del diagrama en Fig. 3. Notar el orden de progresión de las respuestas a dichas preguntas procede en sentido inverso al flujo del tiempo, es decir: se parte del objetivo, en base a eso se analiza el estado actual, y finalmente se plantean acciones para abordar la brecha entre esos dos estados. Por otro lado, el aspecto de planeamiento estratégico está representado por las preguntas

(verticales) a la derecha del diagrama en Fig. 3, que alinean las acciones a realizar con su basamento tecnológico (o de recursos) y con los objetivos de negocio.

Una ventaja de un roadmap es la comunicación entre las diferentes áreas que resultan afectadas por una determinada estrategia, ya sea para nuevos productos o para un cambio radical desde el punto de vista del negocio, involucrando perspectivas funcionales, organizaciones y tecnológicas. En nuestra experiencia con proyectos de software, la riqueza del proceso de elaboración de la hoja de ruta radica en la integración de los diferentes stakeholders involucrados facilitando el intercambio de ideas, la identificación temprana de riesgos y la búsqueda de consensos. Una vez que se ha elaborado un roadmap, este se comunica formalmente, actuando como punto de referencia para el desarrollo, seguimiento y ajuste de las diferentes acciones que lo componen.

En lo referido a productos y modelos de ciclo de vida de software, el diagrama de la Fig. 4 muestra un roadmap (simplificado) aplicado por Liveware IS en el Proyecto B. Notar que el horizonte de tiempo está clasificado en acciones a corto plazo, mediano plazo y largo plazo. Notar también la inclusión de una perspectiva para el *producto de software* en sí (con acciones y dependencias entre ellas), y de otra perspectiva de *capacidades técnicas* a potenciar en la organización, a fin de dar sustento a las acciones de la perspectiva de producto. Las capacidades técnicas se refieren a buenas prácticas de desarrollo de software, pero también a la elección de un *modelo de ciclo de vida guiado por arquitectura*.

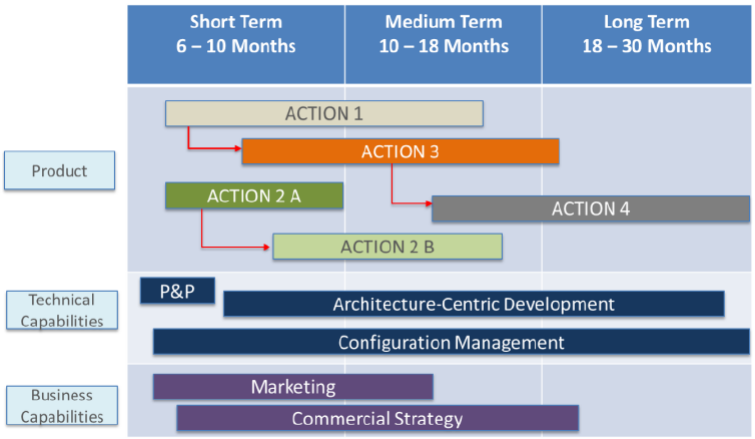


Figura 4. Ejemplo de roadmapping aplicado a un proyecto de TD (de software).

3.2. Métodos de Arquitectura

Esencialmente, la arquitectura de software [4] es un modelo que sintetiza las necesidades de los stakeholders de un sistema en un *conjunto de decisiones técnicas* orientadas a satisfacer los atributos de calidad críticos del sistema (por ej., performance, disponibilidad, seguridad, interoperabilidad, entre otros).

Estos atributos de calidad normalmente se expresan mediante escenarios de calidad, que son definidos y priorizados por los stakeholders. En este sentido, se espera que una solución de arquitectura cumpla con los objetivos del negocio, y adicionalmente, que brinde *prescripciones* para la implementación del sistema.

