INSTITUTO TECNOLÓGICO DE BUENOS AIRES – ITBA

ESCUELA DE POSTGRADO

CONFORMACIÓN DE EQUIPOS PARA ABORDAR LA TRANSFORMACIÓN DIGITAL

Caso de reestructuración organizacional y metodológica

AUTOR: Irigaray, Alejandro (Leg. N° 103265)

DIRECTOR DE TESIS: Veloz Rúa, Jimena

TESIS PRESENTADA PARA LA OBTENCIÓN DEL TÍTULO DE MAGISTER EN DIRECCIÓN ESTRATÉGICA Y TECNOLOGÍCA

BUENOS AIRES

PRIMER CUATRIMESTRE, 2023

Agradecimientos

A mi esposa
A mis padres
A mi familia
A mis amigos
A mis docentes

A Equifax, donde comencé mi vida laboral y marcó mi desarrollo profesional a lo largo de 14 años.
De igual forma, agradezco a mi Tutora de Tesis, Jimena, que gracias a sus consejos y sugerencias hoy he podido culminar este trabajo. A los Profesores quienes al compartir sus conocimientos y experiencias me han ayudado a crecer tanto profesional como personalmente.
¡A todos ellos gracias! pues hoy puedo sentirme dichoso y contento.

El presente trabajo está dedicado a mi familia por haber sido y seguir siendo mi apoyo en todo momento a lo largo de mi vida. A Charo, Paulo y Sergio, mis formadores anteriores. A Sergio, Gastón, a todo el equipo que conformó el programa de *Data Fabric Equifax*, y todas las personas que me acompañaron en esta etapa.

A todos ellos, mi agradecimiento por haber confiado en mí y aportado a mi formación tanto profesional como ser humano.

Dedicatoria

A los protagonistas de esta historia, el equipo de Data Fabric, Equifax

Índice de Capítulos:

Abstract 1

Introducción 2

CAPÍTULO I – CONTEXTO DE LA EMPRESA 2

CAPÍTULO II – ESTRUCTURA PREVIA AL PROGRAMA 7

CAPÍTULO III – METODOLOGÍA PREVIA AL PROGRAMA 11

CAPÍTULO IV – TRANSFORMACIÓN DIGITAL 15

- **Esser** 17
  - **Análisis** 18
  - **Estrategia** 18
  - **Diseño** 18
  - **Impacto organizacional** 19
  - **Transformación** 19

CAPÍTULO V – ESTRATEGIA PARA ABORDAR EL PROGRAMA 20

- **Adopción de la nube** 20
  - **Realojamiento (Rehosting)** 20
  - **Refactorización (Refactoring)** 21
  - **Rediseñar (Redesign)** 21
  - **Reemplazar (Replace)** 22
- **Modelo Operativo** 23
- **De foco en proyecto a foco en producto** 23
  - **Ciclo de liberación de funcionalidades** 24
    - **Alpha** 24
    - **Beta** 24
    - **GA** 25
  - **Herramientas** 25
    - **Aha (https://www.aha.io/)** 25
    - **Jira (https://www.atlassian.com/es/software/jira)** 26
    - **Confluence (https://www.atlassian.com/es/software/confluence)** 26

ServiceNow (https://www.servicenow.com/) 27
CAPÍTULO VI – METODOLOGÍA DE REFERENCIA 27
El modelo Spotify 29
Squads y Tribus 29
Dependencias 30
Capitulos y Guilds 30
Estructura de equipos en Spotify 31
Arquitectura 32
Cultura 34
DevSecOps 36
CAPÍTULO VII – NUEVA ESTRUCTURA 39
Perfiles 39
Descripción de puestos y roles 41
Product Manager 41
Product Owner 42
Tribe leader 42
Developer Full-Stack 42
SRE - Ingeniero de confiabilidad del sitio 44
QA 45
Scrum Master 46
Arquitecto 47
Program Manager 47
Analista de Seguridad 49
Plan de carrera 50
Equipos 50
Tamaño 50
Tribus 51
Squads 52
Chapters 52
COEs 53
CAPÍTULO VIII – IMPLEMENTACIÓN 53

Línea de tiempo del programa 54
Alineación organizacional 55
Distribución en tribus y squads 55
Arquitectura, lineamientos y soporte 57
Perfiles 57
Product Manager 57
Product Owner 58
Developer Full-stack 58
SRE 58
QA 59
Scrum master 59
Tribe leader 60
Analista de seguridad 60
Arquitecto 60
Program Manager 60
Responsabilidades 61
Governance 63
Espacio fisico 64
Proceso de gestión de la iniciativa 66
Gestión de iniciativas 66
Estimaciones de desarrollo 66
CAPÍTULO IX - RESULTADO FINAL 68
CAPÍTULO X – CONCLUSIÓN 71
ANEXOS 71
Perfiles de profesionales en tecnología (Cessi) 75
Arquitecto de software 75
Desarrollador de Software 77
Project Manager 81
Analista Funcional 84
Analista Tester 85
Especialista en seguridad de la información 88

Índice de Cuadros:

Cuadro 4. 1 Definiciones de transformación digital Pág. 15
Cuadro 8. 1 Tribus, Squads y cantidad de miembros Pág. 56
Cuadro 8. 2 Matriz RACI Pág. 62

Índice de Figuras:

Figura 1. 1 Impacto del data breach en la acción Pág. 4

Figura 1. 2 Hitos cronológicos del ataque a EFX Pág. 5

Figura 2. 1 Estructura Direccion de tecnologia en el año 2017 Pág. 9

Figura 2. 2 Plan de carrera para Desarrollo Pág. 10

Figura 3. 1 Áreas de Tecnología según etapas de la iniciativa Pág. 13

Figura 4. 1 Oportunidades y desafíos a través de la TD Pág. 17

Figura 5.1 Estrategia de modernización de aplicaciones Pág. 20

Figura 6.1 Estructura organizacional Spotify Pág. 31

Figura 6.2 Alineamiento y Autonomía Pág. 35

Figura 7.1 El ciclo de desarrollo de Software SDLC Pág. 39

Figura 7.2 Tribus definidas Pág. 52

Figura 8.1 Cronograma de alto nivel del programa Pág. 55

Figura 8.2 Curva de Allen Pág. 65

Abstract

En un entorno complejo, plagado de constantes cambios, como la industria de las tecnologías de la información, las empresas se enfrentan a permanentes desafíos, las mismas tratan de adaptar sus estructuras y equipos y acelerar la toma de decisiones para hacer frente a este contexto.

En este caso se analizó el estado de la empresa Equifax para afrontar la transformación digital. Se explica la nueva estructura elegida, esbozando los motivos que dieron origen a dicha elección. Con una gran influencia de las metodologías ágiles que permitieron comenzar una reingeniería aplicativa completa sin afectar la normal operación.

Sobre la estrategia elegida y los cambios realizados tanto en estructura organizacional como metodológica se trata el presente trabajo.

Palabras Claves
Cultura ágil, Cloud, Estrategia, Estructura, Transformación Digital.

Introducción

La transformación digital es un desafío crítico para las empresas en la actualidad, especialmente en la industria de las tecnologías de la información. La necesidad de adaptarse a un entorno en constante cambio, de tomar decisiones rápidas y de ser competitivo en un mercado cada vez más global requiere una estrategia bien definida y un equipo capacitado.

En este trabajo, se analizará el caso de la empresa Equifax y cómo abordó la transformación digital a través de la conformación de su equipo de tecnología y su relación con las diversas áreas de la empresa. Se explorará la estructura elegida y los motivos que dieron origen a ella, así como los cambios realizados en la metodología y en la organización.

El diseño metodológico seleccionado es de tipo descriptivo, basado en la observación directa, el análisis de documentos, libros y artículos universitarios; y la investigación de casos para entender los desafíos y oportunidades en la transformación digital y cómo Equifax respondió a ellos. Además, se explorará la influencia de las metodologías ágiles en la reingeniería aplicativa y los cambios organizacionales.

El estudio de caso de Equifax para Argentina brinda una oportunidad única para comprender cómo una empresa puede abordar la transformación digital a través de la estructura y lineamientos para generar un nuevo equipo de tecnología. Los hallazgos de este trabajo pueden ser útiles para otras empresas que enfrentan desafíos similares en su camino hacia la transformación digital y para futuras investigaciones en este tema.

CAPÍTULO I – CONTEXTO DE LA EMPRESA

Equifax es una agencia multinacional estadounidense de informes de crédito al consumo. Junto a Experian y Transunion, es una de las tres compañías más grandes de Estados Unidos en reportes crediticios. Entre su oferta de productos destaca el cálculo de riesgo crediticio de los consumidores, lo que determina el acceso a préstamos o calificación para la compra de un coche o una casa. Por esta razón, la compañía posee información privilegiada de millones de clientes, como nombres completos, direcciones, números telefónicos, historial crediticio, números de tarjetas de crédito, fecha de nacimiento, números de seguridad social y hasta números de licencias de conducir.

En septiembre de 2017, dicha compañía anunció que había sido víctima de un ataque cibernético. Se trató de un ataque sin precedentes debido a la sensibilidad y el volumen de la información extraída. Entre los datos sustraídos se incluyeron nombres, direcciones, teléfonos, fechas de nacimiento, números de seguridad social y números de licencias de conducir, además de los números de tarjeta de crédito de unas 209.000 personas. Se estima que hubo unos 148 millones de personas afectadas, en su mayoría de Estados Unidos, aunque la filtración también alcanzó a personas de Canadá y Reino Unido. El ataque duró 76 días, donde fueron vulneradas alrededor de 50 bases de datos.

Los atacantes explotaron una vulnerabilidad sobre un componente de código abierto llamado Apache Struts, el cual era utilizado en la plataforma Web de disputas.

El impacto de esta filtración de datos en la empresa fue inmediato. Como demuestra el siguiente gráfico la acción de Equifax perdió el 40% de su valor en la semana de su comunicación.

Figura 1.1

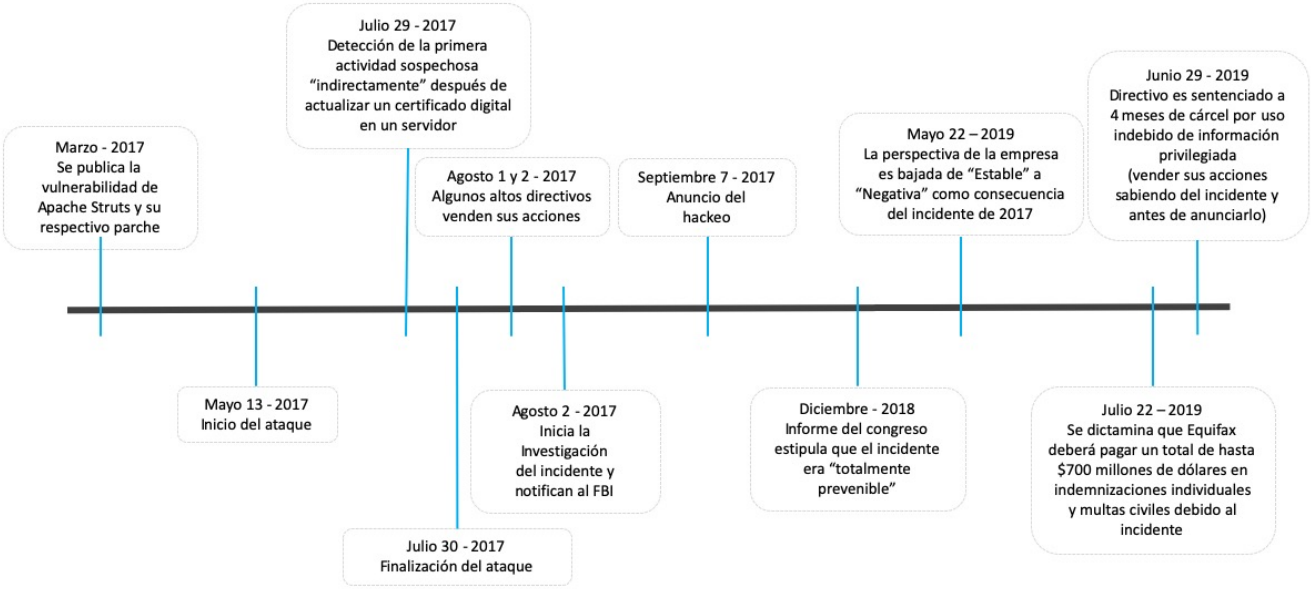
Nota. Impacto del data breach en la acción Enero 2017 - Mayo 2018, por Yahoo Finance, 2017, Yahoo (https://finance.yahoo.com/chart/EFX).

Ante este panorama se tomaron acciones inmediatamente. A la tercera semana de hacerse pública la filtración de datos, ingresó un nuevo CEO a la compañía. Junto a él se incorporó un nuevo Director de Seguridad con experiencia en situaciones similares. Dos meses después se incorporó el nuevo Director de Tecnología con el compromiso de modernizar y asegurar todos los sistemas. Equifax invirtió alrededor de 200 millones de dólares en ciberseguridad durante el primer año.

Un fuerte programa se lanzó a todas las filiales del mundo para relevar el estado de madurez en cuanto a seguridad, evaluando minuciosamente los riesgos en cada una. Para luego dar inicio a la iniciativa de remediación o transformación de todos los aplicativos. Dicha iniciativa abarcó la seguridad en todos sus aspectos, desde gestión de ingresos del personal y terceros a las facilidades hasta actualizaciones de versiones de componentes de software, reemplazo o apagado de aplicativos catalogados como riesgosos.

A continuación, se muestra una línea de tiempo sumarizando los hitos más importantes y las consecuencias de dicho ataque:

Figura 1.2.



Nota. Hitos cronológicos del ataque a EFX, El ciberataque a Equifax como caso de estudio, por Magazcitum, 2020, Magazcitum (https://www.magazcitum.com.mx/index.php/archivos/5222#.YdygTBNBw-Q).

Finalizado el análisis de situación de Equifax se inició el proceso de Transformación Digital de la compañía. Citando a Bryson Koehler en una entrevista (Apigee, 2019). [1]

"Nuestra transformación es una migración completa de nuestras capacidades y una reingeniería completa de nuestros activos de datos, nuestros burós y nuestras aplicaciones hacia la nube. Lo veo fundamentalmente como Equifax moviéndose de un proveedor de datos por lotes a un tomador de decisiones en tiempo real profundamente integrado con nuestros negocios y clientes. De eso se trata la transformación digital" [...] Cambiar nuestra manera de trabajar, ser más colaborativos, lo cual representa mayormente un desafío de gestión de gente. [...] Seguridad y confianza son las cosas más importantes para

[1]Apigee. (2019, 20 de septiembre). The Equifax digital transformation, powered by Google Cloud [video]. Youtube. https://www.youtube.com/watch?v=gxCQHE4pD_8

nosotros en Equifax”

El hecho de haber sufrido uno de los ataques más grandes de la historia, demandó tomar medidas de carácter urgente, lo que dio origen al programa de transformación digital persiguiendo la modernización y el aseguramiento de que todos los sistemas fuesen seguros. Si bien el programa fue de alcance global, este trabajo se centra en el impacto que generó afrontar el programa en Argentina.

Como consecuencia del ataque, Equifax aceleró sus inversiones en todas sus filiales, dando como resultado el programa de transformación digital. Dicho programa persiguió la estandarización de tecnología, procesos y estructuras.

Este programa implicó la adopción de plataformas corporativas, como así también la reingeniería de todos los aplicativos con el objetivo de estar funcionando en la nube para poder aprovechar sus beneficios, tanto en lo referente a seguridad, como ser encriptación total de toda la información almacenada en lo relacionado a agilidad en los cambios en los sistemas, facilitando nuevos enfoques metodológicos.

El programa abarcaba el clúster de APU (Argentina, Paraguay y Uruguay). Comenzando por Argentina se estimó una duración de tres años para Argentina. La transformación abarcaba más de 100 aplicativos, que daban soporte a más de 500 productos para 9000 clientes.

Transcurridos 12 meses de programa se comenzaría en paralelo con Uruguay y Paraguay, ambos de una escala similar, comprendiendo aproximadamente 30 sistemas en cada uno, para finalizar el programa para los tres países al mismo momento. El tiempo total estimado fue: tres años para Argentina, dos para Uruguay, y dos para Paraguay.

Tecnológicamente el sistema principal de Argentina era un monolítico escrito enteramente en Oracle PlSql con una antigüedad de aproximadamente 20 años. Durante los últimos 10 años los nuevos productos se basaron en una arquitectura de tres capas, escritos principalmente en java para los servidores y en html y javascript para la capa de presentación. Sin embargo, el acoplamiento era muy fuerte ya que todos los aplicativos consumían servicios de la base de datos core. El sistema core no podía funcionar 7x24 ya que diariamente se contaba con una ventana de mantenimiento de aproximadamente 2 horas para poder realizar cambios y depuración de procesos. El core de Uruguay también era una aplicación monolítica, en este caso escrita en tecnologías .net, mientras que Paraguay contaba con dos bases principales una basada en Oracle con servicios Java y .net y otra basada en sybase, Java y .net -producto de una adquisición realizada aproximadamente 5 años antes. Producto de dicha adquisición ambos

sistemas se comportaban como uno único a efectos lógicos, sin embargo, ambos sistemas operaban de manera independiente.

Como concluye Cuenca Diaz en su exposición sobre ciberseguridad (Cuenca Diaz, 2012):

- El alcance del desarrollo seguro abarca todas las etapas del desarrollo de software.
- Se deben estandarizar los controles y requisitos, mediante guías y formularios.
- La seguridad no es un producto, es una sumatoria de personas, procesos y tecnología.
- Suele ser más costoso aplicar seguridad al final y no durante el proceso.

El programa de transformación digital perseguía fundamentalmente la estandarización de tecnología, procesos y estructuras. Para así poder garantizar la seguridad, operación y mantenimiento de todos los sistemas. Los sistemas legados debían cursar una reingeniería para comenzar a funcionar en la nube, una tecnología no adoptada en Equifax Argentina.

CAPÍTULO II – ESTRUCTURA PREVIA AL PROGRAMA

Organización regional

Argentina, Paraguay y Uruguay conformaban uno de los tres clústeres de Latinoamérica (APU), siendo los otros dos Chile, Perú y Ecuador (PEC) y Honduras, El Salvador y México (CAMEX). Estos tres grupos conformaban toda la línea de Latinoamérica de Equifax. Cada clúster contaba con un Director General y las estructuras de reportes no eran homogéneas entre ellos.

En el caso de APU las direcciones de marketing, legales y ventas eran federadas, si bien existía un reporte único por área cada país operaba de manera independiente. En el caso de tecnología, seguridad, operaciones y finanzas eran estructuras centralizadas, todas operaban como una única unidad.

Tecnología

La dirección de tecnología en el año 2017 operaba con una organización con personas distribuidas entre Argentina, Paraguay y Uruguay, y estaba compuesta por 5 Áreas:

Arquitectura
PMO (oficina de gestión de proyectos)
Proyectos
Mantenimiento
Infraestructura

Arquitectura estaba compuesta por tres integrantes, su función era marcar los estándares de desarrollo y la evolución estratégica de los sistemas, análisis y estimaciones de nuevas iniciativas, confección y presentación de los respectivos casos de negocio.

La oficina de gestión de proyectos estaba compuesta por cuatro integrantes. Su principal función era consolidar el estatus de todas las iniciativas y manejar la demanda de todas las áreas de la empresa hacía tecnología.

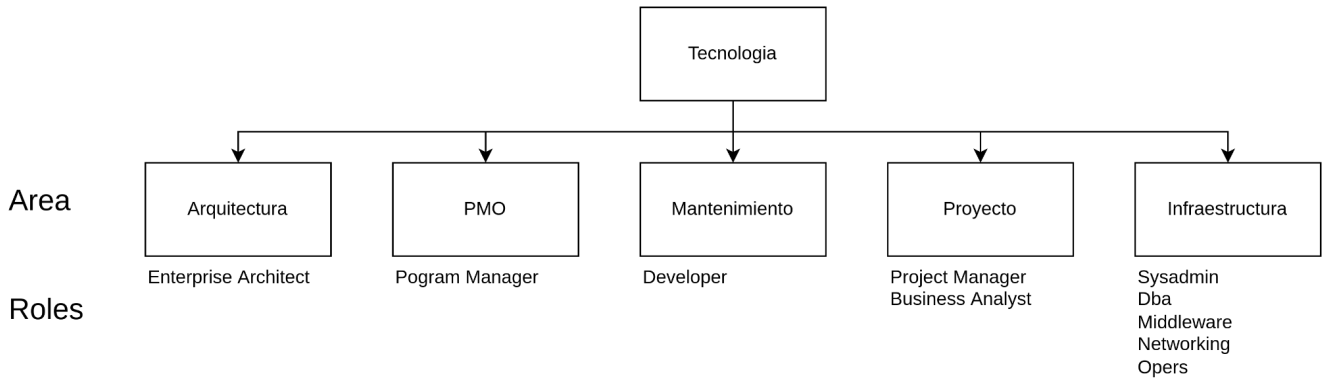
Proyectos se dedicaba al manejo de iniciativas nuevas o que implicaran un cambio complejo sobre los sistemas existentes. Se componía de 10 personas -entre Analistas de Negocio y Gerentes de Proyecto- y una cantidad variable de desarrolladores que se contrataban por proyecto específico. En general cuando

se comenzaba un proyecto nuevo se conformaba un equipo de trabajo compuesto por un gerente de proyecto, un analista y entre 4 y 6 desarrolladores. El gerente quedaba asignado un 50% de su tiempo a la iniciativa, mientras que el analista estaba al 100%.

Mantenimiento, donde se daba seguimiento a las incidencias diarias de cualquier sistema, se componía de aproximadamente 15 desarrolladores distribuidos en los tres países (Argentina, Paraguay y Uruguay), generalmente las personas de cada país atendían los sistemas locales por una cuestión de experiencia y conocimiento.

Infraestructura cuya función era garantizar el correcto funcionamiento y la disponibilidad de todos los sistemas, como así también manejar la operación y cualquier función relacionada con la recuperación de los sistemas como por ejemplo estrategias y gestión de copias de respaldos (backups) o aseguramiento del plan de recuperación ante desastres (DRP). Compuesta por aproximadamente 20 personas.

Figura 2.1.



Nota. Estructura Dirección de tecnología en el año 2017 (2022), fuente propia.

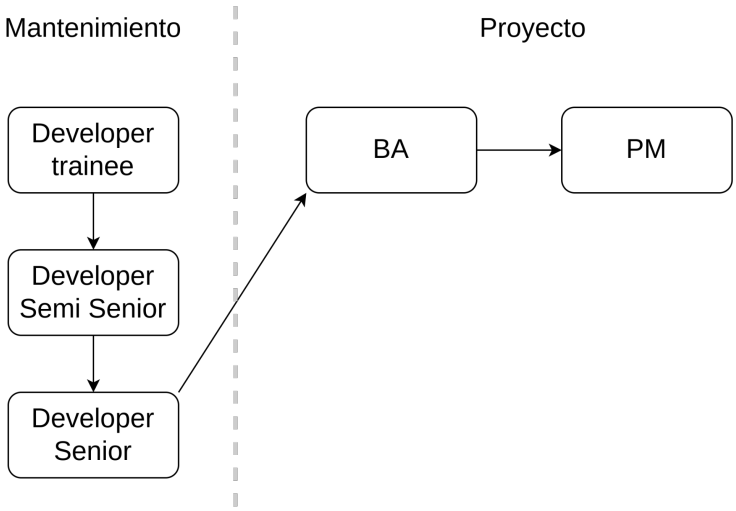
Plan de carrera de tecnología

Para la mayoría de las posiciones de tecnología no había un plan de carrera claro. Basándose en movimientos anteriores algunas personas cambiaban de área, siendo las más frecuentes marketing,

seguridad y operaciones. Aquellas personas que quisieran convertirse en expertos en sus tecnologías podían aspirar a una posición similar para otra región, pero al no contar con un horizonte claro, generalmente ocurría que al cabo de algunos años optaban por abandonar la empresa, causando un impacto grande ya que el conocimiento especializado adquirido residía en muy pocas personas.

El caso de desarrollo, contemplando tanto Mantenimiento como Proyecto era particular. Usualmente el plan de carrera para los integrantes de dichos equipos estaba dado por un crecimiento que implicaba un cambio de rol. Es importante destacar que estos cambios además implicaban un incremento salarial. Con lo cual se daba un esquema habitual en el que el buen Desarrollador (que siempre pertenecía a Mantenimiento) se convertía en Analista de Negocio (pasando al área de Proyecto), y el buen Analista de Negocio se convertía en Gerente de Proyecto.

Figura 2.2.



Nota. Plan de carrera para Desarrollo, área de Tecnología, en el año 2017 (2022), fuente propia.

CAPÍTULO III – METODOLOGÍA PREVIA AL PROGRAMA

Con la estructura presentada anteriormente se daba soporte a la demanda de requerimientos. Las principales áreas de las cuales se recibían solicitudes eran Marketing y Operaciones. Marketing responsable por el desarrollo de nuevos productos y la evolución de los existentes, y Operaciones responsable de la carga y control de la información. En menor medida se recibían solicitudes de áreas como Legales, en caso de que hubiese nuevas disposiciones que afectaran al tratamiento de los datos y Finanzas, dado que el sistema de facturación era un desarrollo propio por lo que cualquier modificación implicaba desarrollo.

Hasta el momento del inicio del programa se utilizaba mayormente un enfoque en cascada. Basado en el modelo propuesto por Bohem en 1980, el cual se caracteriza porque una etapa del ciclo de vida de un programa no puede comenzar sin que se haya finalizado la etapa anterior.

Si bien se habían adoptado ciertas prácticas relacionadas con las metodologías ágiles, nunca se llegó a un estadio maduro, se trataba más bien de esfuerzos independientes que de prácticas formales.

Como se mencionó anteriormente el manejo de la demanda estaba centralizado en la oficina de gestión de proyectos. Dicha área registraba todas las solicitudes internamente -no se contaba con una herramienta colaborativa donde los requirentes pudiesen ver toda la demanda-. Una vez registrados y realizado un chequeo completo, se enviaban dichas solicitudes al área de arquitectura para ser estimados en costos y tiempos -la mayoría de las veces involucrando a proveedores externos- Una vez estimados, volvían a la Pmo para que sean priorizados en conjunto con el negocio. Usualmente se realizaba una reunión semanal con todos los involucrados donde se establecía y se comunicaba el orden y los motivos de todos los pendientes. Cabe destacar que la demanda nunca pudo cumplirse plenamente, llegando a haber más de 100 requerimientos pendientes, de los cuales en una estimación de alto nivel surgió que había 20 de complejidad alta (más de 6 meses), 40 de complejidad media (de 2 a 6 meses) y 40 de complejidad baja (menos de un mes).

Siguiendo con el proceso, si el requerimiento era priorizado entonces se hacía una transferencia de conocimiento desde arquitectura a desarrollo para realizar el desarrollo de manera interna o para que proyectos pudiera dar seguimiento y supervisar los desarrollos del proveedor. Generalmente para desarrollos internos se asignaban 2 desarrolladores y la duración no solía superar el mes de dedicación.

Para proyectos contratados se asignaba un administrador de proyectos y un analista de negocios del staff, y usualmente se contrataban equipos de aproximadamente 5 personas de un proveedor. Solo se contaba con dos proveedores homologados para desarrollos, lo cual implicaba un manejo de la capacidad de recursos en conjunto con el proveedor, ya que generalmente se optaba por continuar con gente que ya había trabajado en otros proyectos dentro de la empresa. La antigüedad de la relación entre la empresa y los proveedores era de más de 10 años.

Previo a la contratación de proveedores debía pasar obligatoriamente por un proceso de caso de negocio, donde se solicitaba el monto necesario para la contratación y se explicaban los motivos de dicho desarrollo. Esta aprobación constaba de dos etapas, una primera etapa local enfocada en el chequeo financiero, y luego otra etapa regional donde se realizaba un chequeo tecnológico para garantizar que se cumpliesen los estándares y la estrategia definida.

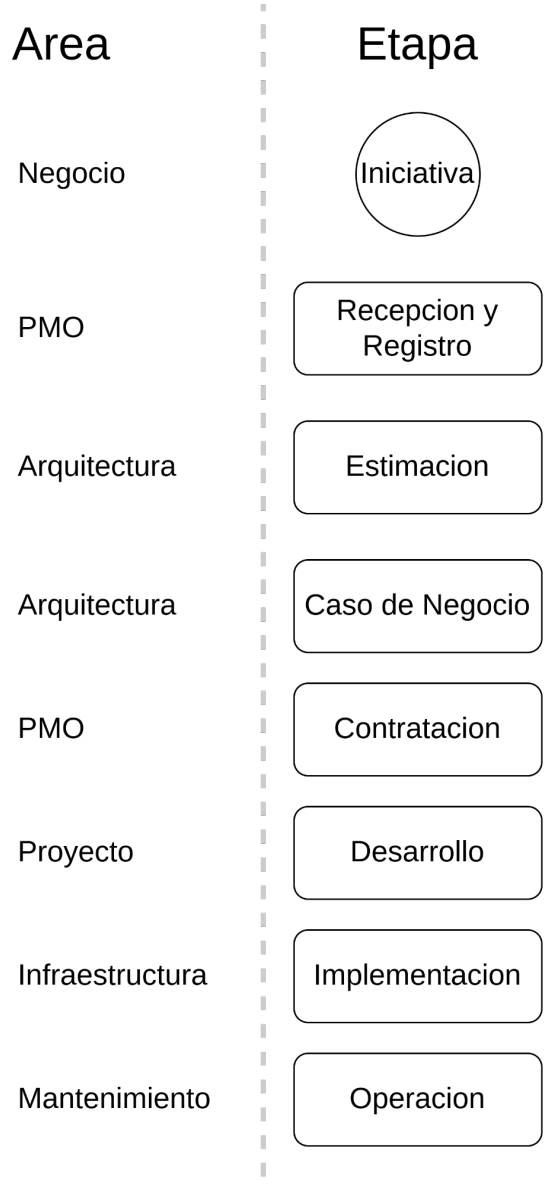
Dicho proceso de elaboración y aprobación era muy variable dependiendo estrechamente del monto de la solicitud, pudiendo demandar 1 mes en los casos más sencillos y entre 6 y 9 meses en los casos más complejos.

Una vez aprobado el caso de negocio entonces se convocaba a una reunión de inicio de proyecto con todos los stakeholders y el equipo involucrado.

Ya en etapa de desarrollo se involucraba a infraestructura para la preparación de los ambientes necesarios para llevar a cabo la iniciativa. Generalmente se contaba con un mínimo de dos ambientes uno para desarrollo y otro para pruebas.

Una vez finalizado el desarrollo se comenzaba con el proceso de solicitud de cambio con el objetivo de formalizar y tener trazabilidad de cualquier cambio realizado en producción. Dicho proceso involucraba diversos actores para la correspondiente aprobación. Dicho proceso duraba al menos una semana, ya que la implementación sólo podía realizarse en las ventanas de mantenimiento definidas (dos días a la semana). Para lograr la aprobación era necesario obtener la aprobación en una junta de aprobación semanal donde había que explicar y defender el cambio.

Figura 3.1.



Nota. Áreas de Tecnología según etapas de la iniciativa, en el año 2017 (2022), fuente propia.

La empresa operaba acorde a un efecto “Silo” (Tett G, 2015). Al hablar de Silos Tett utiliza una metáfora para referirse a compartimentos estancos que en una empresa no se comunican, compiten entre sí y sufren de puntos ciegos y de visión en túnel, lo que disminuye la capacidad de anticipar crisis y reaccionar de forma adecuada. Estas organizaciones tienden a encasillar el pensamiento en estructuras ordenadas, sistemas de información, burocracia, comunicación y operación.

El ‘Efecto silo’ consiste en la sobre-estructuración de una organización. Los silos son característicos de una excesiva y mal equilibrada visión vertical de la organización. Por lo tanto, el ‘Efecto silo’ hace que los problemas o la cooperación interdepartamentales se compliquen, sobre todo en niveles medios o bajos.

Los silos dificultan y retardan los flujos de información, ya que ésta debe fluir hacia arriba y, posteriormente, una vez que se haya tomado decisiones, ha de regresar hacia abajo para ser ejecutada.
Una organización con silos es sinónimo de una organización burocrática, poco ágil y resistente a la innovación.

Esta manera de operar se observaba tanto entre departamentos, por ejemplo en la comunicación entre marketing y tecnología. Como también intra área, como se expuso en tecnología, donde desarrollo e infraestructura no compartían procesos. Tecnología y Seguridad también operaban como Silos. No existía un proceso en común durante la etapa de desarrollo donde ambas áreas trabajaran juntas, más bien se trataba de un esquema donde seguridad auspiciaba de auditor, levantando expuestos de seguridad y solicitando planes de cumplimiento, pero sin una participación activa. Esto usualmente implicaba que estas fechas de compromiso fuesen replanificadas bajo un nuevo acuerdo, ya que la decisión por parte de negocio tenía más peso que la de seguridad.

CAPÍTULO IV – TRANSFORMACIÓN DIGITAL

Si bien no hay una única definición comúnmente aceptada sobre transformación digital (Schalmo, 2018), como podemos observar en las definiciones de la figura 5.1, la gran mayoría se centra en la transformación de la experiencia del usuario y en el descubrimiento de nuevos o mejorados modelos de negocios apoyándose en la tecnología para sustentar diversos tipos de decisiones.

Cuadro 4.1.

Referencia	Definición
BMWi (2015: 3)	La digitalización representa la interconexión completa de todos los sectores de la economía y la sociedad, así como la capacidad de recopilar información relevante y analizar y traducir esta información en acciones. Los cambios traen ventajas y oportunidades, pero crean desafíos completamente nuevos
Bowersox et al. (2005: 22ff)	Digital Business Transformation es el proceso de reinventar un negocio para digitalizar operaciones y formular relaciones de cadena de suministro extendidas. El desafío de liderazgo de DBT [Digital Business Transformation] se trata de revitalizar las empresas que ya pueden tener éxito para capturar todo el potencial de la tecnología de la información en toda la cadena de suministro"
Westerman et al. (2011: 5)	"La transformación digital (DT), el uso de la tecnología para mejorar radicalmente el rendimiento o el alcance de las empresas se está convirtiendo en un tema candente para las empresas de todo el mundo. Los ejecutivos de todas las industrias están utilizando avances digitales, como análisis, movilidad, redes sociales y dispositivos integrados inteligentes, y están mejorando su uso de tecnologías tradicionales como ERP, para cambiar las relaciones con los clientes, los procesos internos y las propuestas de valor".
Mazzone (2014: 8)	"La Transformación Digital es la evolución digital deliberada y continua de una empresa, modelo de negocio, proceso de idea o metodología, tanto estratégica como tácticamente"
PwC (2013: 9)	La transformación digital describe la transformación fundamental de todo el mundo empresarial a través del establecimiento de nuevas tecnologías basadas en Internet con un impacto fundamental en la sociedad en su conjunto.
Boueé and Schaible (2015: 6)	Entendemos la transformación digital como un trabajo en red consistente de todos los sectores de la economía y el ajuste de los jugadores a las nuevas realidades de la economía digital. Las decisiones en los sistemas en red incluyen el intercambio y análisis de datos, el cálculo y la evaluación de opciones, así como el inicio de acciones y la introducción de consecuencias.

Nota. Definiciones de transformación digital (2018). D. R. A. Schallmo, C. A. Williams, Digital Transformation Now!, p 10

La transformación digital se basa en cuatro pilares fundamentales: Tecnología, Procesos, Personas y Datos. Los tres primeros quedan explicados por el marco de trabajo PPT. Este marco fue definido en los primeros años de la década de 1960, y ha sido utilizado en incontables ocasiones para mejorar la eficiencia operativa de los empleados y las herramientas. El marco hace foco en la interacción entre los tres elementos. Las personas realizan el trabajo, los procesos guían al trabajo para que sea más eficiente, la tecnología ayuda a las personas a realizar su trabajo y a automatizar procesos. De esta manera los negocios se benefician al encontrar el balance y optimizar las relaciones entre ellos. Para considerarse dentro de un proceso de transformación hace falta adicionar el cuarto elemento. Se trata de los datos, gracias a las capacidades computacionales -mayormente soportados por tecnologías en nube- se posibilita el análisis de grandes volúmenes de datos (big data), con el fin de tomar mejores decisiones, ya sea tomando data histórica o previendo el futuro.

Al ser la transformación digital un concepto donde la tecnología es un pilar fundamental, muchas personas consideran que se trata únicamente en digitalizar las operaciones y servicios de una organización, sin tomar en consideración otros aspectos fundamentales como las personas, gobernanza, procesos, políticas, cultura, productos o servicios. En la publicación de Harvard Business Review (Behnam Tabrizi, Ed Lam, Kirk Girard, and Vernon Irvin, 2019) se afirma que durante el 2019 la Transformación Digital fue el punto de mayor interés para los CEOS, sin embargo, el 70% de las iniciativas de Transformación Digital no alcanzaban sus objetivos. Esto se debía mayormente a que las tecnologías proveen posibilidades para ganar eficiencias y acercamiento a los clientes. Sin embargo, si las personas carecen de la mentalidad adecuada para cambiar y las prácticas organizacionales son defectuosas, la Transformación Digital simplemente magnificará estas fallas.

En un sentido aplicable a Equifax la Transformación implicaba la implementación de nuevas tecnologías -tanto para la explotación de los datos como para el desarrollo de aplicativos-, la adquisición de talento, y la mejora y estandarización de procesos para mejorar la operación del negocio para así satisfacer a los clientes. De este modo desde lo tecnológico se garantizaba contar con las herramientas más modernas en materias de seguridad, complementando desde la perspectiva de procesos con revisiones constantes sobre la calidad del código y alertas constantes sobre posibles vulnerabilidades, y apoyándose en equipos de seguridad cuyos lineamientos atravesarían los límites geográficos para brindar estándares globales se garantizaría que Equifax cumpliese con los mayores estándares de seguridad.

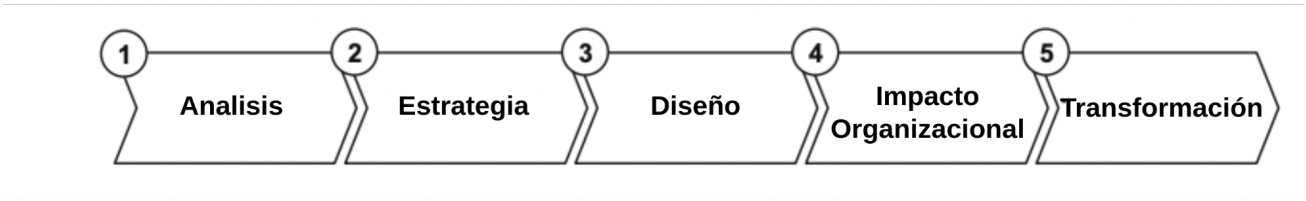
Como se explicó anteriormente la Transformación no es ámbito particular del área de tecnología, sino que por el contrario abarca a todas las áreas de una empresa. Es por este factor que fue fundamental la cadencia de la iniciativa desde el primer nivel de la jerarquía organizacional.

Existen diversos enfoques para llevar adelante un proceso de transformación digital que pueden ayudar como base para una hoja de ruta. El presente trabajo se basa en el libro Digital Transformation Now (Schallmo y Williams, 2018). En este trabajo se seleccionará uno de ellos y se hará foco en las etapas que permitan explicar el cambio y guiar el análisis del impacto en el programa anteriormente mencionado en Equifax.

Esser

Esser (2014) define cinco fases que perfilan el plan de desarrollo de una estrategia de transformación y su implementación

Figura 4.1.



Nota. Esser M (2014) Oportunidades y desafíos a través de la transformación digital. http://www.strategy-transformation.com/digitale-transformation-verstehen/

Análisis

El análisis se enfoca en cuatro áreas: clientes, competidores, mercados y negocios, y capacidades. Se analizan y segmentan las necesidades y valores de los clientes. Los competidores se describen y miden por su desempeño actual y su posicionamiento en el mercado. También se consideran los nuevos competidores recién llegados al mercado. Se analiza el mercado de acuerdo con su tamaño, su potencial, sus limitaciones y desarrollos futuros. Finalmente se recopilan las capacidades empresariales disponibles.

En el caso de Equifax esta etapa se centró en clientes y capacidades. Debía recuperar su lugar en el mercado, dando fuertes señales hacia los consumidores y los dueños de los datos. Garantizando las mejores prácticas en relación a la seguridad informática.

Estrategia

La segunda fase de estrategia incluye definir la posición en el mercado, decidir cómo el negocio quiere diferenciarse y seleccionar el grupo objetivo de clientes.

Aquí el centro de Equifax fue tratar de estabilizar su posición en el mercado, ya que luego del ataque se encontraba en una situación muy delicada. No se trató de buscar una estrategia diferenciadora ni de cambio de segmento objetivo de clientes.

Diseño

El diseño se basa en tres áreas: la visión de la experiencia del cliente, la propuesta de valor y la identificación de oportunidades. La visión de la experiencia del cliente incluye una declaración sobre lo que le gustaría lograr a la empresa. La propuesta de valor responde a la pregunta de cómo y con qué servicios se abordará a los clientes. Finalmente, la identificación de oportunidades evalúa las actuales y las nuevas ideas de diseño.

Aquí Equifax primero se centró en brindar ayuda y soporte a las personas cuyos datos habían sido robados, por ejemplo, mediante el lanzamiento de una web de consulta sobre si su información había estado dentro de los datos robados o brindando un producto de alertas de uso de la información de manera gratuita durante un año. Por otro lado, a los clientes directos se les garantizaría que Equifax cumplía con los controles de seguridad adecuados, y en muchos casos se debería trabajar en conjunto con ellos, por ejemplo para mantener toda la información encriptada y que la misma fuese transmitida sólo por canales seguros.

Impacto organizacional

El impacto organizacional se refiere a las personas, la estructura y la cultura dentro de la empresa. Además, se examinan los procesos y sistemas, y finalmente se definen el gobierno y el control.

Transformación

La etapa de transformación dicta la hoja de ruta y la gestión del programa. Además, se planifican la comunicación interna y la gestión del cambio. Por último, se define el branding y la comunicación externa.

Con respecto a estas dos etapas para el caso de Equifax serán abordadas con mayor detalle en los próximos capítulos.

Para poder transitar el programa y lograr un nuevo modelo operativo se consideran los siguientes puntos como críticos y centrales:

- Atraer y retener al talento humano especializado en tecnologías a utilizar
- Rediseñar el negocio y los productos alrededor de las expectativas de los clientes
- Cocreación y alineamiento entre Tecnología, Negocio y Seguridad
- Alinear objetivos con las metas comerciales

CAPÍTULO V – ESTRATEGIA PARA ABORDAR EL PROGRAMA

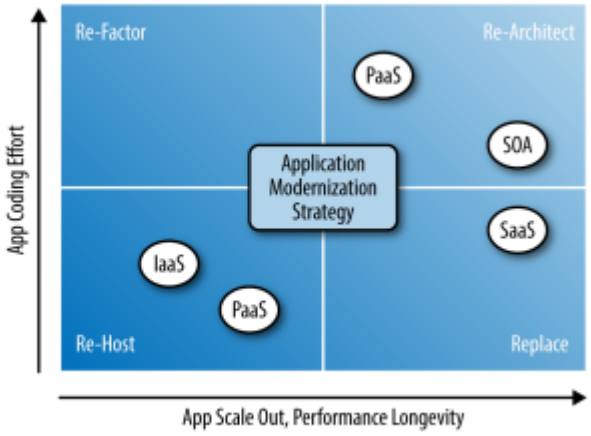
En la alineación corporativa había algunos puntos obligatorios que cumplir, adoptar la nube como base de nuestra tecnología, definir y utilizar un marco ágil para todo el ciclo de vida del producto y formar la estructura que mejor soportara dicho marco.

Desde el punto de vista comercial la empresa debía seguir con su operación habitual, brindando el servicio a todos sus clientes, y en paralelo afrontar este programa.

Adopción de la nube

Para adoptar la tecnología de la nube existen 4 estrategias de modernización de aplicaciones para migrar sistemas heredados hacia la nube.

Figura 5.1.



Nota. James Bond. (2015) Estrategia de modernización de aplicaciones.

The Enterprise Cloud, Best Practices for Transforming Legacy IT. (p 182).

Realojamiento[2]

[2] Del inglés Rehosting

Consiste en mover la aplicación a la nube tal cual está on-premise. Se llevará con la misma arquitectura, el mismo dimensionamiento y la misma configuración. Esta estrategia se denomina Lift & Shift cuando se hace una copia de la aplicación utilizando herramientas automáticas. Y se habla de Reinstall o Greenfield cuando se hace una instalación desde cero desplegándose en la nube de la misma manera que estaba on-premise.

Generalmente se requiere de pruebas reducidas y cambios menores en la configuración, se suele hablar de un esfuerzo de días o semanas. Si bien parece una alternativa tentadora el gran inconveniente que presenta es que no se pueden aprovechar algunos beneficios de la nube como ser resiliencia o elasticidad, como consecuencia suelen presentar costos de operación elevados.
Esta estrategia se suele usar generalmente motivada por la urgencia dentro del contexto de una gran migración en la que hay que mover un alto número de aplicaciones.