El enfoque propuesta está basado en un ciclo de vida guiado por arquitectura [8]. Un ciclo guiado por arquitectura es aquel que utiliza el diseño arquitectural como punto focal para la ejecución de las actividades de producción y evolución de software. De esta forma, es posible articular distintos métodos para cubrir desde el análisis de objetivos del negocio y requerimientos clave, hasta el inicio del proceso de desarrollo en sí. El tándem de métodos de referencia, aplicado por Liveware IS en los proyectos mencionados, se describe en la Tabla 3 junto con una breve descripción.

TAREAS	METODO	MODALIDAD	DETALLES
Workshop de Objetivos de negocios	PALM	Workshop estructurado	Participativo e involucra a los stakeholders relevantes y está basado en escenarios
Workshop de atributos de calidad	QAW	Workshop estructurado	Participativo e involucra a los stakeholders relevantes basado en escenarios y en requerimientos de arquitectura
Diseño de la Arquitectura de referencia	ADD	Ciclo iterativo de diseño guiado por los atributos de calidad	
Documentación de la Arquitectura candidata	Views & Beyond	Esquema de documentación basado en vistas y perspectivas.	LIVEWARE utiliza templates basados en EA y soporte de documentos Word y/o Excel. Puede incluir Wikis
Evaluación de arquitectura candidata	ATAM (o ATAM-lite)	Método formal de evaluación de arquitecturas basado en escenarios.	Sesiones de revisión con participación de los stakeholders relevantes y los arquitectos responsables del diseño.
Asegurar conformidad entre arquitectura e implementación	Workshop de Familiarización	Workshop estructurado	LIVEWARE utiliza templates basados en EA y soporte de documentos Word y/o Excel.

Cuadro 3. Resumen del métodos de arquitectura de software (SEI).

Notar que esta articulación de métodos es una *recomendación*, pero que puede sufrir modificaciones dependiendo de las características propias del proyecto en cuestión. Esto es, puede no incluirse un método particular, o puede que algún método se aplique con ajustes. Un ejemplo del primer caso es el método de diseño ADD, ya que en algunos casos no hace falta un diseño de la solución, porque la arquitectura ya fue definida por la organización. Un ejemplo del segundo caso es el método ATAM, que puede variar de acuerdo a la disponibilidad de los stakeholders o al nivel de documentación del sistema bajo análisis.

3.3. TD centrada en Arquitectura

Una reflexión que surge de las sub-secciones anteriores es que la arquitectura de software puede verse como un lenguaje para conectar e interpretar aspectos de negocio y tecnología, y su alineamiento en el tiempo. Por esta razón, resulta

natural pensar la arquitectura de software como un nexo clave para un roadmapping exitoso. Arquitectura y roadmapping se complementan en el sentido que este último destaca la dimensión temporal de una TD, aspecto que no siempre está explícito en los métodos de arquitectura tradicionales [9]. En este contexto, un proceso de TD puede organizarse en términos de un roadmap que incluye métodos y artefactos de arquitectura, según se presenta en la Fig. 5.

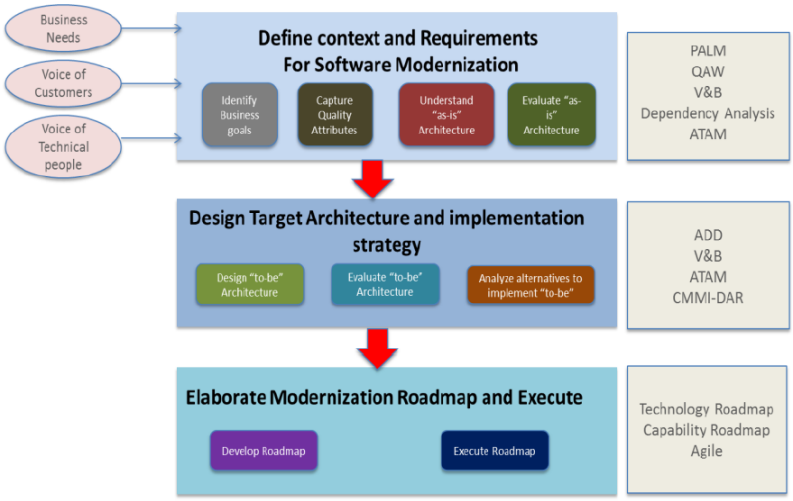


Figura 5. Enfoque evolutivo para TD con base en roadmapping y arquitectura.