Refactorización[3]

La refactorización de una aplicación es similar al realojamiento, pero algunos componentes de la aplicación se mejoran o cambian. Se puede dividir la aplicación en componentes como servidores web frontend y bases de datos backend de modo de poder agrupar cada nivel y equilibrar la carga para proporcionar una mayor disponibilidad, elasticidad y manejar más tráfico y carga de trabajo. En este escenario, la aplicación necesita poca o ninguna reprogramación porque la infraestructura de la nube proporciona parte de la escalabilidad de una manera transparente para la aplicación.

Rediseñar[4]

Si una aplicación no puede realocarse o refactorizarse, porque por ejemplo de hacerlo presentaría problemas de rendimiento. Debería considerarse el rediseño de esta. En general se trata de aplicaciones

[3] Del inglés Refactoring
[4] Del inglés Redesign

heredadas viejas y muy complejas, que pueden incluir diversos componentes como ser plataformas multicapas, bases de datos, middleware y servidores tanto de backend como de frontend.

Suele ser la estrategia más trabajosa en términos de esfuerzo, pero la que mayores beneficios brinda ya que aprovecha al máximo los beneficios de la nube.

Reemplazar[5]

Consiste en adquirir servicios existentes en modalidad Saas (Software as a Service) y eliminar la aplicación heredada. Suele ser una solución rápida y efectiva. En este caso la principal desventaja es que en general no aplica para desarrollos a medida. Ejemplos de aplicaciones candidatas son aplicaciones de mail, CRM (Customer Relationship Management), ERP (Enterprise Resource Planing), etc.

Por las características del sistema legado (una aplicación monolítica fuertemente acoplada). La alternativa seleccionada fue rediseñar, bajo un esquema de desacoplamiento, siguiendo una arquitectura de microservicios. Se generarían nuevas versiones de los productos interrelacionando la malla de servicios, para así brindar a los clientes una oferta superadora a los productos legados. Algunas características fundamentales para motivar la adopción de parte de los clientes eran:

Usabilidad: todas las aplicaciones serían diseñadas para ser utilizadas en distintos dispositivos, ya que los anteriores no eran responsivas.

Disponibilidad: Todas las aplicaciones se diseñarían para poder funcionar de manera continua (7x24)

Mantenibilidad: a diferencia del sistema legado donde había una pequeña ventana de mantenimiento diaria, aquí se diseñaría para permitir realizar la mayoría de los cambios en cualquier horario. Siguiendo esquemas de deployment como ser: blue-green o A/B Testing.

Para poder realizar la reingeniería de manera adecuada se procedió a realizar una racionalización de productos. Identificando las familias y a que nuevo aplicativo debería ir cada producto. De allí que de un total de aproximadamente 400 productos quedaron definidos 10 aplicativos para clientes, y

[5] Del inglés Replace

aproximadamente 20 sistemas internos que brindarán las mismas funcionalidades que el monolítico legado.

Modelo Operativo

Un modelo operativo representa cómo y quién pone en marcha la estrategia empresarial. El modelo operativo define las formas en que las personas colaboran para lograr los grandes objetivos que se describen en la estrategia empresarial. Los modelos operativos suelen describirse como las personas, los procesos y la tecnología que hay detrás de la estrategia empresarial.[6]

Definida la estrategia de rediseñar entonces era responsabilidad de cada región realizar este rediseño bajo las normas dictadas por la corporación. Antes de productivizar un desarrollo se debía cursar por un proceso de revisión arquitectónica, donde se involucraba a diversos equipos donde cada uno debería dar su aprobación explícita en base a las evidencias presentadas.

De este modo se brindaba autonomía a los equipos, pero con un control centralizado se garantizaba cierto nivel de homogeneidad entre las aplicaciones.

De foco en proyecto a foco en producto

Para habilitar el rediseño de estos aplicativos se debía comenzar a pensar en producto, en lugar de proyectos como había sido tradicionalmente. Esto implicaba que se redefiniese enteramente el producto, siguiendo algunas guías para garantizar consistencia entre los diferentes aplicativos, como por ejemplo un uso homogéneo de la paleta de colores, ubicación de ciertos componentes en la pantalla, lineamientos de seguridad y desarrollo.

Esto también motivó a que se pausaran los desarrollos en sistemas legados y toda nueva modificación o funcionalidad que se necesitase sería implementada en los nuevos aplicativos, ayudando así a su futura adopción de parte de los clientes. Aquellos requerimientos que no pudiesen esperar a las

[6] Recuperado de https://learn.microsoft.com/es-es/azure/cloud-adoption-framework/operating-model/terms

nuevas versiones entonces debían realizarse en el sistema legado y planificarse como funcionalidad mandatoria en los nuevos aplicativos. Todo producto fue definido a partir de documentación nueva, con el fin de replicar la lógica de los sistemas legados de las que no se tenía certeza de porqué existía. En general las aplicaciones tenían mucho código muerto o funcionalidades que dejaron de ser utilizadas pero jamás fueron deprecadas. De este modo cada aplicativo comenzaba a tener su propia hoja de ruta y la responsabilidad de decidir qué funcionalidades se implementarían y en qué tiempo.
Para guiar los roadmaps se diseñaron diversas etapas en la vida de un producto.

Ciclo de liberación de funcionalidades

Alpha

El software Alpha tiene todas las funciones, pero a veces sólo funciona parcialmente. Se espera que las versiones alfa sean inestables, quizás un poco inseguras, pero definitivamente utilizables. Pueden tener errores y problemas conocidos que aún no se han resuelto. Antes de lanzar un alfa, asegúrese de tener en cuenta que los lanzamientos alfa siguen siendo lanzamientos y la gente no esperará una compilación nocturna de la fuente de CVS. Un alfa debería funcionar y tener pruebas mínimas y corrección de errores ya terminados. (Hill, 2022)

Es la primera versión del aplicativo. Se trata de una etapa muy temprana del producto, se hace foco en los prototipos de UI y el entendimiento del producto por parte de todo el equipo. Si bien puede que el aplicativo sea inestable ya debe cubrir todas las funcionalidades especificadas, si por algún motivo las mismas no fueron implementadas entonces debe estar bien identificada la brecha y su inclusión no debe alterar otro componente.

Beta

El software beta tiene todas las funciones y es funcional, pero está en el ciclo de prueba y aún le quedan algunos errores por corregir.

En general, se espera que las versiones beta sean utilizables y ligeramente inestables, aunque definitivamente no son inseguras. Las versiones beta normalmente impiden una versión completa en menos de un mes. Pueden contener pequeños errores conocidos, pero no importantes. Toda la

funcionalidad principal debe implementarse por completo, aunque aún se puede resolver la mecánica exacta. Los lanzamientos beta son una gran herramienta para despertar el apetito de los usuarios potenciales al brindarles una visión muy realista de dónde estará su proyecto en un futuro muy cercano y pueden ayudar a mantener el interés al brindarles algo. (Hill, 2022)

Se trata del producto mínimo viable finalizado, utilizable y listo para ser probado por el usuario interno y algunos usuarios externos selectos con acompañamiento. Puede que surjan cambios y correcciones, pero no se aceptan cambios de alcance, los mismos se gestionan con su adecuado proceso, ya que el alcance debió haber sido acordado en la fase previa. Se comienza con la estrategia de desembarco al mercado

GA

"Versión de desarrollo" es un término mucho más vago que "alfa" o "beta". Por lo general, elijo reservar el término para la discusión de una rama de desarrollo, aunque hay otras formas de usar el término. Tantos, de hecho, que siento que el término se ha abaratado. El popular administrador de ventanas Enlightenment no ha lanzado más que versiones de desarrollo. La mayoría de las veces, el término se usa para describir versiones que ni siquiera son alfa o beta y si tuviera que lanzar una versión pre-alfa de una pieza de software para mantener vivo el interés en mi proyecto, probablemente así es como lo habría hecho. para etiquetarlo. (Hill, 2002)

Se trata de la versión final del producto. El mismo es liberado a los clientes. Los contratos del producto deben estar finalizados y validado, se dispara la venta del producto

Herramientas

Las herramientas tenían un rol fundamental en esta transformación, para la estructuración y seguimiento de cada producto. La selección fue la siguiente:

Aha (https://www.aha.io/)

Se trata de una herramienta para la gestión de portfolios. La herramienta facilita la gestión de las hojas de ruta de todos los productos. Permite plasmar la estrategia y la evolución de cada producto

identificando dependencias entre ellos. Consta de una funcionalidad que agregó mucho valor donde todos los usuarios pueden contribuir con sus opiniones y sugerencias en un portal de ideas, contribuyendo a una democratización de los productos dando no sólo visibilidad a quienes no participan directamente del producto sino también la posibilidad de aportar valor.

El concepto central que se tomó del software fue el Producto. De un Producto se desprendían funcionalidades. Y el conjunto de funcionalidades se agrupaba bajo el concepto de release[7].

Jira[8] (https://www.atlassian.com/es/software/jira)

Según la página oficial: "Jira Software proporciona herramientas de planificación y hojas de ruta para que los equipos puedan gestionar a los interesados, los presupuestos y los requisitos de las funciones desde el primer día." A diferencia de Aha, su uso esta más orientado al desarrollo del software, permitiendo documentar y seguir todo el ciclo. Además permitía vincular el desarrollo con las funcionalidades definidas en Aha.

Entre Aha y Jira se definió un mapeo de conceptos que permito la integración de ambos productos, dando una visión completa del producto y de su estadio actual. El concepto de Release de Aha fue mapeado con la Épica de Jira, y el Feature de Aha con la Historia de usuario en Jira.

Confluence (https://www.atlassian.com/es/software/confluence)

Es un software de colaboración en equipo. Aquí los equipos plasmaban toda la documentación de su desarrollo, incluidos documentos de las ceremonias de la metodología y el relevamiento con los usuarios.

[7] En español "lanzamiento"

ServiceNow (https://www.servicenow.com/)

Con esta herramienta se llevaba el control de todos los activos de Tecnología. Se identificaba la relación entre los diversos componentes y los impactos en negocio que generarían en caso de fallas. Fue muy útil contar con una base actualizada para realizar el análisis de impacto en etapa de migración.

Como enfoque de integración entre las herramientas Aha y Jira. Las herramientas que definían la evolución del producto y su desarrollo se optó por realizar la siguiente relación

La etapa de migración fue planificada para que una vez que los nuevos productos estuviesen en producción los clientes abandonaran la versión anterior y comenzaran a usar productos superadores. Dependiendo del abanico de productos con los que contasen los clientes se podía migrar uno a uno o de manera completa. Para identificar estos casos se segmentó la cartera de clientes en diversos grupos.
Con el objetivo de acelerar al máximo los ciclos de desarrollo se trazó una etapa de fundación. La misma consistía en el entendimiento de madurez actual y la adopción de procesos básicos que serían mandatorios para un desarrollo rápido, escalable y de calidad. Para la evaluación de madurez para la adopción de la nube se trabajó de la mano de un proveedor de primer nivel con experiencia en diversas industrias.

CAPÍTULO VI – METODOLOGÍA DE REFERENCIA

Para una transformación digital exitosa es imprescindible que se trate de un cambio que implique a toda la organización. Es un error muy común pensar que la transformación es sólo tecnológica. Se trata de un proceso de cambio global que afecta a todas las áreas, la tecnología está presente, pero no es el fin, sino un medio más para lograr la transformación.

Como se explicó en el capítulo anterior debíamos dejar de orientar nuestras iniciativas basadas en proyectos a enfocarlas por productos, y de este modo poder satisfacer las necesidades de los clientes de una manera más rápida y efectiva.

Por la naturaleza del programa las prácticas ágiles parecían ser la clave. Al tratarse de una organización nueva inmersa en una cultura existente era importante seleccionar un marco de rápida adopción y que permitiese convertir los roles de las personas existentes hacia los roles necesarios de la estructura objetivo. El punto común de los marcos más conocidos (Scrum, Kanban, XP) respecto de los métodos de ingeniería del software se basan en el desarrollo iterativo e incremental, donde los requisitos y soluciones evolucionan con el tiempo según la necesidad del proyecto. Así el trabajo es realizado mediante la colaboración de equipos auto organizados y multidisciplinarios, inmersos en un proceso compartido de toma de decisiones a corto plazo.

De todos los marcos ágiles se optó por tomar como referencia el modelo Spotify (Kniberg H & Ivarsson A. (2012). Scaling Agile @ Spotify with Tribes, Squads, Chapters & Guilds. Recuperado de: https://blog.crisp.se/wp-content/uploads/2012/11/SpotifyScaling.pdf). Sin embargo es importante tomar en cuenta la afirmación de Ben Linders (Linders, 2016) de que el modelo de Spotify puede ayudarnos a comprender cómo se hacen las cosas en Spotify, pero no debe ser copiado de manera estricta en nuestras organizaciones. La manera de hacer las cosas en Spotify cambia constantemente a medida que la gente aprende y descubre nuevas cosas. Además, no hay una única manera de realizar las cosas. Floryan (Floryan 2016), líder de equipo en Spotify, comentó que la gente suele mirar el modelo de Spotify como un ejemplo a seguir o imitar. Esto puede resultar en un sesgo de efecto halo, donde las personas miran algo que es exitoso y esperan tener éxito cuando hacen lo mismo en su empresa. Esta no es la forma más útil de ver el modelo de Spotify; imitar es un camino realmente difícil de tomar. Floryan sugirió ver el modelo de Spotify como una descripción simplificada de un sistema o proceso. El modelo puede ayudarnos a

entender cómo se hacen las cosas en Spotify, pero no es algo que debas copiar en tu propia organización. Tomando en cuenta estas advertencias el programa implicó entender como Spotify adoptó un marco ágil para luego poder adaptar dicho marco a la organización.

El modelo Spotify

Squads y Tribus

El modelo Spotify se basa en una unidad de desarrollo básica llamada **squad**, lo que en otras metodologías se conoce como scrum team. Cada equipo está diseñado para ser autónomo y poder llevar adelante el ciclo de vida completo del producto o de la funcionalidad. Son equipos autoorganizados por lo que pueden decidir su propia manera de trabajo.

A los squads se los alienta a trabajar con principios de Lean Startup (Ries 2011). El concepto de Lean Startup se basa en la idea de implementar un sistema en donde se fomenta la experimentación para la creación de nuevos productos. Lo fundamental en este método resulta ser la respuesta que se recibe por parte de los clientes (externo e internos), acerca de su experiencia con dichos productos. El primer paso es entrar en la fase de construcción tan rápido como sea posible con un producto
mínimo viable (MVP - por sus siglas en inglés Minimum Viable Product). El MVP es aquella versión del producto que permite comenzar a entender si el producto puede funcionar, con un mínimo esfuerzo y el mínimo tiempo de desarrollo. Al MVP le faltan muchos elementos que pueden ser esenciales más adelante. Es importante medir el impacto del MVP. Desde la perspectiva de desarrollo de software esto implica puestas en producción rápidas y tempranas.

Un **squad** no tiene un líder formal, pero tiene un Product Owner (PO). El product owner es responsable de priorizar el trabajo que realizará el equipo, sin embargo, no está relacionado con cómo se hará dicho trabajo. Los POs de los distintos equipos colaboran para tener un Roadmap de alto nivel documentado, y cada PO es responsable por mantener el backlog de su squad.

Cada squad también tiene acceso a un **agile coach** que ayuda a evolucionar y mejorar la manera de trabajar. Idealmente cada squad es completamente autónomo con contacto directo con los stakeholders y no tiene dependencias bloqueantes con otros squads.

Una colección de squads que trabajan en áreas relacionadas conforma una **tribu**. Si cada squad puede ser visto como una mini-startup entonces la tribu sería la "incubadora". Cada tribu tiene un líder de tribu que es el responsable de brindar el mejor ambiente posible para los squads de dicha tribu. Los squads de una tribu comparten físicamente la misma oficina, normalmente uno cerca del otro.

El tamaño de las tribus se basa en el concepto del "número de Dunbar", el cual intenta prevenir que los grupos se hagan de un tamaño tan grande que generen burocracia, reglas restrictivas, política, etc. Es por esto que el número de integrantes de las tribus es menor a 100. Las tribus comparten reuniones donde comentan sus avances, realizan demos, nuevas herramientas, etc.

Dependencias

Si bien los squads son diseñados para ser autónomos, las dependencias existen y no siempre son malas. Muchas veces los squads necesitan trabajar en conjunto para llegar a un producto completo y robusto. Sin embargo, el objetivo siempre es minimizar estas dependencias. Para lograrlo cada squad identifica las dependencias con otros squads. Luego se trata de eliminar las dependencias estrictas, sobre todo las bloqueantes. Esto muchas veces genera repriorización, reorganización, cambios arquitecturales o soluciones técnicas alternativas. La identificación de estas dependencias se hace en cada squad, y luego con reuniones bajo demanda se comparten las dependencias entre los product owners. Una dependencia común se da entre las áreas de desarrollo y operaciones, en Spotify existe un equipo diferente de operaciones, pero su trabajo no es armar los releases de los squads, su trabajo es brindarle soporte a los squads para que ellos mismos puedan realizar sus propios releases.

Capítulos y Guilds

En Spotify identificaron un potencial inconveniente que se generaba por la propia autonomía de los squads, básicamente se trataba de un caso de comunicación. Si por ejemplo una persona se enfrentaba con un problema en un squad y no consultaba con el resto de los squads, tal vez se perdiese la oportunidad de que alguien que ya hubiese pasado por ese mismo inconveniente lo ayudara, se perdía la economía de escala.

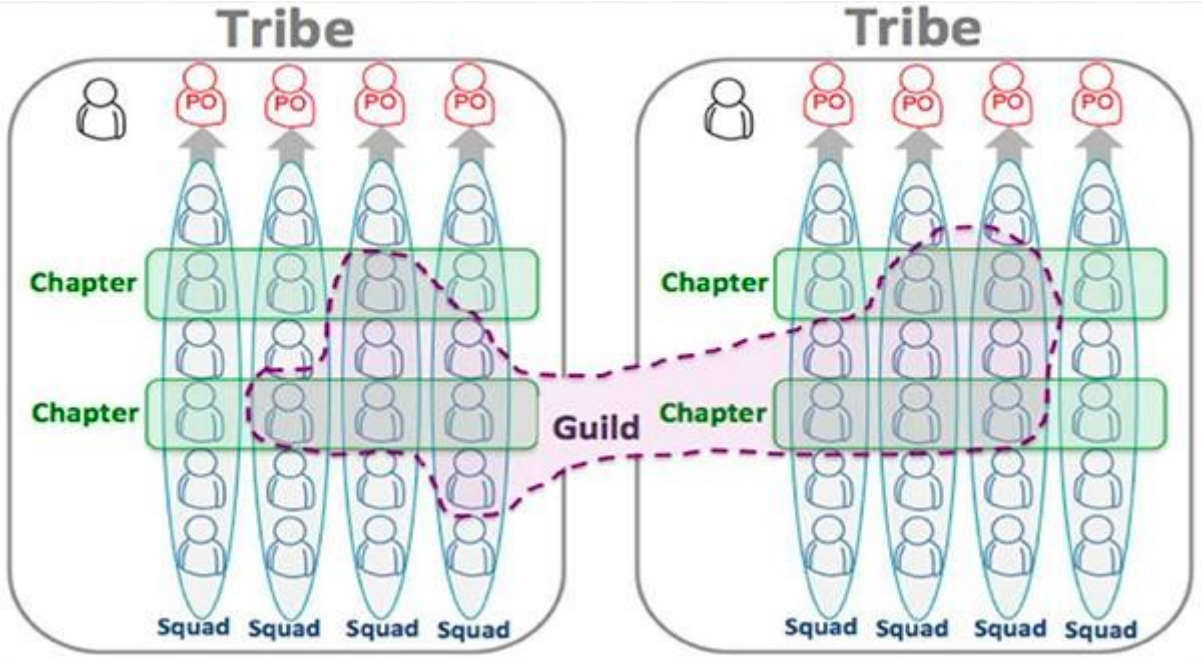
Para evitar dicho problema se crearon los guilds y los squads, con esta nueva estructura se recuperaba la economía de escala sin sacrificar la autonomía. El **capítulo** se conforma por la gente que trabaja en la misma tribu y posee las mismas áreas de competencia. Ejemplo de capítulos podrían ser el capítulo de desarrollo web, testing o backend. Cada capítulo se reúne regularmente para discutir sus experiencias y desafíos. El líder del capítulo es el gerente de los miembros del capítulo, con todas las responsabilidades tradicionales que conlleva como ser el desarrollo de las personas, establecer los salarios, etc. Sin embargo, dicho líder de capítulo también es parte de un squad que lo obliga a permanecer trabajo día a día, y por consiguiente a tener contacto con la realidad. Los miembros de los capítulos no están distribuidos uniformemente sobre los squads, algunos necesitan de más perfiles que otros.

El **Guild** es algo más orgánico, algo así como una "comunidad de interés", un grupo de personas que quieren compartir conocimiento, herramientas, código y prácticas. Los capítulos son siempre parte de una tribu, mientras que los guilds, generalmente traspasan ese límite. Ejemplos de guilds son el guild de tecnología web, el guild de testing, el de ágiles, etc.

Frecuentemente un guild incluye todos los capítulos que trabajan en ese área y sus miembros, por ejemplo el guild de testing incluye a todos los testers de todos los capítulos, pero todos aquellos interesados pueden sumarse a ese guild. Cada guild posee un coordinador de Guild.
La estructura completa propuesta puede asociarse con una estructura matricial, sin embargo la diferencia radica en que la matriz en Spotify está orientada hacia el delivery, mientras que en la mayoría de las organizaciones las personas con el mismo perfil conforman un equipo, el cual asigna a estos miembros a proyectos y reportan a un gerente funcional.

Estructura de equipos en Spotify

figura 6.1.



Nota: Estructura organizacional Spotify (2012) Henrik Kniberg & Anders Ivarsson en Spotify

Arquitectura

Una arquitectura de software puede considerarse como un modelo abstracto que describe la estructura general de un sistema, el cual se construye mediante la toma de decisiones de diseño. De hecho, puede establecerse una analogía entre el concepto de arquitectura en la construcción y en el desarrollo de software. En ambos dominios, sea el diseño de un edificio o un sistema de software, la arquitectura define la forma en que son satisfechos los requerimientos de los interesados y permite guiar la implementación y mantenimiento del producto resultante. Algunos autores han incluso sugerido que la arquitectura de un sistema es el puente que conecta los requerimientos y la implementación de dicho sistema (Garlan, 2014).

Un estilo de arquitectura es una familia de arquitecturas que comparten determinadas características. Los estilos de arquitectura no requieren el uso de tecnologías concretas, pero algunas tecnologías son más adecuadas para ciertas arquitecturas (Microsoft, 2022)[9]. Una arquitectura de

[9] Microsoft (2022). Estilos de arquitectura. https://learn.microsoft.com/es-es/azure/architecture/guide/architecture-styles/

microservicios consta de una colección de servicios autónomos y pequeños. Cada uno de los servicios es independiente y debe implementar una funcionalidad de negocio individual dentro de un contexto delimitado. Un contexto delimitado es una división natural de una empresa y proporciona un límite explícito dentro del cual existe un modelo de dominio (Microsoft, 2022)[10].

El estilo arquitectónico de Spotify está altamente orientado a microservicios, conteniendo más de 100 sistemas distintos, donde cada uno puede ser mantenido y puesto en producción de manera separada. Técnicamente cualquiera está habilitado para modificar cualquier sistema, ya que los squads se agrupan por funcionalidad. Generalmente necesitan modificar múltiples sistemas para obtener una nueva funcionalidad en producción. El riesgo de este modelo es que la arquitectura de un sistema se desordene y nadie vele por la integridad del sistema como un todo.

Para mitigar este riesgo existe un rol llamado **dueño del sistema**. Todos los sistemas tienen un dueño, o un par de ellos, se sugiere contar con un par de ellos. Para sistemas críticos, los dueños del sistema suele ser una pareja de devops, esto es una persona con la perspectiva desde el desarrollo y otra persona desde la perspectiva de operaciones.

El dueño del sistema es la persona a quien acudir para cualquier problema técnico o arquitectural relacionado a ese sistema. Es quien guía y coordina a las personas que incluirán código en su sistema para asegurarse que no habrá inconvenientes entre diversos aportes. Su foco central está en temas como calidad, documentación, deuda-técnica, estabilidad, escalabilidad y el proceso de puesta en producción.

El dueño del sistema no es un cuello de botella, puesto que él no tiene que escribir el código o tomar todas las decisiones. Típicamente es un miembro de un squad o líder de capítulo quien además tiene la responsabilidad de ser dueño del sistema. Sin embargo, de vez en cuando se toma el "día del dueño del sistema" donde realiza el trabajo específico para ese sistema. Normalmente tratan de mantener esa responsabilidad en menos de un décimo del tiempo de la persona, pero varía mucho entre diferentes sistemas.

[10] Microsoft (2022). Estilo de arquitectura de microservicios. https://learn.microsoft.com/es-es/azure/architecture/guide/architecture-styles/microservices

También poseen un rol de arquitecto, una persona que coordina el trabajo en alto nivel entre diversos sistemas. Revisa los deployments de los nuevos sistemas para asegurarse de que se hayan evitado errores comunes, y que estén alineados con la visión de arquitectura. Sus retroalimentaciones son siempre sugerencias, la decisión final del diseño del sistema siempre recae en el squad que lo construye.

Cultura

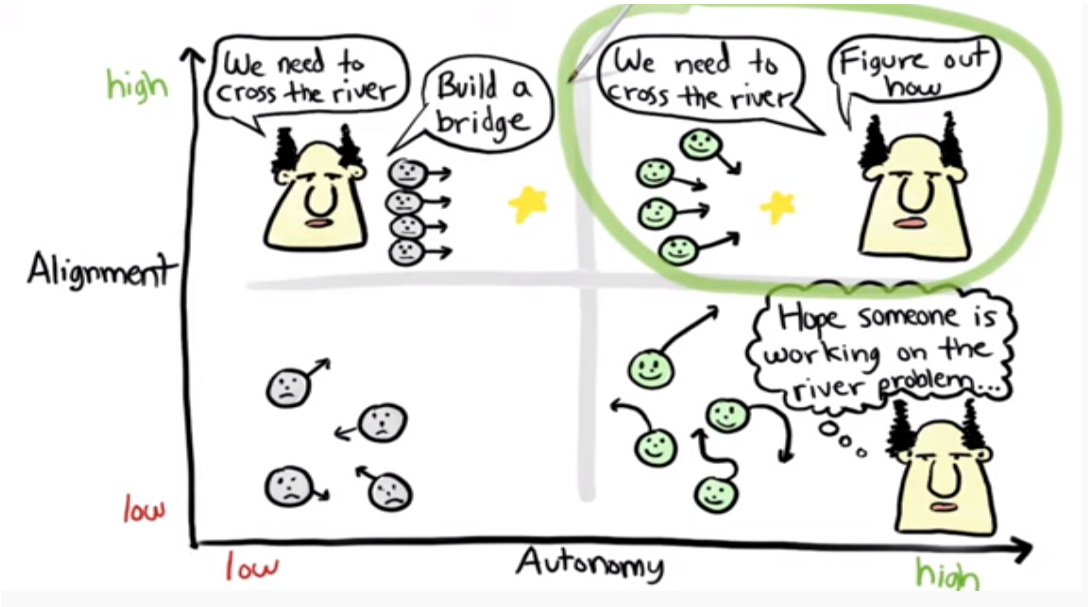
Sobre la cultura Spotify Henrik Kniberg (2017) destaca algunas de las prácticas de scrum habituales como ser las reuniones de planificación, retrospectiva, diarias, mediciones, etc. atentaban contra la agilidad buscada por lo cual decidieron dejar estas prácticas como opcionales. Destaca que las reglas son un buen comienzo, pero que deben ser rotas si es necesario. Decidieron que los principios ágiles pesan más que cualquier práctica metodológica específica. Así fue que renombraron el rol de Scrum Master a Agile Coach, marcando que querían líderes que estuvieran al servicio del equipo antes que "maestros de los procesos". Los **squads** son multidisciplinarios y autoorganizados de generalmente menos de 8 personas, se sientan juntos y tienen responsabilidad total sobre el desarrollo. Cada squad tiene una misión de largo plazo. La autonomía básicamente implica que el equipo decide que construir, cómo hacerlo y cómo trabajar juntos mientras realizan dicha construcción. Hay ciertos límites como ser la misión del squad, la estrategia de producto y los objetivos de corto plazo. Los equipos son autónomos, pero no se debe caer en la suboptimización, la misión es más importante que cualquier squad individual. Su objetivo es que los squads sean débilmente acoplados, pero estrechamente alineados. Kniberg remarca que aún no están en el estadio deseado pero que experimentan mucho para lograrlo, la experimentación es clave para madurar.

Con respecto al alineamiento y a la autonomía plantea cuatro escenarios:

1. Bajo alineamiento y baja autonomía, básicamente genera una cultura de micro-management
2. Alto alineamiento y baja autonomía, los líderes son buenos comunicando que problema debe ser resuelto, pero también le dicen a sus equipos cómo deben resolverlo
3. Alto alineamiento y alta autonomía, los líderes se centran en que problemas se deben resolver, pero alientan al equipo para que descubran la mejor manera de resolverlo

4. Bajo alineamiento y alta autonomía, significa que los equipos hacen lo que quieren, yendo hacia diferentes direcciones, los líderes no son de ayuda y el producto se vuelve confuso y difícil de mantener

Figura 6.2.



Nota. Alineamiento y Autonomía. Henrik Kniberg. (2017) Spotify Engineering Culture - Part 1.

https://youtu.be/Yvfz4HGtoPc?list=RD4GK1NDTWbkY&t=231

En lo que a alineamiento respecta en Spotify como cada equipo tiene autonomía acuñan el concepto de polinización cruzada. Cada equipo utiliza una práctica o herramienta, cuando varios equipos comienzan a utilizar la herramienta entonces se convierte en el camino de menor resistencia y esto a su vez fomenta que otros squads comiencen a adoptarla también. Esto brinda un balance deseable entre flexibilidad y consistencia. Se prefiere la polinización cruzada por sobre los estándares.

Como se mencionó anteriormente la arquitectura está basada en microservicios, diversos componentes donde cada componente tiene un squad dueño. Cada squad es dueño de diversos componentes. Para evitar dependencias, si un equipo 1 necesita un desarrollo en un sistema del equipo 2, lo ideal sería que el equipo 2 hiciera las modificaciones. Pero si éste último no tiene tiempo, o tiene otras

prioridades entonces el equipo 1 puede realizar las modificaciones bajo la supervisión del equipo 2. Tienen una cultura que fomenta la revisión de código de pares, esto mejora la calidad y, aún más importante, se comparte el conocimiento.

Un punto fundamental para lograr la autonomía es cuán fácil es la puesta en producción del código. Si la puesta en producción es difícil, entonces se tenderá a minimizar la frecuencia, lo que conlleva a que cada puesta en producción tenga más código y sea más difícil de implementar, generando así un círculo vicioso. Pero si la puesta en producción es sencilla entonces significa que se hará con mayor frecuencia, por consiguiente tendrá menos código y la implementación también será más sencilla, se genera un círculo virtuoso. Por consiguiente, se vela por una cultura de liberaciones de código pequeñas y frecuentes.

Para fomentar esta práctica velan por un sistema de autoservicio, donde cada equipo pueda poner su código en producción sin depender de nadie. Para simplificar esta tarea existen squads específicos que se dedican exclusivamente a ayudar a otros squads a que sea una tarea sencilla.
Como es una cultura basada en la confianza confían en las ventajas que provee el modelo de fallar rápido, aprender rápido y mejorar rápido. Aquí es donde es de suma importancia seguir la estrategia de MVP del método Lean Startup mencionado anteriormente.

DevSecOps

Para brindar la autonomía en el desarrollo y puesta en producción del software, apalancádonos en las herramientas y beneficios de la nube se optó por seguir el conjunto de prácticas conocidas como devsecops.

DevSecOps es un término que evolucionó desde su predecesor DevOps. De acuerdo con (Bass, Weber, & Liming, 2015) “DevOps es un conjunto de prácticas destinadas a reducir el tiempo entre el compromiso de un cambio en un sistema y el cambio que se coloca en la producción normal, al tiempo que garantiza una alta calidad”

DevOps es un proceso de desarrollo y entrega de software que se apoya en la automatización y monitoreo de los procesos de integración de software, testing, despliegue y cambios en la infraestructura. Esto se logra estableciendo un conjunto de prácticas y hábitos, junto con un entorno donde la construcción, testeo y liberación del software ocurren de manera continua, rápida, frecuente y confiable

De acuerdo a la literatura, la habilitación de DevOps requiere una serie de etapas:

1. Implementar Integración y Testing continuo (Continuous Integration, o CI), mediante la automatización de los procesos de compilación (building) y la automatización de las pruebas (testing automatizado).

2. Implementar Entrega y luego Implantación Continua (Continuous Delivery o CD y Continuous Deployment) mediante la automatización del proceso de instalación de las aplicaciones en los distintos ambientes (QA, Stress y en el caso de la Implantación Continua también en Producción). La Entrega Continua permite asegurar que las aplicaciones operarán adecuadamente en un ambiente productivo mediante un proceso automatizado de pruebas. La Implantación Continua es el paso que sigue, que permite automatizar totalmente la entrega en producción. No todas las organizaciones pueden instituir la Implantación Continua, e incluso en algunos casos puede no ser posible (por ejemplo, debido a marcos regulatorios).

3. Gestionar la configuración de los distintos ambientes, lo cual es necesario para lograr la Implantación Continua, mediante la implementación de tecnología que permita configurar y desplegar infraestructura de manera controlada (conocido también como Infrastructure as Code).

4. Monitorear y obtener retroalimentación a lo largo de toda la cadena de entrega (Continuous Monitoring), mediante herramientas de monitoreo en los ambientes productivos (alertas, análisis dinámico de logs, monitoreo de la plataforma).

DevSecOps (diminutivo de desarrollo, seguridad y operaciones, en inglés) automatiza la integración de la seguridad en todas las fases del ciclo de vida de desarrollo de software, desde el diseño inicial hasta la entrega de software, pasando por la integración, la prueba y la implementación.[11]

[11] IBM Cloud Education. 2020. ¿Qué es DevSecOps?. https://www.ibm.com/mx-es/cloud/learn/devsecops

DevSecOps representa una evolución natural y necesaria en la forma en que las organizaciones de desarrollo se acercan a la seguridad. En el pasado, un equipo de seguridad independiente añadía la seguridad al software al final del ciclo de desarrollo, y otro equipo independiente de control de calidad (QA) la probaba. DevSecOps integra estas prácticas en un único flujo de integración continua automatizado y gestionado mediante un equipo interdisciplinario.

CAPÍTULO VII – NUEVA ESTRUCTURA

Para acercarnos al modelo de referencia descrito en el capítulo anterior decidimos conformar tribus y squads. La autonomía de los squads era obligatoria, por lo que en cualquier squad debíamos poseer diferentes perfiles para que pudiesen llevar a cabo todo el ciclo de vida del producto adelante. El ciclo de vida del producto se suele separar en las etapas de introducción, crecimiento, madurez y declive. Estas fases deben ser cubiertas por las fases del ciclo de vida del desarrollo de software (SDLC).

Figura 7.1

Nota. (2002) El ciclo de desarrollo de Software SDLC

Perfiles

Comenzamos por trazar tres perfiles mandatorios:

- Una persona con formación funcional con dos responsabilidades bien definidas. Por un lado, hacia fuera del equipo de TI, interpretar las necesidades del negocio, negociar con él y tener una visión clara de la evolución del producto. Por otro lado, hacia dentro del equipo de TI, transmitir estas necesidades y ayudar la priorización y bloqueantes que pudiesen surgir. Esta figura fue identificada como Product Owner (PO).

- Un perfil de desarrollo full-stack tanto de backend como de frontend, que pudiesen crear el producto en base a los requerimientos funcionales y a las definiciones del perfil anterior.
- Finalmente, un último perfil relacionado con la infraestructura, esto es generación de ambientes principalmente en la nube y puesta en producción del código. Este perfil fue identificado como SRE

El concepto de la ingeniería de confiabilidad (SRE) surgió en el equipo de ingeniería de Google, y se le atribuye a Ben Treynor Sloss. Para desempeñarse como SRE es necesario contar con una trayectoria de desarrollo de software con experiencia adicional en operaciones, o bien de administración de sistemas o de operaciones de TI con habilidades de desarrollo de software.

Los SRE se encargan de la forma en que se implementa, configura y supervisa el código, así como de la disponibilidad, la latencia, la gestión de los cambios, la respuesta ante emergencias y la gestión de la capacidad de los servicios en la producción.

La ingeniería de confiabilidad del sitio permite que los equipos determinen las características nuevas que se pueden lanzar y en qué momento, gracias al uso de acuerdos de nivel de servicio (SLA) para definir la confiabilidad requerida del sistema mediante indicadores de nivel de servicio (SLI) y objetivos de nivel de servicio (SLO).

Para este perfil es fundamental el conocimiento de tecnologías cloud.

Adicionalmente estos equipos debían organizarse metodológicamente, por lo cual un rol de Scrum Master era necesario. Sin embargo, se pretendió evitar asignar una persona específicamente para llevar adelante estas tareas. El espíritu fue que esa persona debía utilizar las herramientas metodológicas que considerara correctas para acelerar la velocidad de producción del equipo, por tal motivo esa persona debía ser parte y nunca aislarse de las actividades relacionadas directamente con el producto.

Como último rol identificado necesario para lograr la autonomía sobre todo el ciclo de vida restaba el rol de control de calidad (QA), al igual que en el caso del Scrum Master no se quería asignar a una persona específicamente para realizar dichas tareas. Idealmente ambos roles serían responsabilidad del PO, su fundamento para ser Scrum Master estaba dado en que la persona con la visión global del producto

sería el ideal para detectar las ceremonias ágiles necesarias para maximizar la productividad, y por el lado de QA al ser quien interpretaba las necesidad del negocio y quien diese el visto bueno del entregable entonces si el resultado no era el esperable entonces el inconveniente se había dado en la comunicación o el entendimiento entre el negocio y el PO.

Finalmente un perfil adicional era el de Gerente de Producto, PM por sus siglas en inglés -Product Manager-. Las funciones primordiales de este rol consistían en definir la estrategia del producto, definir y mantener el roadmap del producto, definir los requerimientos del producto y garantizar la calidad del producto a través del testing y las pruebas de usabilidad. Este perfil debía ser externo al área de Tecnología, pero trabajar de una manera sumamente alineada con el PO.

Adicionalmente se identificaron como necesarios roles de arquitectura, seguridad, service delivery manager y un portfolio manager. Estos conforman un squad adicional para brindar soporte y lineamientos al resto. Sería el único squad que no estaría asignado a una única iniciativa.

Descripción de puestos y roles

Product Manager

Cada producto debía contar con un Product Manager (PM). Éste sería el encargado de gestionar el producto con el fin de hacerlo más atractivo y funcional. Entre sus tareas más relevantes figuraban:

Elaborar el plan de marketing

Generar ideas de productos

Estudiar y entender necesidades de los usuarios

Definir la visión de un producto

Realizar análisis de resultados o ventas

El PM definiría la dirección del producto a través de la investigación, el establecimiento de una

visión, la alineación y la priorización.

Product Owner

Un Product Owner es el profesional encargado de optimizar el valor del producto desarrollado por una empresa. Traducido al castellano como propietario de producto y mencionado a menudo con las siglas PO, el Product Owner debe asegurarse de que el equipo con el que trabaja aporta auténtico valor al proyecto, encargándose de gestionar las tareas del backlog

Ser el frente de conversación con el PM

Entregar las estimaciones de nuevas funcionalidades

Planificar fechas de entrega

Responsable técnico del producto (seguridad, compliance, etc)

Tribe leader

El líder de Tribu es el responsable de organizar todos los squads bajo su responsabilidad maximizando el valor generado. Es quien posee la visión tecnológica completa de la familia de productos de su tribu. Debe velar por las sinergias producto de la reutilización de componentes y compartirlos frente a sus pares para que esta sinergia puede extenderse más allá de su propia tribu.

Developer Full-Stack

Es un desarrollador que trabaja tanto en el front-end como en el back-end de un sitio web, software o aplicación, es decir: está tanto en la parte del cliente como del servidor. Para entender qué es el full stack es preciso diferenciar primero el front-end del back-end.

Front-end

Es la parte de un sitio web con la que el usuario interactúa de forma directa, por lo que incluye el texto, imágenes, vídeos, gráficos, call to action, FAQ (preguntas frecuentes), formularios para registrarse, etc. Un programador de front-end deberá tener en cuenta todos aquellos aspectos que mejoren la experiencia de usuario para crear interfaces intuitivas y rápidas, como la selección de colores, disposición de elementos, botones, menús, urls, etc. Para ello, deberá combinar un buen diseño con la creatividad y la funcionalidad.

Back-end

Incluye toda la información del servidor y base de datos que hacen posible que una aplicación web funcione y a la que no tiene acceso el usuario. Por lo tanto, un desarrollador de back-end deberá contar con conocimientos de lenguaje de programación, además de manejar bases de datos y herramientas de almacenamiento web. Saber trabajar con lógica y manejar conceptos abstractos son dos claves de este perfil.

Es decir, los objetivos y funciones de un desarrollador del front-end y del back-end son diferentes pero complementarios. La figura del full stack developer asumiría ambos roles teniendo una visión completa de un proyecto/negocio

Así, las funciones del full stack developer combinan las propias del front-end y back-end, por lo que estarían:

- Desarrollo de la estructura y arquitectura web: para organizar la información de la forma más eficiente posible.
- Diseño de la interfaz y parte visual.
- Facilitar la navegación y mejorar la experiencia de usuario.
- Asegurar que la web se visualiza bien en diferentes navegadores y dispositivos.
- Traducir el diseño a código HTML y CSS.
- Garantizar la conectividad de la web.
- Configurar y mantener los servidores.
- Manejar las librerías del servidor.
- Automatizar tareas de desarrollo de software.
- Optimizar las aplicaciones web y que sean rápidas.
- Crear y gestionar bases de datos eficientes y conectarlas con las soluciones.
- Configurar la seguridad y prevenir ataques.

SRE[12] - Ingeniero de confiabilidad del sitio

Los equipos de SRE se encargan de la forma en que se implementa, configura y supervisa el código, así como de la disponibilidad, la latencia, la gestión de los cambios, la respuesta ante emergencias y la gestión de la capacidad de los servicios en la producción.

La ingeniería de confiabilidad del sitio permite que los equipos determinen las características nuevas que se pueden lanzar y en qué momento, gracias al uso de acuerdos de nivel de servicio (SLA) para definir la confiabilidad requerida del sistema mediante indicadores de nivel de servicio (SLI) y objetivos de nivel de servicio (SLO).

Los SLI son medidas definidas de aspectos específicos sobre los niveles de servicio que se proporcionan. Los más importantes son la latencia de las solicitudes, la disponibilidad, la tasa de errores y el rendimiento del sistema. Los SLO se basan en el valor o el rango objetivo para un nivel de servicio específico basado en el SLI.

El objetivo de nivel de servicio se determina en función del tiempo de inactividad que se acordó como aceptable, denominado "estimación de errores", que representa el límite máximo de interrupciones y errores permitidos.

Con la SRE no se espera una confiabilidad al 100 %, sino que se planifican y se aceptan las fallas.

Es posible que el equipo de desarrollo alcance el límite de la estimación de errores al lanzar una característica nueva. Pero al utilizar este recurso junto con el SLO, puede determinar si debe lanzar un producto o un servicio según el margen de error disponible.

Si un servicio está dentro de los parámetros de estimación de errores permitidos, el equipo de desarrollo puede lanzarlo cuando lo desee. Sin embargo, si el sistema tiene demasiados errores o interrupciones más prolongadas de lo que se prevé en la estimación, no se podrán realizar lanzamientos nuevos hasta que los errores estén dentro de los parámetros.

[12] https://www.redhat.com/es/topics/devops/what-is-sre

El equipo de desarrollo realiza pruebas automatizadas de las operaciones para demostrar la confiabilidad.

Los ingenieros de confiabilidad del sitio dividen el trabajo entre las tareas operativas y las del proyecto. Según las prácticas recomendadas de Google para los SRE, pueden dedicar hasta un 50 % del tiempo a las operaciones y deben estar bajo supervisión para no excederse.

El resto del tiempo deben dedicarlo a las tareas de desarrollo, como crear funciones nuevas, ampliar el sistema e implementar la automatización.

El equipo de desarrollo puede ocuparse del resto del trabajo operativo y de los servicios con bajo rendimiento para evitar que los ingenieros inviertan demasiado tiempo en las operaciones de una aplicación o un servicio.