El enfoque está estructurado en 3 fases (indicadas por las capas horizontales), cada una de ellas compuesta por distintas actividades. Estas fases se corresponden con las tres preguntas temporales (desplegadas abajo y en forma horizontal en Fig. 3) de un roadmapping. Los bloques a la izquierda del diagrama indican métodos o técnicas recomendadas como soporte de las actividades de cada fase. El punto de partida es el análisis de las necesidades del negocio, particularmente la "voz del cliente", pero también las necesidades del personal técnico.

La primera fase (*Define Context and Requirements for Software Modernization*) involucra: i) la identificación de los objetivos del negocio (por ej., mediante PALM), ii) la captura de atributos de calidad y su tratamiento con escenarios (por ej., mediante QAW), iii) el entendimiento de la arquitectura as-is, y iv) la evaluación de la arquitectura as-is. En nuestra experiencia, la arquitectura as-is suele ser mucho más concreta que la arquitectura to-be. Sin embargo, la arquitectura as-is suele estar pobremente documentada o poseer documentación desactualizada, lo cual suele requerir un esfuerzo importante de relevamiento e incluso de reconstrucción del diseño a partir del código fuente (y de entrevistas con el personal). Los métodos sugeridos para las actividades iii) y iv) son V&B para documentación, algún tipo de herramienta de análisis de dependencias (especialmente si se decide aplicar reconstrucción), y ATAM para la evaluación de la arquitectura as-is.

La segunda fase (*Diseñar Arquitectura Objetivo y Estrategia de Implementación*) involucra: i) diseñar la arquitectura to-be (por ej., mediante ADD), ii) evaluar la arquitectura to-be (por ej., mediante ATAM), y iii) analizar alternativas de implementación para la arquitectura to-be. La arquitectura to-be muchas veces es solo una arquitectura nocional, propuesta por el negocio. Si bien no es necesario un alto nivel de detalle en esta arquitectura, es recomendable un nivel de definición suficiente que capture las principales vistas y decisiones de arquitectura, y que permita una modalidad de evaluación estilo ATAM. La última actividad requiere un análisis de "reusabilidad" entre los componentes pensados para la arquitectura to-be y los componentes de la arquitectura actual, y adicionalmente, una planificación de la estrategia de construcción (o reuso) en función del tiempo. Es aquí donde aparecen las arquitecturas intermedias, y puede aplicarse una estrategia de "valles y olas" para la planificación de capacidades del sistema [10]. Normalmente, en las primeras arquitecturas intermedias suelen coexistir componentes legados con componentes "nuevos" del sistema. Esta última actividad es exploratoria, y no define (todavía) una hoja de ruta. Por ejemplo, las dependencias entre las arquitecturas intermedias y sus componentes no necesariamente se determinan en esta fase, ya que requieren un análisis que abarque otras perspectivas.

La tercera fase (*Elaborar Hoja de Ruta y Ejecutar*) involucra: i) la elaboración del roadmap propiamente dicho, y ii) su ejecución por parte de la organización. El roadmap se construye en base a las alternativas relevadas en la fase anterior, pero también incluye las otras perspectivas. La organización (o sponsor del proyecto) puede realizar ajustes en la hoja de ruta atendiendo consideraciones de capacitación de personal, presupuesto, plazos de entrega pre-acordados, alcance funcional, o exposición al riesgo, entre otras. Finalmente, una vez formalizado el roadmap y acordado con los stakeholders, este comienza a ejecutarse. En cada arquitectura intermedia, se evalúa el estado de la ejecución, se determina si el sistema to-be es realista, y se realizan las correcciones de curso necesarias.