La automatización es una parte importante del trabajo de los ingenieros de confiabilidad del sitio. Si deben resolver un problema varias veces, deben automatizar la solución. Así también se garantiza que las tareas operativas ocupen solo la mitad de su carga de trabajo. Mantener el equilibrio entre las operaciones y la labor de desarrollo es un elemento clave de la SRE.

Para ser ingeniero de confiabilidad del sitio, es necesario contar con una trayectoria de desarrollo de software con experiencia adicional en operaciones, o bien de administración de sistemas o de operaciones de TI con habilidades de desarrollo de software.

QA[13]

Un QA o analista QA es el profesional responsable de asegurar la calidad del software y de prevenir fallos en él. Es la persona encargada de garantizar el correcto funcionamiento del producto desde el primer momento, pero también de confirmar que satisface las expectativas de los usuarios que lo utilizarán.

[13] Del ingles Quality Assurance

Como su propio nombre indica, es un rol vinculado a la calidad. Y lo es en un amplio sentido. Así, un QA trabaja en conseguir tanto la calidad del producto como del proceso, esforzándose para que todas las fases del desarrollo de la aplicación se lleven a cabo eficientemente.

Para poder garantizar la calidad del software y el cumplimiento de las expectativas de los usuarios, un analista QA realiza las siguientes funciones:

- Participar en la definición del producto.
- Analizar los requerimientos.
- Plantear la estrategia de pruebas.
- Diseñar los escenarios y casos de prueba.
- Revisar, planificar, preparar y ejecutar las pruebas.
- Reportar errores, plantear soluciones y validar la corrección.
- Automatizar pruebas, monitorearlas y mantenerlas.
- Impulsar mejoras en los procesos.
- Simular la ejecución de productos y evaluar su rendimiento.

Scrum Master[14]

El Scrum Master tiene dos funciones principales dentro del marco de trabajo: gestionar el proceso Scrum y ayudar a eliminar impedimentos que puedan afectar a la entrega del producto. Además, se encarga de las labores de mentoring y formación, coaching y de facilitar reuniones y eventos si es necesario.

1. Gestionar el proceso Scrum: el Scrum Master se encarga de gestionar y asegurar que el proceso Scrum se lleva a cabo correctamente, así como de facilitar la ejecución del proceso y sus mecánicas. Siempre atendiendo a los tres pilares del control empírico de procesos y haciendo que la metodología sea una fuente de generación de valor.

2. Eliminar impedimentos: esta función del Scrum Master indica la necesidad de ayudar a eliminar progresiva y constantemente impedimentos que van surgiendo en la organización y que afectan a su capacidad para entregar valor, así como a la integridad de esta metodología. El Scrum Master debe ser el

[14] https://www2.deloitte.com/es/es/pages/technology/articles/roles-y-responsabilidades-scrum.html

responsable de velar porque Scrum se lleve adelante, transmitiendo sus beneficios a la organización facilitando su implementación.

Arquitecto

El Arquitecto debía ser la guía y referencia para todos los desarrollos identificando soluciones de software que beneficiaran a los equipos. Debía fomentar el talento dentro de los squads y establecer una cultura de mejora continua, para que cada equipo pudiese presentar los diseños de sus propias soluciones.

Asimismo, debía ofrecer soporte a procesos globales implicados en la evolución del software, por ejemplo debía participar del proceso de revisión de arquitectura global examinando y comprobando la evidencia del código y la infraestructura del proyecto previo a la defensa del proyecto.

El Arquitecto tendría que colaborar estrechamente con compañeros y usuarios finales de otras partes del negocio, por lo que era imprescindible contar con habilidades de comunicación. Estas habilidades también eran útiles para presentar conceptos e informar de resultados a miembros de la dirección.

El Arquitectos de Software además tendría bajo su responsabilidad el análisis de problemas y la sugerencia de mejoras, para así minimizar las demoras por deficiencias en el diseño de los productos. También debía poseer sólidas habilidades de liderazgo, ya que no tendría autoridad formal sobre el resto de los squads.

Program Manager

Un program manager gestiona y alinea proyectos complejos a largo plazo con los objetivos más ambiciosos de la empresa. También coordina los equipos de proyectos y las necesidades presupuestarias. Roles y responsabilidades del program manager.

Si bien los roles de un program manager y de un project manager son diferentes, comparten algunas responsabilidades, como la supervisión de los proyectos y la coordinación de las comunicaciones. Los roles y responsabilidades en el ámbito de la gestión de programas varían de una empresa a otra, pero

algunas de las funciones básicas incluyen la estrategia del proyecto, el control de calidad, la emisión de los informes de desempeño, la gestión de cambios y la gestión de riesgos.

Un buen candidato tendrá todas estas habilidades y mucho más, como ser un buen líder con la capacidad de coordinar varios equipos. A continuación, te mostramos algunos de los roles y responsabilidades adicionales de un program manager.

Implementar estrategias

Los program managers tienen la responsabilidad de coordinar proyectos con objetivos establecidos que apoyen las metas de toda la empresa. Esto significa trabajar con los project managers para garantizar que se cumplan los objetivos diarios del proyecto y que los programas generales estén encaminados.

Los program managers también optimizan los procesos internos y definen las mejores prácticas para un conjunto de proyectos. Esto permite que los project managers tengan una orientación clara para gestionar los proyectos. También ayuda a evitar que se cometan errores debido a la falta de comunicación y al tiempo dedicado al trabajo.

Otras tareas estratégicas incluyen la coordinación de proyectos con el project manager para garantizar que los entregables estén alineados.

Si bien los project managers tienden a centrarse en la colaboración para proyectos puntuales, los program managers se aseguran de que la colaboración sea entre departamentos. Y aunque es un gran desafío, también es una de las partes más gratificantes del trabajo.

Los program managers también tienden a centrarse en crear hojas de ruta de programas para los objetivos de alto nivel de la empresa. Esto varía ligeramente de un project manager que creará hojas de rutas de proyectos individuales.

El program manager puede gestionar la colaboración mediante herramientas nuevas que optimizan la comunicación, como un software de gestión de programas o juegos para fortalecer el espíritu de los equipos (teambuilding). Si bien son métodos diferentes, tienen como objetivo promover el trabajo en equipo y aumentar la visibilidad.

Analista de Seguridad

Es el profesional responsable de planificar, evaluar y proteger la red y los sistemas de una empresa de cualquier ciberataque.

Los analistas de seguridad cibernética pueden trabajar en varias industrias, por lo tanto, sus funciones diarias pueden variar. En general, las tareas que realizan estos expertos se pueden organizar en cinco categorías:

1. Supervisión de redes

Consiste en monitorear regularmente la seguridad de la red de la empresa para estar al tanto de la actividad que se lleva a cabo y detectar ataques o accesos no autorizados e identificar las vulnerabilidades de la red.

2. Administración de software

Se trata de la instalación, administración y actualización de softwares de seguridad como como firewalls y encriptaciones. La misión es asegurarse de que el software cuente con todas las medidas adecuadas de ciberseguridad.

3. Pruebas de seguridad informática

El analista de ciberseguridad realiza pruebas de vulnerabilidad y análisis de riesgos con el objetivo de evaluar la seguridad actual de los sistemas de una empresa e identificar puntos de mejora para mayor protección. Estas pruebas de vulnerabilidad se conocen como pruebas de penetración, pen test o pentesting.

4. Investigación de tendencias de seguridad

Es fundamental que este experto en ciberseguridad se mantenga actualizado sobre tendencias, avances y mejoras tanto en los métodos de ciberataques como en las prácticas en ciberseguridad.

5. Informar hallazgos y desarrollar planes de ciberseguridad

Una tarea clave en el trabajo de un analista cibernético es comunicarse con la empresa cliente y presentarle soluciones y/o recomendaciones como, por ejemplo:

Actualizar el plan de recuperación en caso de ataques.

Asegurar copias de seguridad.

Desarrollar e implementar mejores prácticas de ciberseguridad.

Discutir las debilidades y mejoras de los sistemas de la compañía.

Informar las causas de las brechas de seguridad.

Plan de carrera

A diferencia de cómo estaba planteado el desarrollo profesional en el pasado, a partir de esta transformación el plan de carrera no obligaba a un cambio de rol, si no que se permitiría que cada cual pueda desarrollarse en la rama que más le agradara sin contar con un incentivo económico para cambiar de rama.

Equipos

Con el ánimo de generar una comunicación más eficaz y efectiva, y empoderar a los equipos la topología elegida fue un modelo holocrático. Básicamente se persiguió una estructura menos verticalista, minimizando los escalones de reportes jerárquicos. Cada squad tendría un Product Owner, y todos los product owners reportarían al líder del programa. Para agilizar la comunicación existían los líderes de tribu, sin embargo los POs no reportaban directamente a ellos, si no que fungían un rol de facilitador a sus pares.

Tamaño

Una vez decidida la columna vertebral de los equipos entonces fue necesario definir el tamaño adecuado de cada uno. Comenzamos por guiarnos por la regla acuñada por Jeff Bezos de two-pizza teams, donde un squad debe tener un número de integrantes que pueda ser alimentado con dos pizzas. Con esto en mente la selección de tamaño fue de entre 4 y 8 integrantes. La cantidad de ajuste de los equipos se daba en los desarrolladores. como mínimo cada equipo constaba de un PO, un SRE y dos Developers. En el caso de ser solo dos developers el seniority debía ser alto, idealmente dos seniors.

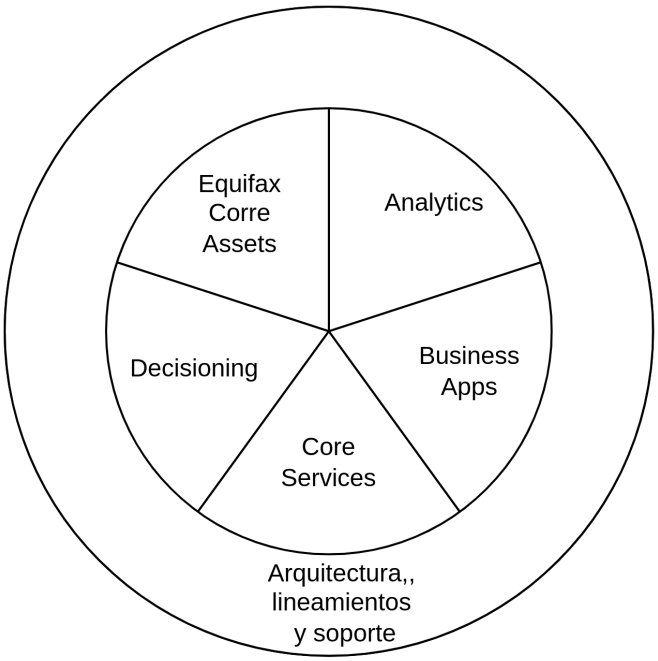
Tribus

Con base en el modelo de Spotify descrito en el capítulo anterior. Nos organizamos en tribus. Con el total de tribus se abarcaban todos los productos y las funcionalidades que permitían la operación de la empresa.

Las tribus definidas fueron:

- Analytics. Relacionada con analítica avanzada, una de las core capabilities de la empresa
- Business Apps. Aplicaciones de Negocio, todos aquellos productos con acceso directo por parte de los clientes, tanto internos como externos.
- Core Apps. Aplicaciones Core, aquellos componentes tecnológicos comunes a todos los productos como ser autenticación y autorización, logging, facturación, etc
- Decisioning. Aplicaciones relacionadas con el producto de mayor facturación en Equifax Argentina, Paraguay y Uruguay. Se relacionaba con sistemas expertos basados en reglas. Cada país tenía diferentes implementaciones y debían migrar hacia una plataforma corporativa. Por complejidad se decidió separarlo de Business Apps.
- Equifax Core Assets. Relacionadas con el ciclo de vida de los datos. Nuevamente cada país contaba con desarrollos a medida y debíamos adoptar una nueva plataforma corporativa común a todos los países de Equifax.
- Arquitectura, lineamientos y soporte. Relacionada con temas de arquitectura de solución y seguridad, gestión y ejecución del programa.

Figura 7.2



Nota. Tribus definidas (2022), fuente propia.

En cada tribu se asignaría un "líder de tribu", quien actuaba como un único foco de conocimiento sobre el evolutivo de la familia de productos. Ese líder no dedicaba el 100% de su tiempo a este rol, si no que se trataba de un PO de un squad perteneciente a esa tribu.

Squads

Dentro de cada tribu se asignaría una cantidad inicial de squads. Los squads serían la unidad mínima para escalar en un futuro. Si por ejemplo en algún momento del programa surgía la necesidad de desarrollar un nuevo producto, entonces se generaría al menos un squad más para atenderlo. De aquí el hecho fundamental de que los squads contemplasen todos los roles necesarios para llevar adelante una iniciativa. Esto siguiendo el principio de "you build it, you run it".

Chapters

Se definieron 5 capítulos con el fin de que con el fin de lograr coherencia y sinergias entre los distintos squads. Estos capítulos no contarían con un líder formal definido. Cada uno de estos capítulos

dictaría las buenas prácticas en base a las experiencias que fuesen transcurriendo. Sería una evolución constante. Utilización de nuevas tecnologías, modificación o creación de procesos, adopción de buenas prácticas serían todos alcances de cada capítulo.

Estos capítulos sólo contarían con gente del mismo perfil, aunque no necesariamente con la misma experiencia. Los capítulos identificados fueron:

- SRE
- Front end
- Back end
- PO
- Seguridad

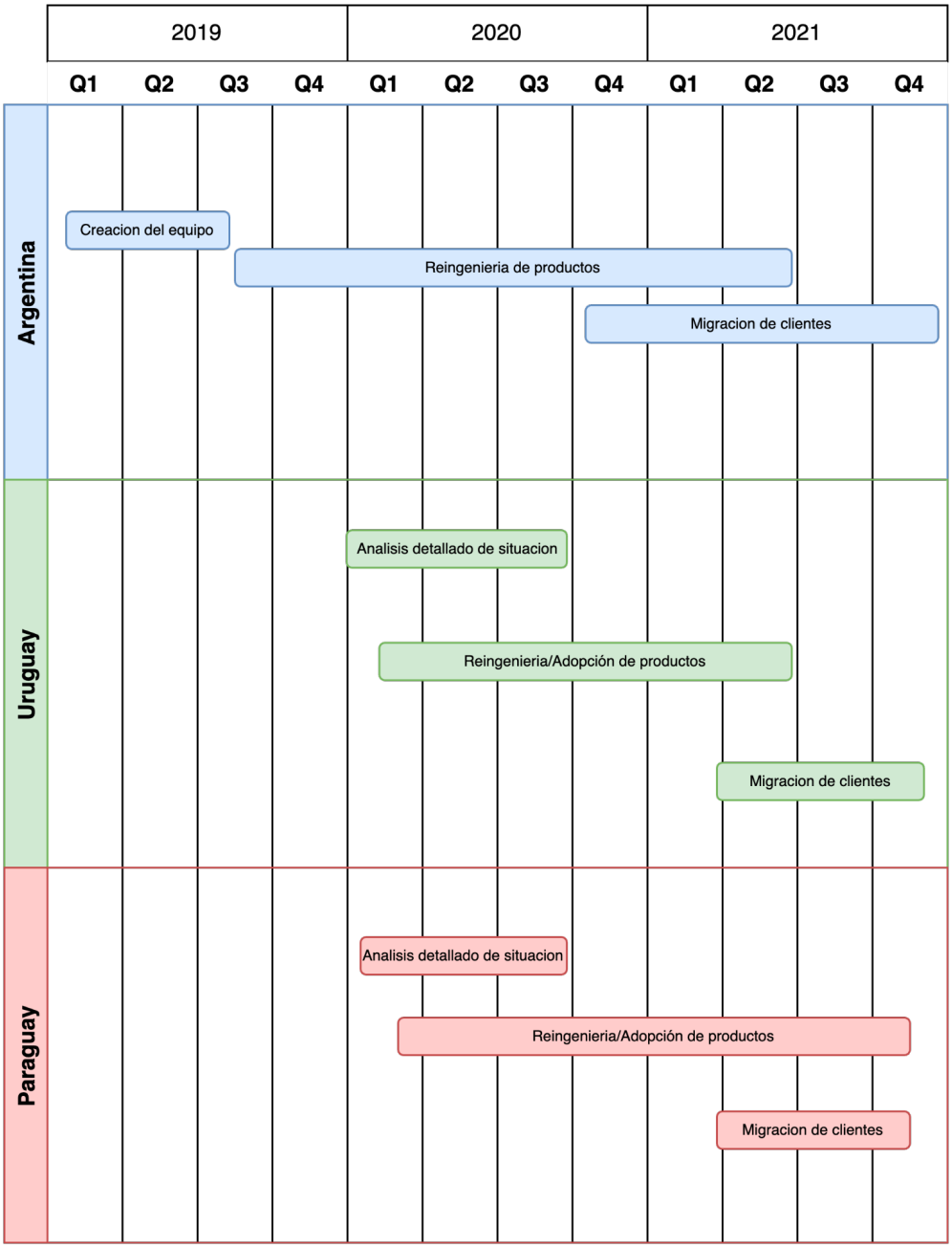
COEs (Marciniak, 2012)

Globalmente Equifax había conformado dos centros de excelencia centrales para ayudar a las regiones a desarrollar y adoptar la tecnología de la nube de manera segura. Dichos COEs eran el Cloud COE y el COE de Seguridad. Ambos centros marcaban lineamientos y buenas prácticas para que las distintas regiones las adoptaran con el fin de agilizar el desarrollo y la puesta en producción de los productos. Estos equipos no participaban en el día a día del desarrollo, sin embargo eran aprobadores de toda iniciativa siendo el último eslabón del proceso de puesta en producción, lo que significaba que si no se estaba alineado con sus estándares entonces no era factible obtener la aprobación necesaria y por consiguiente no se podría lanzar o evolucionar el producto.

CAPÍTULO VIII – IMPLEMENTACIÓN

Línea de tiempo del programa

Figura 8.1



Nota. Cronograma de alto nivel del programa (2022), fuente propia.

Alineación organizacional

El cambio debe nacer y venir impulsado desde el C-Level (topdown[15]), pero la responsabilidad debe ser compartida de forma social y colaborativa con todas las personas de la organización, permitiendo también una transformación digital bottom-up[16]. Con esta premisa en mente varios meses antes de comenzar la ejecución del programa se dispararon una serie de reuniones con todos los stakeholders identificados para comentar el alcance, alinear expectativas y obtener retroalimentación de todos ellos, fuese para confirmar el plan o para planificar y poder atacar todos los hallazgos. Estas reuniones fueron fundamentales para que cada área pudiera darle la importancia necesaria al proyecto, esto ayudó a que el objetivo de transformación digital fuera volcado a diversos objetivos en todas las áreas.

Se detectaron áreas con mayor resistencia que otras, para aquellas más resistentes fue necesario intensificar las juntas para transmitir con mayor claridad el alcance y permitir que fuesen y se sintieran parte fundamental de la toma de decisiones respecto del programa. Por mencionar un ejemplo el feedback del área comercial sirvió para identificar el período donde los clientes podrían ser migrados como así también detectar un camino adicional al programa donde todos los clientes debían ser notificados de que los datos pasarían a estar en la nube.

Distribución en tribus y squads

Según la planificación realizada se necesitaba de aproximadamente 80 personas, de las cuales 15 personas eran internas y las 65 restantes debían ser contratadas a través de proveedores. Esto se debía a ciertas restricciones de contratación que impedían aumentar el número de empleados efectivos. Además, los procesos internos de contratación eran lentos y esto atentaba contra la necesidad de escalar los equipos rápidamente. Por estos factores se decidió ir por un esquema basado en contrataciones mediante diversos proveedores. Esto ayudaba la velocidad de conformación de equipos, brindando una mayor celeridad mucho mayor que la conformación de equipos compuestos solo por gente interna, por experiencias anteriores un proveedor podía contratar gente en aproximadamente 2 semanas desde la aprobación de la persona, mientras que internamente estábamos en un tiempo aproximado de 2 meses. Para estas contrataciones trabajamos con 5 consultoras, focalizando las búsquedas en donde cada consultora tenía

[15] En español “de arriba hacia abajo”
[16] En español “de abajo hacia arriba”

mayor experiencia. De este modo separamos dos consultoras para perfiles de administradores de infraestructura, dos para perfiles de desarrollo y una exclusivamente para gestión de proyecto.

Era de suma importancia conservar el conocimiento internamente con lo cual se tenía el acuerdo de que una vez finalizado el proyecto se podrían efectivizar a las personas claves dentro de la nueva estructura. El número definitivo se definiría en un futuro ya que a partir de la demanda se adaptaría la estructura. Nunca habría menos de un squad por tribu, a menos que se decidiera eliminar toda la familia de productos.

La distribución de personas quedó definida en 6 tribus y 9 squads, conformando la siguiente estructura:

Cuadro 8.1

Tribu	Squads	Miembros por squad
Analytics	1	6
Aplicaciones de negocio Apps	4	6
Aplicaciones Centrales	2	6
Decision	2	7
Datos	2	8
Arquitectura, lineamientos y soporte	1	8

Nota. Tribus, Squads y cantidad de miembros (2022), fuente propia.

Arquitectura, lineamientos y soporte

Con el objetivo de homogeneizar rápidamente las maneras de trabajar y garantizar que el trabajo fuese coherente entre las diversas áreas, decidimos sacrificar el concepto de polinización cruzada y conformar un equipo que pudiese definir los estándares y las buenas prácticas a adoptar. En ese equipo se incorporó a un representante de cada proveedor así había un alineamiento preciso en todos los aspectos. Este equipo era el encargado de brindar las arquitecturas de alto nivel, es decir se debían construir ciertos componentes, pero cómo hacerlos dependía de cada equipo.

Perfiles

Uno de los objetivos una vez finalizado el programa era conservar el conocimiento dentro de la compañía. Los roles más estratégicos debían ser contrataciones directas. Se tomó parte del staff que se dedicaba a proyectos, ya que ningún proyecto nuevo sería desarrollado en los sistemas legados. Si no que cualquier nueva funcionalidad debía ser incorporada en la nueva versión del producto, nuevamente sería una ventaja adicional para que los clientes optaran por migrar al nuevo producto.

Para aquellos perfiles que fueron tercerizados se tomó como base de búsqueda de candidatos las descripciones de perfiles profesionales sugeridos por la Cámara de la industria Argentina del Software (Cessi). En el anexo se pueden encontrar las mismas. En todos los casos se asumió que los candidatos debían tener experiencia en las tecnologías requeridas, pero que debían continuar su formación hacia el rol objetivo.

Product Manager

Este perfil debía ser externo al área de tecnología. Idealmente cada PM tendría un único producto asignado, y dicho producto al menos un squad dedicado.

Para cubrir este rol se optó por asignar a las personas que se desempeñaban como ingenieros de producto, a excepción de los productos relacionados con datos a los cuales se les asignó un PM proveniente del área de Analítica y Operaciones.