Actualmente, Liveware IS se encuentra aplicando el enfoque definido en Fig. 5 en otro proyecto del dominio financiero (denominado Proyecto C), que está inmerso en un proceso de TD, debido a las presiones de la competencia y a la necesidad de ampliar su base de clientes.

Proyecto C. El objetivo es desarrollar una arquitectura de referencia corporativa en la cual inscribir distintos sistemas tanto propios como de terceros. Este caso de estudio plantea un horizonte de tiempo a 4 años, donde deben desarrollarse arquitecturas para productos financieros con variaciones de acuerdo distintos países. Adicionalmente, este proyecto plantea un desafío adicional para la arquitectura, ya que posee una variedad de integraciones de las cuáles depende el éxito del sistema. En este contexto, el enfoque base descripto (Fig. 5) debió ser refinado para considerar dichas integraciones. Los desafíos se resumen en el Cuadro 4.

Las experiencias iniciales de Liveware IS con implementaciones de roadmapping y arquitectura de software han mostrado varios beneficios, a saber:

Desafío 1.C: Evolución de un sistema legado y distribuido en varios países con tecnología propietaria a un nuevo sistema unificado y basado en integraciones con productos de terceros.
Desafío 2.C: Horizonte de tiempo largo (4 años), con versiones intermedias entre el sistema as-is y el sistema to-be, y coordinación de despliegues por países
Desafío 3.C: Definición de una arquitectura corporativa, y derivación de arquitecturas de referencia para cada país.
Desafío 4.C: Utilización del proyecto de evolución para mejor el gobierno y la gestión de IT dentro de la organización.

Cuadro 4. Resumen de Proyecto C (transformación digital).

- las técnicas de roadmapping facilitan el desarrollo de planes (relativamente) detallados y la gestión de riesgos técnicos del proyecto.
- contar con un roadmap para TD contribuye a que el negocio pueda comunicar planes a los clientes y dar soporte a estrategias de marketing.
- la arquitectura de software permite "visualizar" la necesidad de nuevas capacidades de recursos humanos (por ej., capacitación) y de la organización (por ej., mejorar la gestión de configuración o el aseguramiento de calidad) para lograr el éxito del proyecto de TD.

Como comentario, se hace notar que si bien la utilización de métodos ágiles para procesos de TD es algo bastante difundido, en grandes procesos de cambios es necesario contar con una visión integradora que haga eficientes a los equipos ágiles y permita gestionar aspectos de deuda técnica. La aplicación del enfoque propuesto muestra que, en grandes proyectos, la agilidad sin integración de todos los elementos (técnicos, humanos y del negocio) es un riesgo que debe ser mitigado. Precisamente, el roamapping está orientado hacia este objetivo.

4. Conclusiones y Perspectiva

La TD es una oportunidad para que las empresas amplíen sus posibilidades, ofrezcan mejores opciones a sus clientes y puedan competir en mejores condiciones en mercados cada vez más complejos y dinámicos. En este sentido, los responsables de las organizaciones deben dedicar tiempo para entender, planificar y aplicar las tecnologías de TD a sus negocios. Para ello, el roadmapping constituye un enfoque flexible para una amplia gama de objetivos y contextos [11]. Las hojas de ruta son "lentes" estratégicos simples y adaptables para atacar la evolución/desarrollo de sistemas a través de la alineación y el consenso de todos los stakeholders a partir de disponer de una visión compartida del programa o proyecto. La arquitectura de software es útil para reforzar esta compartida.