A todos ellos se les brindó una capacitación interna para explicar su nuevo rol dentro del proyecto. Se hizo un fuerte hincapié en que debían ser los dueños del producto y gestionar la relación con toda la empresa con relación a su producto, esto incluía la gestión de la relación con sus pares Product Managers, como así también gestionar la relación con el equipo de desarrollo a través de su contacto con el PO.

Product Owner

Este rol fue cubierto por gente de la plantilla de Equifax del equipo de tecnología. Se asignaron tanto Business Analysts como Project Managers. Todos ellos eran de carácter funcional, pero con pasado técnico lo que les facilitó el diálogo con el resto del equipo.

Al igual que con los Projects Managers la capacitación fue interna, se destacó la importancia de mantener una comunicación muy fluida con el Product Manager, y que ningún pedido debía ser atendido si no era por pedido explícito y de mutuo acuerdo con el PM.

Developer Full-stack

Internamente había muy pocas personas con conocimientos profundos sobre las nuevas tecnologías nube, y siendo que la gran mayoría de los colaboradores debían ser desarrolladores se decidió contratar a los desarrolladores a través de los proveedores. No era un perfil tan crítico en cuanto al conocimiento de la compañía, pero sí en cuanto a sus habilidades duras.

A cada consultora se le otorgó un número de vacantes a cubrir, esta distribución fue uniforme. Las entrevistas iniciales las debían realizar ellas, pero nadie ingresaría si no había una última validación de alguien de Equifax. No era necesario que fuesen desarrolladores full-stack, pero si debían conocer back o front y estar dispuestos a aprender el complemento. La descripción de puestos compartida con las consultoras para encontrar candidatos fue “Desarrollador de software”, tomada de la CESSI

SRE

En el caso de los ingenieros de fiabilidad de sitios, internamente no había gente capaz de cubrir el rol. Sin embargo, en el mercado también era difícil puesto que la tecnología de GCP (Google Cloud Platform) no estaba tan extendida en la Argentina. En general se valoraba a aquellos candidatos con

conocimiento en nube, de los cuales la gran mayoría mencionaba a aws (Amazon Web Services). Sin embargo, la gran mayoría de los candidatos acercados no tenía conocimiento en nube, si no un pasado como sysadmin o administradores de aplicaciones web. En todos los casos se brindarían capacitaciones en las nuevas tecnologías. Algo que resultó muy atractivo para los candidatos.

El perfil base para la búsqueda fue el descripto por la CESSI bajo en nombre de "Administrador de Redes, Comunicaciones y Sistemas Operativos". Se solicitó conocimiento en linux o unix como mandatorio.

QA

Los controles de aseguramiento de la calidad no estaban maduros en Equifax Argentina, de hecho, en muchos casos no existía una práctica formal adoptada. Este hecho condicionaba la elección de personas internas para marcar el camino. Por tal motivo los POs no estaban en condiciones de tomar este rol, así que -con el objetivo de que en un futuro si lo hicieran- se decidió que este rol fuese cubierto por los desarrolladores. Debía haber al menos uno por squad.

La descripción de puesto tomada fue "Analista Tester".

Scrum master

Las metodologías ágiles no estaban maduras y si bien las herramientas estaban en uso desde unos años atrás, las ceremonias de Scrum no eran algo habitual. Al igual que en el caso de QA la idea era que las funciones las llevase a cabo el PO. El racional era que no se alejaran del día a día para asi ser capaces de seleccionar las herramientas necesarias para acelerar el desarrollo y la puesta en producción, y evitar el inconveniente de sumar tareas que no tuvieran impacto beneficioso para el equipo. No debían percibir estas funciones como una carga adicional de trabajo, si no que debían notar que era una función de ayuda.

La elección fue que este rol fuera "flotante" y que todos los integrantes del equipo supiesen que en algún momento iban a tener que llevar a cabo este rol. De este modo se buscaba minimizar la resistencia y que todos colaboraran para madurar las prácticas.

En este caso no se realizó una búsqueda específica para este perfil, puesto que como se mencionó

antes sería cubierto por otra posición del equipo. A todos los candidatos se les consultó por su experiencia de trabajo con metodologías ágiles. Aquellos con mayor experiencia y voluntad de llevar adelante el rol serían los primeros Scrum Masters designados.

Tribe leader

El líder de tribu fue un lugar que se reservó para un PO de un squad perteneciente a esa tribu. Este rol fue ocupado por el PO de mayor experiencia. Su objetivo principal era mantener la coherencia de los desarrollos de los productos relacionados y buscar sinergias. Nunca fue pensado como un rol de tiempo completo.

Analista de seguridad

Como se mencionó anteriormente si bien existía un COE de seguridad, consideramos necesario tener un interlocutor único con dicho área que pudiese ir guiando al resto de los equipos para facilitar la comunicación y el entendimiento de los procesos que dicho COE dictaba. Por este motivo se contrató una única persona que pertenecería al squad de lineamientos, el único squad que no estaba asociado directamente a un producto, para poder guiar al resto de los equipos. La seguridad era responsabilidad de cada squad, la aprobación de los procesos responsabilidad del PO. Pero todos se apoyarían en esta persona.

La descripción de puesto que más se acercaba a nuestra búsqueda fue “Especialista en seguridad de la información”.

Arquitecto

Del mismo modo había un arquitecto en el equipo de lineamiento para facilitar y validar las soluciones que presentase cada equipo. Análogo al caso del analista de seguridad, las soluciones serían sugeridas por el squad y validadas por el arquitecto central. Dado que el conocimiento más valorado de esta persona era técnico, pero debía conocer componentes existentes, se solicitó el traspaso de una persona contratada que estaba trabajando en la compañía desde hacía varios años.

Program Manager

El rol de program manager era un rol de vital importancia puesto que debía ser quien coordinase e informara del avance del proyecto hacia la corporación. Para llevar a cabo esta tarea debía ser el responsable de definir el governance de seguimiento del proyecto. Implicaba detectar dependencias que

detectaran e informaran los POs, informar el avance hacia el board local y global.

La búsqueda fue realizada tomando como base la descripción de Project Manager de la CESSI, solicitando que hubiesen trabajado en más de un proyecto a la vez.

Responsabilidades

Para ordenar los roles y las responsabilidades que cada miembro del programa debía seguir se realizó una matriz RACI, en acuerdo con todas las unidades de negocio. Una matriz RACI define las funciones y responsabilidades de cada individuo frente a las actividades de un proceso. En la matriz se identifican los siguientes roles:

Responsable: el papel de quien trabaja una actividad.

Alto responsable o el que rinde cuentas: el papel del responsable final por el resultado de la decisión o acción.

Consultado: el papel de quien está involucrado antes de la decisión o acción.

Informado: el papel de aquellos que necesitan para tomar nota de la decisión o acción.

La matriz final fue el producto de una evolución constante, en acuerdo entre todas las áreas. Cada vez que se ajustaba la matriz, ésta era compartida internamente entre el equipo del programa, ya que la comunicación con las áreas externas no se veía afectada, el canal de entrada siempre fue el PM. A continuación, se muestran los roles de los miembros de los equipos y sus funciones referente a los principales procesos, y su herramienta, para el seguimiento y la ejecución de un proyecto de software.

Cuadro 8.2

	Roadmapping Aha	Planificación Jira	ITSM ServiceNow	Documentación Confluence
Product Manager	A	I	I	I
Product Owner	C	R	A	R
Developer	I	R	R	R
SRE	I	R	R	R
QA	I	R	R	R
Scrum Master	I	A	I	R
Tribe Leader	I	I	I	R

Nota. Matriz RACI (2022), fuente propia.

En esta asignación se demuestra la autonomía de cada squad. Con base en la definición del Product Manager (Roadmap de producto), el resto de los procesos son atendidos por el equipo evitando dependencias con equipos externos.

Al comienzo del programa los candidatos eran escasos y con pretensiones salariales que no coincidían con el presupuesto estimado o generarían inequidad en el equipo, esto motivó a planificar el programa sabiendo que habría que capacitar y desarrollar a las personas que aspiraran a esta función.

Si bien se pretendía que el rol de QA quedara bajo las funciones del PO. Esto con el fin de minimizar los errores comunicacionales, ya que al ser quien el PO quien interpretaba los pedidos del PM, si luego de verificar el comportamiento del producto y éste no se comportaba como esperaba el PM entonces el punto de falla en la comunicación sólo podría haberse dado entre PM y PO, resultando muchos más fácil identificar el problema y trabajar en consecuencia.

Al haber definido que el rol de PO lo tomaban las personas internas, priorizando el conocimiento de negocio sobre el conocimiento técnico. Era una habilidad más a desarrollar y podría sobrecargar a las personas. Razón por la cual circunstancialmente el rol fue asignado a desarrolladores por ser una función que se asemejaba más a sus tareas diarias.

En primera instancia tratamos de buscar perfiles certificados en los grandes proveedores de cloud (aws, gcp, azure). Apalancando las búsquedas en nuestros proveedores. Su devolución fue homogénea, en el mercado argentino estos perfiles eran muy difíciles de encontrar. Los pocos que se encontraron tenían pretensiones muy elevadas que hacían dudar que pudieran desempeñarse en un ambiente colaborativo como el que perseguía.
Por dicho motivo decidimos formar los perfiles necesarios como parte de nuestro cloud journey.

Sabiendo que debíamos escalar rápidamente las mismas búsquedas se dispararon en todos los proveedores. En las entrevistas se priorizo la detección de actitud más que de aptitud. Y solicitamos que hubiese al menos una persona de cada proveedor para seguimiento y gestión de gente en caso de que en el gran número de incorporaciones diésemos con personas que no estuvieran en sintonía con el equipo y el proyecto.

Para orientarnos a producto se identificó a un responsable único de cada producto. Aquellos productos a los que accedía el cliente de manera directa se asignó un responsable del área de Marketing. Para los productos internos se identificaron responsables de diversas áreas. Estos responsables en todos los casos eran los dueños del producto, debían dirigir y evaluar todas las funciones y la evolución del producto. Esto incluía tareas específicas como solicitar el presupuesto para dar continuidad.

Governance

El programa se organizaba en sprints, cuya duración recomendada fue de dos semanas, sin embargo, si había algún motivo que ameritase considerar un sprint de 1 o 3 semanas este se evaluaba y podía ser factible. La sugerencia de duración siempre provenía del propio squad, siguiendo los principios de autonomía y autogestión.

Asimismo, cada squad definía las ceremonias scrum, así como su frecuencia, que consideraba necesarias. Un Sprint usualmente tendría los siguiente eventos o ceremonias:

- El Sprint Planning al comienzo del Sprint
- Daily Scrums a diario

- Un Sprint Review al final del Sprint para inspeccionar el incremento realizado.
- Una Retrospectiva para inspeccionar el equipo y levantar mejoras que se apliquen en el siguiente Sprint.

A nivel equipo de tecnología, semanalmente se realizaba una reunión entre todos los Product Owners y el líder del programa para identificar dependencias y sinergias en los desarrollos. Estas reuniones fueron muy productivas para ir comprendiendo y acelerando los procesos globales de aprobación. Adicionalmente se realizaba una reunión mensual con todo el equipo para comentar los logros y los próximos pasos.

A nivel reporte hacia la dirección había una reunión semanal entre Product Managers, Product Owners y todos los stakeholders involucrados en la iniciativa. Y finalmente existía una reunión mensual donde se enviaba un reporte hacia la corporación y donde se exponía el estado hacia la corporación. Allí participaba el líder del programa por la parte de negocio, el líder del programa por la parte de TI y el Program Manager.

Espacio físico

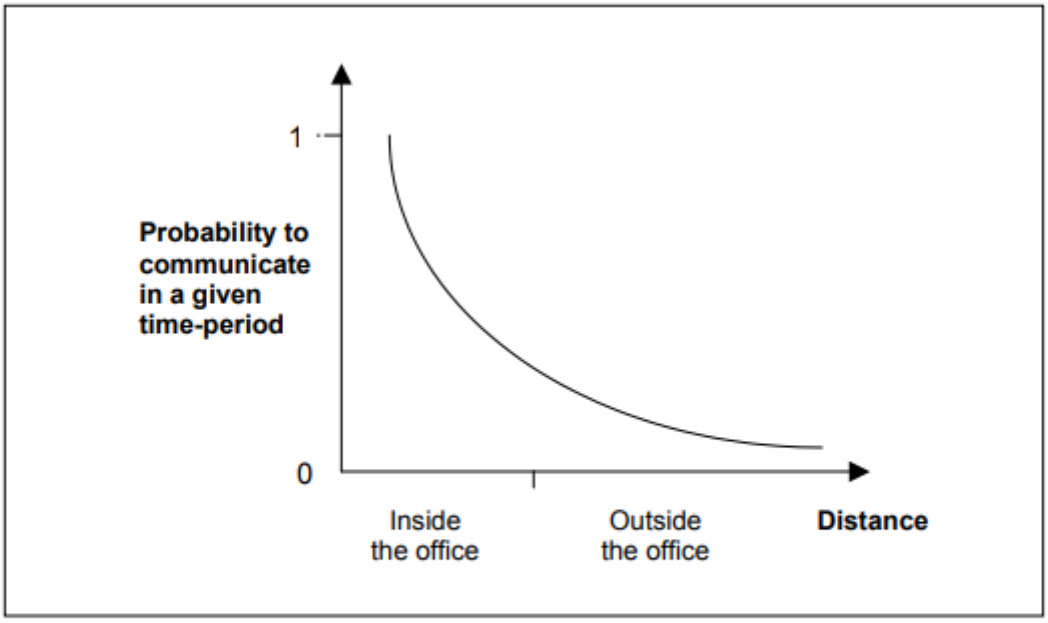
Respecto del conocimiento que desarrollan y transfieren los equipos hay evidencia de que la distancia espacial puede tener un efecto inhibitorio significativo en la comunicación, impidiendo el intercambio y la transferencia de conocimientos. Por cierto, la probabilidad de comunicación disminuye exponencialmente con la distancia física entre los interlocutores (Boutellier, Gassmann, von Zedtwitz 2000). Esta relación, que ha sido comprobada en el micro- (distancias en metros, por ejemplo, en el mismo edificio) así como en el macro-rango (kilómetro distancias), se representa en la llamada curva de Allen (figura 8.2), llamada así por el autor del estudio original.

Por este motivo se consideró de suma importancia fomentar la interacción personal de los integrantes para así lograr formar y fortalecer la cultura se decidió ubicar físicamente a los miembros lo más cercano posible. Inicialmente se contemplaba un sólo día de trabajo remoto para el equipo, por ser considerado un factor atractivo para la contratación de talento. Ese día debía ser acordado por todo el squad con el objetivo de que los cuatro días restantes de la semana el squad estuviera completo de manera

presencial. Una vez que se lograra identidad y se adoptara un marco de trabajo se irían extendiendo los días de trabajo remoto.

Asimismo, un squad debía compartir la misma mesa, mientras que las tribus se conformaban por mesas contiguas.

Figura 8.2.



Nota. Curva de Allen, fuente Boutellier, Gassmann, Von Zedtwitz (2000), p 188

Con estas líneas en mente debíamos ingresar y ubicar alrededor de 65 personas, para los cuales debimos buscar un espacio de trabajo ya que no había espacio físico suficiente en las oficinas. La alternativa fue buscar opciones en espacios de coworking. Las opciones que se alineaban a los tiempos fueron Wework y Regus. En ambos casos no se contaba con un espacio único donde entrara el equipo completo. Además, debido al escaso tiempo para comenzar con el programa no fue posible seguir analizando opciones o esperar a obtener el espacio ideal dentro de estas empresas con el fin de que estuviese todo el equipo junto, por lo que optamos por priorizar la cercanía de la tribu, garantizando que en cada oficina estuviese una tribu completa. Por esta razón decidimos contratar Wework, donde necesitábamos de menos oficinas y nos posibilitaba ubicar tribus enteras en ellas.

Proceso de gestión de la iniciativa

Gestión de iniciativas

Toda iniciativa debía ser canalizada a través del PM. Éste actuaba como interfaz entre el mundo y el producto. El pedido hacia él podía surgir de los comerciales, usuarios finales o del mismo equipo que desarrollaba el producto. Como dueño del producto era el responsable de evaluar y analizar si éstos pedidos eran coherentes con la estrategia de producto, gestionando y priorizando el roadmap del producto.

Una vez que éste aceptaba un nuevo feature para el producto, entonces se reunía con el PO -quien podía o no participar con el equipo- para así entender dicha funcionalidad y estimarla en alto nivel. Esta estimación permitía darle información suficiente como para priorizar dicho pedido.

Si la iniciativa entraba en etapa de desarrollo, entonces el PO priorizaba el backlog para ejecución. Conformando de este modo dos etapas bien definidas de estimación. Una de alto nivel que permitía al PM ubicar la necesidad en el roadmap, y otra de bajo nivel permitiendo al equipo de desarrollo gestionar el detalle.

Estimaciones de desarrollo

Para las estimaciones por equipo si bien cada equipo podía optar por su método favorito la recomendación fue iniciar por poker estimation o planning poker.

Según María Najarro (Najarro, 2017) el Planning Poker es un proceso de planificación de un proyecto por parte de todo el equipo implicado que utiliza objetivos o puntos de historia para valorar un proyecto. El método consiste en determinar un número basado en la serie Fibonacci que representa el grado de complejidad, en este caso el esfuerzo necesario para completar una tarea.

Cada participante cuenta con una baraja de cartas y el cliente explica un objetivo. Cuando todo el equipo está alineado y entiende la tarea que se tiene que puntuar. De forma individual, cada participante

pone sobre la mesa la carta que representa la puntuación que él o ella considera necesaria para llevar a cabo dicha tarea del proyecto en cuestión. De haber puntuaciones muy dispares, cada participante explica los motivos de su decisión. Y si fuese necesario, puede repetirse la votación hasta llegar a un consenso. Así, el resultado es una estimación consensuada y validada por todo el equipo.

Entre las ventajas más notorias de este sistema se pueden resaltar:

- Todos los miembros del equipo expresan su opinión sin sentirse condicionados por el resto.
- Al ser conscientes del esfuerzo que supone, aumenta el grado de implicación de los componentes del equipo.
- Al sentirse partícipes, el grado de compromiso con el proyecto también aumenta.
- Hay más efectividad a la hora de estimar las fechas de entrega del proyecto.

CAPÍTULO IX - RESULTADO FINAL

Al cabo de un año y medio de programa el equipo se encontraba afianzado y compartiendo una cultura común de trabajo. En cada equipo estaban claros tanto los roles y responsabilidades de cada miembro, como la metodología de trabajo adoptada. El esquema de governance y reporting se seguía independientemente de la presencia de los líderes, confirmando la autonomía del equipo. El índice de rotación de personal había disminuido pasando de un 20% en el periodo abril - diciembre 2019 a un 13% en el periodo enero - octubre 2020. Se habían contratado algunos miembros en relación directa con el objetivo inicial de que el conocimiento permaneciera en la empresa. De las 7 ofertas de contratación directa realizadas 5 miembros del equipo la aceptaron.

De la planificación original respecto a los roles del equipo, quedó pendiente traspasar las responsabilidades de QA y Scrum Master a los POs. Ambos roles aún permanecían dispersos entre los integrantes de los squads, sin embargo el hecho de que todos hayan pasado por la experiencia de comportarse como Scrum Master demostró ciertas habilidades de algunos integrantes, por ejemplo permitió identificar a aquellos con capacidades de liderar sin autoridad, algo que los posicionó como candidatos para ser POs.

A lo largo del proyecto no todos los equipos lograron sus objetivos. Dos de ellos no funcionaron de la manera esperada. En ambos casos se identificó el mismo síntoma. Estos equipos mostraron mucho avance en la fase de prototipado, pero dedicaron demasiado tiempo a esta construcción que perdieron de foco el motivo principal de validación que perseguía el prototipo. Se realizaron muchas demos al negocio, y en general el feedback era muy positivo, sin embargo a medida que se acercaba la fecha de entrega de la fase beta entonces se replanificaba para un futuro. Cuando se detectó este inconveniente entonces se realizó un micromanagement (gestionar de cerca el trabajo sobre las actividades críticas dando especificaciones detalladas para ejecutar cada paso) sobre ambos equipos, sin embargo esto causó malestar en los equipos. Si bien se logró el compromiso de entrega, al cabo de unos meses los equipos prácticamente se reconstruyeron. En todos los casos fue por iniciativa de los integrantes, la mayoría optó por abandonar la empresa empleadora, y en menor medida solicitaron el cambio de proyecto.

La calidad del desarrollo de los productos había aumentado considerablemente, se pasó de registrar un promedio de 30 incidentes semanales a menos de 5. Los tiempos de puesta en producción habían disminuido de semanas a días, gracias a las prácticas de devops.

Algunas métricas adicionales que se observaron fueron:

- En materia de seguridad, los parches estaban al día y en constante análisis. Una actualización de esta índole se resolvía en el día, mientras que anteriormente demandaba al menos una semana.
- La evaluación de cambios de alcance se realizaba en el sprint en curso, tomando usualmente 3 días. Lo que permitía al PM aceptarlo o rechazarlo. Anteriormente no se tomaban métricas, en muchos casos no se realizaba el análisis y simplemente no se aceptaban.
- Se redujo el costo de la infraestructura con respecto al on premise. Si bien los productos aún no estaban en producción los costos que generaban en la nube eran considerablemente más bajos que en el onpremise. Las estimaciones más pesimistas fueron de un factor 5 veces menor.

Quedó pendiente modificar el proceso de gestión de cambios, lo cual podría haber contribuido en una mejora de aproximadamente 10 días. El objetivo de puesta en producción en cualquier momento del día no se pudo probar. La modificación de ese proceso implicaba cuestiones políticas y estructurales que no se llegaron a abordar.

Para octubre de 2020 los equipos trabajan con autonomía y cada producto tenía su evolución trazada. Todos los productos en desarrollo habían sido presentados a las diversas geografías de Equifax, con gran aceptación. Por esta razón se decidió modificar el plan original para extender la metodología a los 8 países restantes de Latinoamérica. De manera que los productos fueran comunes a todos los países, pero responsabilidad de una única tribu.

Este cambio de enfoque no fue difícil de asimilar, y se pudo demostrar el poder de escalabilidad del modelo. Del mismo modo que en la etapa anterior se redimensionaron los equipos para tener más squads para poder hacer frente a una mayor carga de trabajo. La manera de crear nuevos squads fue dividir a los actuales e incorporar nuevos miembros de diversos países, de este modo se persiguió que la cultura

y la manera de trabajar se dispersara entre los nuevos equipos. De igual modo que al comienzo del proyecto, la rápida escalabilidad se dio gracias a contrataciones tercerizadas.