Las principales ventajas observadas por Liveware IS en proyectos de TD utilizando técnicas de roadmapping y arquitecturas son: i) la visibilidad de las decisiones técnicas y su desarrollo en el tiempo, ii) la interrelación (y alineamiento) de dichas decisiones con otras capacidades claves de la empresa (por ej., de producto, de negocio, u operativas, entre otras). El esfuerzo actual del grupo universidad-empresa está focalizado en formalizar el conocimiento capturado por el enfoque propuesto en una herramienta de tipo Wiki que facilite tanto la difusión del mismo, como la documentación ordenada de guías, métodos de arquitectura, y artefactos para su aplicación.

Los proyectos de modernización o de TD no necesariamente son siempre semejantes; nivel de madurez de la organización, el nivel de obsolescencia de las plataformas tecnológicas, diferentes dominios y objetivos de negocios conllevan la necesidad de que los métodos detallados en este artículo deben ser adaptables a diferentes escenarios, pero sin perder el enfoque integrador y facilitador de la ejecución coordinada de las acciones identificadas en la hoja de ruta. En este sentido, la colaboración universidad-empresa apunta a disponer de procedimientos de tailoring de los métodos en función de los proyectos.

Desde la investigación propia del proyecto PDTS, adicionalmente a la captura y sistematización de experiencias, está la necesidad de refinar las relaciones entre modelos y técnicas de arquitectura y técnicas "clásicas" de roadmapping. En este sentido se plantea como trabajo futuro, apoyar la aplicación del enfoque con soporte de herramientas para algunas de las actividades planteadas en las distintas fases. Por ejemplo, se han comenzado a estudiar extensiones de herramientas de roadmapping [12] con elementos de arquitectura de software (por ej., decisiones técnicas, escenarios, atributos de calidad, vistas arquitecturales, etc.).

Referencias

1. G. Westerman, D. Bonnet, and A. McAfee, *Leading digital : turning technology into business transformation*, 2014.
2. Capgemini-Consulting, *Digital Transformation: A Roadmap for Billion-Dollar Organizations.* MIT SLOAN Institute, 2011.
3. M. G. Moehrle, R. Isenmann, and R. Phaal. (2013) Technology roadmapping for strategy and innovation charting the route to success. Berlin; New York.
4. L. Bass, P. Clements, and R. Kazman, *Software Architecture in Practice*, 3rd ed. Addison-Wesley Professional, 2012.
5. E. Y. Nakagawa, M. Guessi, J. C. Maldonado, D. Feitosa, and F. Oquendo, "Consolidating a process for the design, representation, and evaluation of reference architectures," in *2014 IEEE/IFIP Conference on Software Architecture*, April 2014, pp. 143–152.
6. A. Martini, J. Bosch, and M. Chaudron, "Architecture technical debt: Understanding causes and a qualitative model," in *2014 40th EUROMICRO Conference on Software Engineering and Advanced Applications*, Aug 2014, pp. 85–92.
7. I. Ozkaya, A. Diaz-Pace, A. Gurfinkel, and S. Chaki, "Using architecturally significant requirements for guiding system evolution," in *2010 14th European Conference on Software Maintenance and Reengineering*, March 2010, pp. 127–136.
8. A. J. Lattanze, *Architecting Software Intensive Systems: A Practitioners Guide*, 1st ed. Boston, MA, USA: Auerbach Publications, 2008.
9. N. Ernst, M. Popeck, F. Bachmann, and P. Donohoe, "Creating software modernization roadmaps: The architecture options workshop," *2016 13th Working IEEE/IFIP Conference on Software Architecture (WICSA)*, pp. 71–80, 2016.
10. M. Erder and P. Pureur, "Transitional architectures for enterprise evolution," *IT Professional*, vol. 8, no. 3, pp. 10–17, Jan 2006.
11. C. F. R. Phaal and D. Probert, *Technology Roadmapping: linking technology resources to business objectives.* University of Cambridge, 2001.
12. E. Poort, "Just enough anticipation: Architect your time dimension," *IEEE Software*, vol. 33, no. 6, pp. 11–15, 2016.