CAPÍTULO X – CONCLUSIÓN

Si bien el programa no concluyó acorde a lo planificado por un cambio en la estrategia. Éste cambio fue motivado por los resultados favorables que se fueron percibiendo en tan solo 18 meses de ejecución, y persiguió extender éstas prácticas al resto de Latinoamérica.

Hubo algunos factores que se destacaron sobre otros, uno de los más destacables fue la velocidad de formación del equipo, apalancándose en contrataciones de terceros. En este punto cabe destacar que la formación total del equipo se produjo a los 9 meses de iniciado el proyecto. En ese momento alineamos los beneficios y los salarios de todas las consultoras, ya que hasta ese momento esa disparidad contribuyó a una mayor rotación del personal. Seguramente si este acuerdo hubiese sido hecho al comienzo hubiese sido más provechoso. Otro factor que incidió negativamente en el factor de rotación fue haber contratado a los perfiles que iban a ser formados en tecnologías cloud sin tener acceso a las herramientas. Por desconocimiento de los nuevos procesos globales esos accesos demoraron meses y las personas no toleraron la espera.

Para la escalabilidad y generación de una cultura común resultó muy favorable incorporar gente a un squad en funcionamiento, y una vez que creciera lo suficiente partirlo en dos equipos más chicos. De este modo la cultura de trabajo se esparcía rápidamente y tenía un alto nivel de coherencia entre todos los equipos. Esta estrategia se usó en todos los equipos, la única restricción era que esos equipos se dividirían en otro que perteneciera a la misma tribu.

La asignación de equipos dedicados para cada producto fue muy bien percibida por toda la organización, ordenó desde un inicio la comunicación y el rumbo estratégico de la compañía. La relación Project Manager - Product Owner fue crucial para este modelo, y es uno de los puntos donde se debe hacer foco. Dicha relación debe ser armoniosa y ambos tomar un rol colaborativo entre ellos, para fomentar estos acuerdos el haber definido que uno sea backup del otro durante las ausencias, como por ejemplo época de vacaciones contribuyó de manera positiva al fortalecimiento de esta relación.

El principio de autonomía por el que siempre se veló fue un aspecto positivo, sin embargo no fue sencillo, al comienzo del programa la mayoría de los equipos no sabían qué rumbo tomar, puesto que por

ejemplo esperaban que se les brindara una arquitectura y a partir de la misma empezar a construir. En ese momento hubo que realizar varias juntas para explicar el modelo de trabajo, pero a fin de cuentas fue muy apreciado por todos los equipos.

Esta autonomía generó algunos casos de conflicto. Donde se destacaron dos situaciones en algunos equipos. Por un lado, como los equipos decidían qué variable modificar de la triple restricción (costo - alcance - tiempo), algunos equipos donde todos los integrantes, incluido el PM, siempre optaban por mover la fecha de entrega continuamente, alegando que se había ampliado el alcance. Esto perduró por algunos sprints hasta que se les pidió que fijaran la fecha de entrega. Lamentablemente no llegaron a cumplir con el objetivo, pues lo que en realidad estaba ocurriendo es que el avance reportado no reflejaba la realidad. A partir de este caso se definió que para los próximos releases sólo podían trabajar sobre ajuste en alcance y no en fecha, el costo no era una opción puesto que el presupuesto había sido completamente destinado a la creación de equipos. La otra situación se dio cuando los equipos abusaron de la presentación de prototipos y demos perjudicando el desarrollo. Como era una práctica no habitual antes del proyecto el impacto que generó fue muy alto. Los equipos ajenos a IT no lograban determinar cuánto realmente faltaba para productivizar un producto, y desde el equipo de IT nunca se concientizó al respecto. Si bien el prototipado fue muy valioso se recomienda ser muy cuidadoso en organizaciones inmaduras puesto que puede llevar a una situación donde se subestimen tareas de IT relacionadas con la puesta en producción de un producto. A partir de ese caso se decidió establecer tiempos máximos para la generación de prototipos.

Con relación a los roles de los equipos no se logró formalizar que las funciones de Scrum Master y QA quedarán a cargo del PO, la formación y consolidación de procesos que ayudaran a formar rápidamente estos roles demandó más tiempo de lo previsto, esto se debió a que al no haber recursos dedicados estas iniciativas se sacrificaban en pos de mejorar el tiempo de entrega.

Durante la ejecución del programa se detectó la necesidad de crear un rol que nunca había sido contemplado. A los 6 meses de iniciado el proyecto optamos por crear un rol al que llamamos service delivery manager. Su función era entender y conocer los procesos corporativos para solicitar tareas a los equipos globales. Esta función fue de suma ayuda e importancia. Al inicio desconocíamos estos procesos, algunos no existían y otros fueron madurando con el tiempo. La incorporación de este rol, ayudó a todos

los equipos a disminuir los tiempos que demandaban las dependencias con la corporación. Seguramente si hubiésemos contado con este rol desde el inicio los tiempos de delivery hubiesen sido aún mejores.

Durante el transcurso del programa el modelo quedó probado y demostrada su eficiencia en dos momentos claves:

El primero fue marcado en la etapa de la pandemia del COVID (marzo 2020). En ese entonces los equipos más maduros estaban haciendo dos días de trabajo remoto semanal, sin embargo hubo que adaptarse a que todos los miembros estuviesen trabajando de forma remota. Al comienzo hubo ciertas dificultades con los accesos a ciertos sitios corporativos, dado que la conectividad no estaba dimensionada para todo el personal accediendo al mismo tiempo. Sin embargo, esto no impactó en el avance general de los proyectos, las métricas no se vieron impactadas por el cambio en la modalidad de trabajo. Desde marzo en adelante no hubo más presencialidad, aun así los resultados dictaron el cambio de estrategia para extenderse hacia todo Latinoamérica. Al extenderse la no presencialidad hubo que adaptar o modificar ciertas dinámicas con las que se venía operando. Por ejemplo, en las reuniones informativas para el board disminuimos el número de participantes ya que era difícil mantener el orden y la atención de todos los participantes durante las videollamadas. En este caso sólo participaban los PMs para comentar el estado y se complementaba dicha información con correos electrónicos. Otro punto que se vio altamente impactado por la no presencialidad fue la formación de los perfiles menos experimentados. Como el modelo se basaba altamente en la colaboración y la ayuda que los miembros más experimentados pudieran brindar a los recién ingresados al estar distanciados ésta dinámica se rompió puesto que esa colaboración dejó de ser espontánea y debían agendar sesiones de capacitación. En este caso fue importante armar una currícula formal de cursos para que pudiesen aprender los aspectos técnicos, esto se complementó con sesiones formales de formación sobre conocimiento del negocio que brindaban los miembros más experimentados, y en este caso podían participar todos aquellos que lo consideraran necesario, no se limitaba a tribus. A este estadio se llegó luego de 4 meses, seguramente se hubiese minimizado el impacto de haberlo definido más rápidamente. Con respecto al reclutamiento se reasignaron ciertas vacantes juniors a semi seniors -reduciendo el número de integrantes para mantener el presupuesto- con el objetivo de poner un mayor foco a la formación de los juniors, quienes demandaban más atención.

El segundo fue la extensión del equipo hacia Latinoamérica, donde bajo la misma premisa de crecimiento de los equipos -crecer en número de integrantes y luego dividirlos en equipos más pequeños- y la modalidad de trabajo con absoluta autonomía permitieron poner en marcha el programa de una manera mucho más veloz que en el programa original. Esto implicó contratación del equipo, la generación de los mapas de producto y la puesta en marcha del desarrollo de todos los productos contemplados en alcance.

Anexos

Perfiles de profesionales en tecnología (Cessi)

Arquitecto de software

Responsabilidad / Principales competencias

Es el responsable de la definición y diseño de la arquitectura.

Entre sus competencias está:

Definir la arquitectura de los sistemas tomando las decisiones de diseño de alto nivel y estableciendo los estándares técnicos, incluyendo plataformas, herramientas y estándares de programación, teniendo en cuenta los requisitos funcionales, no funcionales y las necesidades del negocio.

Nombres de cargos similares

Diseñador de Software, Diseñador de Soluciones, Desarrollador Senior.

Misión de la ocupación

En cooperación con el Líder de Proyecto, participa en la toma de decisiones adecuadas para lograr una arquitectura del sistema que garantice un mejor desempeño, flexibilidad, mantenibilidad, robustez, reuso o las cualidades que se pretendan de la aplicación.

Actividades que realiza

Negociar con el propietario de la aplicación y el líder de proyecto para tomar las decisiones de diseño de alto nivel que correspondan.

Seleccionar el software: Seleccionar la tecnología a utilizar en conjunto con el líder de proyecto; y definir y revisar estándares y normas aplicables al diseño y construcción, brindando coaching técnico al equipo de desarrollo.

Diseñar la arquitectura: Conducir la construcción del modelo de arquitectura/diseño, subdividiendo aplicaciones complejas en partes o componentes menores, más fáciles de manejar.

Realizar un seguimiento del proceso de desarrollo para asegurarse que sus instrucciones se implementen en forma adecuada; y registrar los cambios de arquitectura que se produzcan.

Asegurar la calidad: Medir la performance de la aplicación y conducir pruebas en relación a la performance, seguridad, etc.

Facilitador: Colabora con otras áreas como seguridad informática, base de datos, operaciones, el equipo de mantenimiento, etc.

Liderazgo Técnico: Capacidad para asumir la dirección técnica y asegurar todos los aspectos de la arquitectura con responsabilidad y autoridad. Poder realizar coaching y mentoring sobre problemas técnicos, ayudando a la evolución profesional del equipo de programadores.

Conjuntamente con el líder de pruebas, definir las pruebas de los requisitos no funcionales y de integración que

se realizarán y efectuar un seguimiento de las mismas.

Estudios

Ingeniería informática o sistemas, licenciatura en sistemas o computación en curso (no necesariamente completas). El puesto requiere experiencia en el rubro.

Conocimientos necesarios, según el nivel del puesto

Dominio de arquitectura de software, metodología de componentes y su interacción.

Conocimiento de las tecnologías de comunicación disponibles.

Estándares y normas a aplicar en la construcción de software de la tecnología a su cargo.

Conocimiento en programación avanzados en varios lenguajes, arquitecturas y paradigmas.

Manejo de herramientas para la gestión de requerimientos y ambientes de desarrollo.

Lecto-comprensión y elementos de redacción en inglés.

Conocimiento avanzado de Bases de Datos (tanto en la rama de programación como administración).

Conocimiento avanzado de comunicación entre aplicaciones: SOA, Servicios Web (SOAP, REST), protocolos y lenguajes de comunicación (XML, JSON).

Conocimientos de Ingeniería del Software.

Prácticas de Testing y Refactoring.

Conocimiento de metodologías de análisis como UML u otras.

Conocimiento de metodologías ágiles como SCRUM u otras.

Conocimiento de herramientas de control de versiones como GIT u otras.

Conocimientos de patrones de software empresarial.

Competencias, habilidades o aptitudes deseables según el nivel del puesto

Iniciativa

Creatividad

PlanificaciónyControl

Pensamiento de Integración

Pasión por la tecnología

Capacidad de análisis y de síntesis para la resolución de problemas complejos

Capacidad de trabajar en equipo, compartiendo información y conocimientos y tomando en cuenta a los usuarios

Actualización permanente en las tecnologías a su cargo, de forma tal de proponer e instrumentar

Gran capacidad de aprendizaje.

Ámbito ocupacional

Puede formar parte de un área de soporte a distintos proyectos o formar parte de un equipo de trabajo organizado por proyecto. Es el referente para el diseño, arquitectura y para solución de problemas complejos.

Posibles especializaciones o áreas de especialización

Aplicaciones móviles.

Requiere conocimiento de sistemas operativos y software para dispositivos, lenguajes o ambientes de programación para sistemas embebidos

Desarrollador de Software

Responsabilidad / Principales competencias

Es el responsable por el desarrollo de las piezas de software.

Entre sus competencias está:

Diseñar, producir o mantener (programar, adaptar e integrar) componentes o subconjuntos de software (clases, módulos, pantallas, rutinas, subsistemas, programas en general) conforme a especificaciones (funcionales y técnicas) para ser integrados en aplicaciones.

Nombres de cargos similares

Programador, Analista-programador, Developer.

Misión de la ocupación

Participar del proceso de programación/implementación teniendo como entrada las especificaciones de software y ajustándose a tiempos y estándares de calidad y trabajo de la organización y del proyecto.

Actividades que realiza

Comprender el requerimiento de software.

Modelizar y refinar especificaciones a fin de determinar un diseño detallado para implantar la

funcionalidad requerida.

Construir o adaptar clases, módulos u otras piezas de software. Reutilizar componentes existentes integrándolos con código propio. Verificar los programas producidos probándolos en forma unitaria.

Revisar el código para resolver defectos o mejorarlo.

Documentar los programas de acuerdo a estándares. Tanto con documentos funcionales y de arquitectura, como con comentarios en el propio código.

Investigar nuevas tecnologías, cumplir con los procesos definidos y proponer mejoras.

Entregar a través de la totalidad del ciclo de vida de aplicaciones concepto, diseño, desarrollo, implementación, prueba y soporte.

Mantener las aplicaciones productivas, corregir errores e implementar nuevas funcionalidades.

Cumplir con las entregas según las estimaciones de tiempo realizadas.

Trabajar en estrecha colaboración con otros desarrolladores.

Trabajar directamente con los desarrolladores y gerentes de producto para conceptualizar, construir, probar y realizar productos.

Reunir los requerimientos de funcionalidad y traducir esas necesidades en soluciones funcionales elegantes.

Construir prototipos en fase de alcance técnico de los proyectos.

Optimizar el rendimiento de las aplicaciones.

Trabajar para construir aplicaciones con foco en la funcionalidad.

Crear interfaces de usuario atractivas.

Detectar oportunidades de mejora o conformidades y observaciones para incluirlas en el sistema de mejora continua.

Participar en la definición y revisión de estándares y normas aplicables al diseño y construcción.

Estudios

Ingeniería informática; licenciatura en sistemas o computación; técnico superior en desarrollo de software (estudios finalizados o en curso), técnico secundario en programación.

Conocimientos necesarios, según el nivel del puesto

Algoritmos y estructuras de datos.

Dominio de lenguajes de programación (Paradigmas: Estructurados, Orientado a Objetos, etc.).

Manejo de herramientas para la gestión de requerimientos y ambientes de desarrollo.

Lecto-comprensión y elementos de redacción en inglés.

Conocimiento de Bases de Datos y lenguaje de consultas estructurado.

Conocimiento de Servicios Web.

Conocimiento de SOA: Aplicaciones orientadas a servicios.

Conocimiento de protocolos y lenguajes de comunicación como XML, JSON.

Conocimientos de Ingeniería del Software.

Nociones de arquitectura de software y aplicaciones N-Capas.

Prácticas de Testing y Refactoring.

Conocimiento de metodologías de análisis como UML u otros.

Conocimiento de metodologías ágiles como SCRUM u otros.

Conocimiento de herramientas de control de versiones como GIT u otros.

Conocimientos de patrones de software empresarial.

Competencias, habilidades o aptitudes deseables según el nivel del puesto

Iniciativa, proactividad y capacidad de análisis para la resolución de problemas.

Flexibilidad en sus enfoques de las situaciones y búsqueda permanente de la calidad.

Actuar ateniéndose a un orden propio que le facilite el acceso a lo que pueda necesitar.

Conocimientos, metodologías y modelos de calidad para la industria del software.

Capacidad de trabajar en equipo, estando dispuesto a preguntar, a compartir información y conocimientos y a tomar en cuenta a los usuarios.

Otras habilidades interpersonales deseables

Creativo

Buen manejo del tiempo

Capacidad de resolución de problemas

Gran capacidad de aprendizaje

Capacidad de autogestión

Actitud, responsabilidad y compromiso

Orientación al Cliente Interno/Externo

Orientación a Resultados

Actitud crítica, de perfeccionamiento y actualización permanente

Capacidad para trabajar en forma cooperativa y constructiva, en equipos multidisciplinarios

Capacidad de abstracción: comprender la “big picture” más allá de detalles de implementación

Permanente capacidad para derivar de la práctica, elaboraciones conceptuales.

Ámbito ocupacional

Forma parte de un equipo de trabajo organizado por proyecto o por práctica donde cada uno asume la responsabilidad de resolver en tiempo y con calidad su parte, teniendo que satisfacer pruebas de conjunto y pidiendo o brindando colaboración cuando se encuentran problemas. En empresas donde se trabaja con objetivos y medidas de productividad, es posible el teletrabajo.

Posibles especializaciones o áreas de especialización

Desarrollador de aplicaciones clientes-servidor:

Diseñar, programar, implementar y mantener aplicaciones de escritorio.

Programar con capacidad de conexión con un servidor de aplicaciones.

Sólidos conocimientos de tecnologías de comunicación de aplicaciones distribuidas.

Sólidos conocimientos de consultas avanzadas en bases de datos robustas con gran volumen de información.

Desarrollador Web:

Diseñar, programar, implementar y mantener aplicaciones Web Front-end y Back-End

Sólidos conocimientos en HTML, CSS, JavaScript y extensiones de JS como jQuery u otros

Conocimiento de protocolos de internet (ej: HTTP, HTTPS, FTP, TELNET, etc)

Conocimiento de lenguajes de estructuras y transferencias de datos serializados como XML, JSON

Conocimiento de programación asincrónica y servicios web: AJAX, SOAP, Rest API

Conocimiento de testing funcional y de stress de aplicaciones web

Capacidad de desarrollo de interfaces web responsivas para distintos tipos de pantallas

Conocimientos de Seguridad Web. Técnicas de prevención de ataques por SQL Injection y XSS

Desarrollador Juegos / Aplicaciones Lúdicas:

Analizar y desarrollar el código fuente a partir de stories, improvements y bugs

Colaborar con las tareas de testing de los juegos desarrollados, creando casos de prueba automatizados cuado sea necesario y eventualmente creando issues o haciendo seguimiento de los mismos o el cierre.

Participación en el Basecamp del proyecto de ser necesario

Desarrollo y generación de ideas para nuevos juegos

Desarrollador Aplicaciones móviles:

Participar en la definición de la interfaz gráfica para móviles

Conocer sobre el desarrollo de apps móviles para las diferentes tecnologías nativas (ios, android, windows)

Integración entre desarrollo mobile y desarrollo web. Comunicación un servidor web a través de API Rest.

Trabajar junto a los desarrolladores web para crear y mantener un marco sólido para soportar las aplicaciones móviles y web

Trabajar con los desarrolladores de front-end para construir la interfaz con el foco en las características de usabilidad

Crear interfaces de usuario atractivas, específicas del dispositivo y experiencias

Normalizar la plataforma para entregar a través de múltiples marcas con mínima duplicación de esfuerzos

Mantenerse al día sobre las últimas tendencias del sector en las tecnologías móviles

Desarrollador de sistemas embebidos:

Conocimientos avanzados de Electrónica Digital (Multímetro, Osciloscopio, Soldadura, Ley de Ohm, Resistencias, Capacitores, Semiconductores, Circuitos Integrados)

Concepto de ánodo y cátodo, Multiplexación, Transistores

Manejo de sensores para tomar información del medio ambiente (luz, temperatura, humedad, humo, etc.)

Lectura de señales digitales (Digital Read, Pulsador, Internal Pull-Up, Pull-Up / Pull-Down, Interrupciones)

Herramientas y protocolos de comunicación (Comunicación Serial, Bit / Bytes, Bluetooth, Wifi)

Conocimientos de procesadores cortex

Desarrollo de sistemas sobre placas open hard como Arduino o Raspberry u otros

Arquitectura de Integración por Serie

Memoria SD, Flash, EEPROM

Request / Response HTTP

Configuración de Router

Arquitectura de Integración por Ethernet

Domótica, Robótica, Internet de las cosas.

Project Manager

Responsabilidad / Principales competencias

Es el responsable de liderar y gerenciar el proyecto para lograr los resultados esperados en tiempo y forma.

Entre sus competencias está:

Conducir las actividades diarias del equipo de proyecto, ejerciendo un control sobre resultados, plazos y calidad. Mantener la relación con usuarios y cliente, motivando y brindando apoyo a los integrantes del equipo y gestionando los recursos necesarios, tomando las decisiones operativas necesarias para mantener el proyecto en tiempo, alcances y costo.

Nombres de cargos similares

Líder de Desarrollo, PM (Project Manager) o SDM (Service Delivery Manager), solo cuando el Proyecto ya se encuentra en operaciones, dando mantenimiento.

Dependiendo del alcance de su rol (si no es exclusivamente Líder de Desarrollo), puede gerenciar todo el proyecto, incluyendo los grupos de procesos y las 9 áreas de conocimiento definidas por PMI (Project Management Institute): Alcance, Costo, Tiempos, Calidad, Integración, Riesgos, Comunicaciones, RRHH y Compras.

Misión de la ocupación

Lograr que el proyecto se desarrolle dentro de los alcances, costos y calidad establecidos en los plazos previstos con la menor cantidad de inconvenientes, anticipando posibles problemas o desvíos y tomando decisiones correctivas o proponiendo alternativas a la gerencia.

Actividades que realiza

Estimar tiempo y recursos necesarios para el desarrollo de un proyecto.

Definir el presupuesto y obtener el mejor rendimiento del mismo.

Liderar y gestionar los proyectos a su cargo.

Establecer un plan del proyecto, controlando su progreso y efectuando el seguimiento de los desvíos.

Administrar los grupos de trabajo a su cargo, motivando a sus integrantes, cuidando su adecuada capacitación o coaching y resolviendo eventuales conflictos.

Colaborar con el analista funcional en la interpretación y comprensión de las necesidades.

Participar en el diseño de soluciones asociadas a los requerimientos y colaborar en la definición de las arquitecturas.

Elaborar informes de avance del proyecto, tomando decisiones correctivas o proponiendo soluciones alternativas a la gerencia.

Asegurar la implementación de productos o servicios de acuerdo al alcance acordado para lograr la satisfacción del cliente.

Involucrar a los usuarios o clientes cuando haga falta, negociando compromisos con los mismos o con otros grupos o personas involucradas.

Asegurar el cumplimiento de los procesos definidos por el área de calidad.

Cuidar los aspectos de un proyecto que puede incidir en sus alcances, plazos y calidad, incluyendo riesgos y manejo de los cambios, realizando el análisis de impacto y negociando compromisos.

Hacer seguimiento de cada fase y cada hito del proyecto, monitoreando tiempos, costes, calidad y riesgos.

Detectar desviaciones: agendar entregas intermedias con el cliente con el fin de mitigar errores y alinear expectativas.

Asegurar que los proyectos sean cerrados apropiada y formalmente.

Definir el perfil de cada miembro de su equipo, como así también controlar las limitaciones y las funciones de cada perfil dentro del proyecto.

Colaborar con la unidad de negocios, ayudando para establecer el precio de venta, si así se requiere.

Asegurar la comunicación a todos los niveles (visibilidad al equipo y avance al cliente).

Estudios

Ingeniería informática; licenciatura en sistemas o computación.

Conocimientos necesarios

Metodologías de desarrollo de sistemas (ágiles, no ágiles). Conocimiento de Scrum/UML/TDD/IC. Diseño Estructurado de Sistemas, Proceso Unificado de Desarrollo de Software, etc.

Manejo de lenguajes de programación orientados a las necesidades de los proyectos.

Conocimientos de ambientes de desarrollo.

Conocimiento de análisis y diseño de bases de datos.

Actividades de Liderazgo (Team Building) / Project Management

Conocimientos deseables

Especialización en Project management, Gerenciamiento de Proyectos o similar.

Inglés, si aplica.

Competencias, habilidades o aptitudes deseables

Liderazgo

Planificación y medición de desempeño

Orientación a resultados

Buen manejo de relaciones interpersonales

Capacidad de trabajo en equipo

Pensamiento Analítico (Big Picture y Breackdown)

Actitud, responsabilidad y compromiso

Orientación al Cliente

Habilidad organizativa

Facilidad para incorporar y asimilar rápidamente los procesos de la organización

Capacidad para la mejora continua

Buen comunicador / orador

Conocimientos de time managment (administración del tiempo)

Buen negociador

Diagrama de Gantt

Ámbito ocupacional

Lidera y tiene a su cargo equipos de trabajo con un objetivo definido, formado entre otros perfiles por analistas funcionales, desarrolladores y testers.

Es el gestor del proyecto y esta experiencia puede traducirse, evolucionar hacia Gestión de Áreas (Gerenciamiento).

En general, reporta al Gerente de Desarrollo o Sistemas.

Posibles especializaciones o áreas de especialización

Gerente de Proyectos o PM

Gerente de Operaciones de un Cliente / Service Delivery Manager

Analista Funcional

Figura ocupacional Analista de Negocios

Responsabilidad o competencia Comprender y analizar el contexto de negocio y procesos de las organizaciones, diseñar e implementar mejoras en sus procesos de negocios y aplicaciones y brindar soporte a la operación.

Misión de la ocupación

Acompañar los cambios y evolución de las organizaciones brindando soluciones de valor agregado sobre sus procesos de negocios y aplicaciones

Estudios

Ingenieros industriales, Contadores públicos, Licenciados en administración, Licenciados en sistemas

Nombres de cargos similares

Consultor Funcional, Analista Funcional, Analista de Procesos

Actividades que realiza

– Analizar los procesos de negocios y diseñar modelos funcionales de las aplicaciones que soportan dichos procesos.

– Analizar requerimientos de negocio e identificar necesidades de cambio o mejora sobre los procesos de negocios y/o aplicaciones.

– Implementar cambios funcionale.

– Definir especificaciones funcionales para nuevos desarrollos o cambios a los existentes.

– Documentar los cambios y diseños acorde a los estándares definidos y metodologías dadas.

– Brindar capacitaciones a usuarios.

– Coordinar y ejecutar pruebas unitarias e integrales.

– Planificar y ejecutar conversiones de datos y puesta en marcha

– Brindar soporte funcional para la puesta en marcha de nuevas aplicaciones, cambios o nuevos desarrollos

– Proveer soporte y/o mantenimiento correctivo/evolutivo.

Conocimientos necesarios

– Conocimiento en aplicaciones

– Conocimiento en procesos de negocios

– Dominio de herramientas y metodologías de documentación y gestión

– Dominio de otros idiomas, principalmente inglés y portugués

Competencias, habilidades o aptitudes deseables

– Iniciativa y Flexibilidad

– Capacidad de análisis para la resolución de problemas.

– Trabajo en equipo

– Conocimiento/experiencia en determinadas industrias

– Gestión de tiempos, alcance, calidad, recursos y riesgos

– Certificaciones

Ámbito ocupacional

– Forma parte de un equipo de trabajo organizado por proyecto o servicio.

– Puede formar parte de la organización directamente o de una empresa que brinda servicios de consultoría.

– Puede trabajar en forma presencial o en forma remota.

– Puede trabajar para clientes nacionales o del exterior.

– Puede desarrollarse como especialista o adquirir habilidades de gestión proyectándose a posiciones gerenciales

Posibles especializaciones o áreas de especialización

– Aplicaciones ERP, CRM

– Especialistas en Procesos: Financieros, Logísticos, Producción

– Especialistas en Industrias: Consumo Masivo, Recursos Naturales, Servicios Financieros, Retail, Lifescience

Analista Tester

Responsabilidad / Principales competencias

Es responsable por realizar el Control de Calidad del Producto de Software (Quality Control), intentando encontrar la mayor cantidad de fallas antes de que el software salga a producción.

Entre sus competencias está:

Construir el plan de testing; diseñar casos de prueba; gestionar los ambientes de prueba; generar los datos de prueba; ejecutar y documentar pruebas, reportando y siguiendo los incidentes encontrados, usando las herramientas apropiadas de acuerdo a los estándares de trabajo de donde se desempeñe.

Nombres de cargos similares

Tester, Líder de Pruebas, Analista Control de Calidad.

Nota: Muchas veces es asimilado al rol de QA o Analista QA. El Analista QA (definido en el perfil Analista de Calidad) tiene más foco en Calidad de Procesos y Aseguramiento de Calidad, mientras que el Analista Tester tiene mayor foco en Calidad de Producto, especialmente en el Proceso de TestingTechnical Support Engineer, Global Support Engineer, Atención al usuario, Customer Care. Existe Soporte Técnico y Soporte Funcional.

Misión de la ocupación

Asegurar que el software o pieza de software funcione de acuerdo con los requisitos; trabaje con sus interfaces de la forma esperada, detectando en forma temprana defectos y evitando su propagación y llegada al cliente (interno o externo).

Actividades que realiza

Diseñar plan de testing.

Definir los casos de prueba en base a los requisitos funcionales, no funcionales y técnicos.

Test de integración: Definir las pruebas de Integración que se realizarán.

Gestionar el armado de ambientes de prueba.

Generar datos o lotes de prueba.

Ejecutar los casos de prueba.

Realizar la documentación de las pruebas (evidencia).

Registrar los incidentes en la base a los defectos encontrados, así como también realizar su seguimiento para asegurar su adecuada corrección.

Analizar y reportar los resultados de las pruebas, estadísticas, etc.

Participar durante la etapa de concepción de los sistemas en la elaboración de estrategias para las pruebas.

Participar en la definición y revisión de estándares y normas aplicables al diseño y construcción.

Test de Performance

Test de Aceptación

Test de usabilidad/navegabilidad

Test de instalación.

Estudios

Ingeniería informática o sistemas, licenciatura en sistemas o computación en curso (no necesariamente completas). Tecnicatura superior en desarrollo de software o formación equivalente de 2 a 3 años de experiencia en el rubro.

Conocimientos necesarios, según el nivel del puesto

Conocimientos de ingeniería del software

Conocimientos de metodologías y modelos de calidad para la industria del software

Metodologías de desarrollo de sistemas (ágiles, no ágiles).

Conocimiento de Scrum/UML/TDD/IC

Conocimiento de CMMI, ISTQB, Modelo en V, RUP

Conocimientos de herramientas para la ejecución y seguimiento de testing

Conocimientos de lenguajes de consulta de Base de Datos

Conocimientos deseables, según el nivel del puesto

Certificación Internacional en Testing – ISTQB – International Software Testing Qualification Board / HASTQB – Hispanic America Software Testing Qualification Board

Conocimiento, si aplica, de diferentes Tipos de Testing (volumen, Smoke test, performance, funcional, técnico). Automatización / Scripting.

Experiencia en las normas de calidad ISO 9000, si participa en Procesos.

Experiencia en implantación de procesos de mejora de alta productividad.

Conocimientos de redes/ comunicaciones, S.O, algún lenguaje de programación.

Inglés intermedio (lectura).

Competencias, habilidades o aptitudes deseables según el nivel del puesto

Actitud, responsabilidad y compromiso, con orientación a resultados.

Detallista (que preste atención al detalle, a las particularidades).

Pensamiento analítico / crítico.

Pragmatismo (adecuar las técnicas y el esfuerzo al alcance del proyecto).

Capacidad de trabajo en equipo, con buen manejo de relaciones interpersonales.

Capacidad de abstracción.

Orientación al cliente Interno / Externo.

Facilidad para incorporar y asimilar rápidamente los productos / sistemas de la organización

Capacidad para medir el desempeño de un proceso (métricas).

Ámbito ocupacional

Puede formar parte de un staff cross organizacional que brinda servicio de testing a los proyectos. Cuando trabaja dedicado a un proyecto, integra el equipo de trabajo. Puede evolucionar hacia desarrollador, si lo desea.

Espacio propicio para la mejora profesional.

Posibles especializaciones o áreas de especialización

Analista Tester con Orientación Técnica es un subperfil dentro de Analista Tester. Se refiere al perfil orientado a casos de prueba de tipo técnico como stress, volumen, performance, etc.

Analista Tester con Orientación Funcional es un subperfil dentro de Analista Tester. Se refiere al perfil orientado a casos de prueba de tipo funcional /Caja Negra, etc.

Tester de Seguridad (Técnicas de Hackeo, etc.).

Especialista en seguridad de la información

Responsabilidad / Principales competencias

Es responsable de diseñar, implementar y mantener una política de seguridad integral con el objetivo de resguardar toda la información de la compañía.

Entre sus competencias está:

Evaluar riesgos y basado en los mismos, diseñar, poner en ejecución y mantener políticas, medidas y sistemas de seguridad informática para la Organización.

Nombres de cargos similares

Analista de seguridad, especialista en seguridad informática.

Misión de la ocupación

Proveer un marco de metodología y estandarización de seguridad de información a la organización y sus proyectos, detectando en forma temprana causas de desvíos, implementando y administrando sistemas, políticas y normatividad de seguridad informática.

Actividades que realiza

Gestionar la seguridad de la información, aplicando las normativas y estándares existentes, guiando a la misma en la implementación de políticas de seguridad y en la implementación de controles de seguridad y el Sistema de Gestión de Seguridad de la Información (ITSM), alineando las actividades programadas en el marco de los estándares existentes y aplicables.

Desarrollar e implementar las políticas y procedimientos de seguridad. Monitorear su cumplimiento.

Gestionar incidentes y riesgos para garantizar la continuidad del negocio, protegiendo los activos críticos.

Aplicar las metodologías, tecnologías y herramientas que existen en las distintas áreas involucradas, como ser criptografía, modelos formales, análisis forense, etc. , así como en las áreas en las que la seguridad informática tiene su aplicación: redes, sistemas operativos, aplicaciones.

Desarrollar periódicamente tareas de pentest (penetration testing) y todo tipo de ataque ético (Ethical Hacking) con el fin de identificar y medir vulnerabilidades para luego gestionar soluciones.

Establecer un plan de Disaster Recovery (estrategia de recuperación ante desastres)

Establecer un plan de contingencia para suministros de energía interrumpidos.

Realizar análisis de riesgos en nuevas tecnologías

Asistir a los desarrolladores en la solución de vulnerabilidades.

Disuadir, Detectar y Responder a incidentes de seguridad física.

Realizar actividades de gestión de riesgos (planificación, detección, mitigaciones).

Alinear las actividades programadas al marco de los estándares existentes (ISO 27001, COBIT, ISAE3402, SOX, otras).

Estudios

Ingeniería informática; licenciaturas en sistemas o computación, preferiblemente con una especialización o capacitación en seguridad aplicada a sistemas de información (ISO 27001, COBIT, ISAE3402, SOX), como así también de ISO9000.

Conocimientos necesarios, según el nivel del puesto

Experiencia probada en arquitectura e implementación de proyectos de seguridad de la totalidad de los departamentos de IT.

Amplio conocimiento de la solicitud y las vulnerabilidades a nivel de infraestructura. Capacidad de explicar estos riesgos a los desarrolladores.

Capacidad para evaluar las especificaciones técnicas y funcionales dentro del proceso de desarrollo de software, identificar las posibles amenazas o áreas de debilidad.

Conocimientos de criptología: Encriptación, Cifrado simétrico/asimétrico, Clave pública, Clave privada, etc.

Conocimientoseninfraestructurasweb,cloudcomputingyvirtualización

Conocimientos de ataques web, SQL injection,XSS, XAS, CSRF, LFI/RFI, etc.

Seguridad de Sistemas operativos (windows server/linux/unix).

Conocimientos de antivirus, malware, adware, spyware, riskware, etc.

Conocimientos de seguridad en Redes, Firewall, proxy, filtrado de conexiones, análisis de paquetes, detección de ataques, etc.

Conocimientos de Seguridad Física y aspectos legales.

Conocimientos deseables, según el nivel del puesto

Project Management

Conocimientos de auditoría

Competencias, habilidades o aptitudes deseables según el nivel del puesto

Pensamiento analítico, iniciativa y capacidad de resolución de problemas.

Capacidad organizativa, ateniéndose a un orden propio que le facilite el acceso a lo que pueda necesitar, así como interés en medir el desempeño de los procesos.

Capacidad de trabajar en equipo, con buen manejo de las relaciones interpersonales estando dispuesto a compartir información y conocimientos y a tomar en cuenta a los usuarios.

Orientación a la mejora continua con capacidad para influenciar a la organización.

Ámbito ocupacional

En general, forma parte de un staff cross organizacional que brinda soporte a diferentes áreas: desarrollo, infraestructuras, calidad, web master, seguridad e higiene, auditorías, etc. Su principal área de ocupación es en el sector de IT trabajando conjuntamente con administradores de servidores y redes.

Administrador de Redes, Comunicaciones y Sistemas Operativos

Responsabilidad / Principales competencias

Es responsable por mantener y monitorear la infraestructura de redes de la Organización.

Entre sus principales competencias está:

Instalar, configurar y mantener el funcionamiento de redes informáticas internas y conexiones a redes externas, de acuerdo con los niveles de servicio operacional y de seguridad que se establezcan.

Nombres de cargos similares

Soporte de infraestructura IT, Soporte de redes, Especialista en redes, Técnico de redes, Ingeniero de redes, Administrador de redes, comunicaciones y sistemas operativos.

Misión de la ocupación

Administrar la operación, seguridad y mantenimiento de la infraestructura de redes asegurando la calidad y performance de los servicios de la Organización.

Actividades que realiza

Diseñar o participar en el diseño de redes internas y conexiones con redes externas.

Instalar o participar en la instalación de redes internas y conexiones con redes externas.

Coordinar tareas con proveedores, instaladores y personal interno, de acuerdo con la naturaleza de las actividades.

Configurar redes internas de acuerdo a los requerimientos operativos y de seguridad que se establezcan (enrutamientos, conmutaciones, accesos de usuarios locales y remotos, privilegios, niveles de seguridad, entre otros).

Mantener permanente coordinación con el área de IT, en relación con los servicios de la red, para asegurar la implementación, operación y niveles de servicio, de acuerdo a los requerimientos que se establezcan de las aplicaciones.

Monitorear el funcionamiento de la/las redes internas y las conexiones con redes externas.

Diagnosticar problemas y ejecutar las acciones correctivas que correspondan de acuerdo a la naturaleza del problema (reconfiguraciones, cambios de equipos, componentes, software de red, entre otros).

Identificar, recomendar e implementar actualizaciones relativas a la configuración, equipos y software de red, de redes internas y conexiones con redes externas.

Mantener registro de estadísticas y situaciones de funcionamiento, mantenimiento, reparaciones, problemas de usuario, actualizaciones, de la/las redes de la organización y conexiones con redes externas.

Establecer y mantener stock de equipos, componentes y accesorios, o asegurar su obtención en tiempo y forma, necesarios para asegurar el funcionamiento de la/las redes de acuerdo con los niveles de servicio que se establezcan.

Mantener contacto e interactuar coordinadamente con proveedores y personal interno de la organización, para asegurar eficacia y eficiencia en procesos de compra, mantenimiento, entrenamiento y soporte en general, toda vez que sea necesario.

Identificar y mantenerse actualizada la información de novedades tecnológicas y de gestión, relacionada con redes y de ICT en general.

Estudios

Ingeniero Electrónico, Telecomunicaciones o Informática (Carrera Universitaria afín a las telecomunicaciones).

Dependiendo del nivel de complejidad de las redes que tiene una organización, y del seniority del puesto, Tecnicatura superior en Electrónica, Tecnicaturas secundarias con cursos de formación profesional orientados a redes.

Conocimientos necesarios, según el nivel del puesto

Sólidos conocimientos de fundamentos de redes y protocolos de comunicación.

Sólidos conocimientos de sistemas operativos y servidores de red.

Conceptos y características de redes LAN, WAN, Wireless, VPN, etc.

Sistemas y tipos de cableados para interconectar dispositivos.

Direccionamiento IP, máscaras, Subneteo, para cumplir determinados requerimientos.

Características y operación de protocolos tales como HTTP, Domain Name System (DNS), Dynamic Host Configuration Protocol (DHCP), Simple Mail Transfer Protocol (SMTP), Telnet, and FTP.

Funcionamiento y configuración de enrutadores estáticos y dinámicos. Tablas de enrutamiento.

Características y operación de protocolos Enhanced Interior Gateway Routing Protocol (EIGRP), Open Shortest Path First (OSPF), RIPv1, RIPv2

Tecnologías tales como VLANs, VLAN Trunking Protocol (VTP), Rapid Spanning Tree Protocol (RSTP), Per VLAN Spanning Tree Protocol (PVSTP), and 802.1q

Utilitarios para verificación de operación y análisis de trafico de redes.

Redes Wireless: componentes, implementación, servicios tales como Service Set Identification (SSID), Basic Service Set (BSS), and Extended Service Set (ESS). Seguridad, Protected Access (WPA), Equivalent Privacy (WEP), and WPA-1/2 networks.

Listas de control de acceso (ACL)

Traducción de direcciones, Network Address Translation (NAT).

Competencias, habilidades o aptitudes deseables según el nivel del puesto

Inglés

Planificación y control

Capacidad para detectar riesgos

Visión de negocio

Capacidad y agilidad de aprendizaje (para poder asimilar y procesar nueva información, particularmente cambiante con rapidez en el área de ICT).

Buen nivel de comunicación oral y escrita

Resolución de problemas (identificación y caracterización de problemas, evaluación de opcionespara la resolución, implementación de la solución más conveniente.

Pensamiento crítico (uso de la lógica y razonamiento para enfocar situaciones y escenarios diversos, y para desarrollar y evaluar conclusiones y actuar en consecuencia).

Capacidad de comunicación interpersonal

Capacidad y predisposición para trabajar en equipo.

Ámbito ocupacional

En la actualidad, las TICs están presentes, transversalmente, en prácticamente todas las organizaciones, dando soporte a la mayoría de las actividades que se realizan en los diferentes ámbitos.

En organizaciones grandes (grandes empresas, organismos de gobierno, etc), la estructura de redes y IT son, en general, grandes y complejas, pero la parte de redes suele depender del área de IT. En este contexto, el administrador de redes típicamente tiene una o dos personas a cargo, que se ocupan de la red y también trabajan en equipo con la gente de IT para desarrollar proyectos, implementar servicios y mantener la operación del día a día. En estos ambientes, es prácticamente mandatorio el mantenerse permanentemente actualizado en los temas técnicos y alcanzar certificaciones de la industria que validen y respalden los conocimientos y habilidades adquiridas. En el caso de empresas proveedoras o de servicios, las oportunidades de carrera suelen estar asociadas a actividades comerciales, por ejemplo ingenieros de pre/pos venta, ingeniero de proyectos, gerencia técnica, etc.

En organizaciones pequeñas, la estructura de redes y de IT es, en general, pequeña y de poca complejidad. Las funciones del administrador de red y las funciones del administrador de IT, se concentran en una sola persona que interactúa con los usuarios para todos los temas. Típicamente, esta persona auto-gestiona su formación adquiriendo experiencia en el trabajo diario, estudiando y eventualmente tomando algunos cursos específicos con el/los proveedores de tecnología.

Posibles especializaciones o áreas de especialización

De acuerdo con la tendencia en el desarrollo de tecnologías en TICs y particularmente en el campo de las redes, que evolucionan para atender requerimientos de convergencia de tecnologías, servicios y de operación, se pueden identificar áreas de especialización tales como:

Data Center: Requiere, entre otros, conocimientos y habilidades que incluyen arquitectura de redes, servidores, virtualización, almacenamiento, seguridad, Cloud Computing, aspectos fundamentales de infraestructura (edilicio, energía, climatización, etc), administración y gestión unificada.

Routing and Switching: Profundización en tecnologías de enrutamiento y conmutación, implementación, configuración, para redes complejas.

Video: Tecnologías de procesamiento de video, aplicaciones y dispositivos orientados a video, integración de tráfico de video en las redes, niveles de servicio.

Voice: Comunicaciones unificadas, equipamiento específico para gestión de tráfico de voz (centrales IP) e integración de servicios (voz, datos, video)

Redes Wireless: Tecnologías de transmisión inalámbrica, arquitecturas LAN inalámbrica, estándares de la industria, acceso de dispositivos móviles a las redes inalámbricas, conexión a redes fijas, administración y seguridad de accesos (encripción, protocolos).

Seguridad: Profundización en tecnologías de seguridad en diversas áreas de IT y que convergen sobre las redes. Herramientas para detección de amenazas y análisis. Esquemas de seguridad extremo a extremo y a través de las redes.

Bibliografía

Bass, L., Weber, I., & Liming, Z. (2015). DEVOPS A Software Architect's Perspective. Westford, Massachusetts: Pearson Education Inc.

Bahga, A. y Madisetti, V (2014). Cloud Computing: A Hands-On Approach.

Beyer, Pettof, Jones, Murphy (2016). Site Reliability Engineering: How Google Runs Production Systems. O'Reilly

Brooks, F. (1982). The Mythical man-month : essays on software engineering. Reading, Mass. :Addison-Wesley Pub. Co.

Chiavenato, I. (2006). Introducción a la Teoría General de la Administración. Edición tercera. Mc

Graw Hill.

Chiavenato, I. (2009). Comportamiento organizacional: La dinámica del éxito en las organizaciones 2da Edición. Mc Graw Hill.

Gruver, Mouser, (2015). Leading The Transformation. O'Reilly

Hope, J. y Player, S. (2012). Mejores prácticas de gestión empresarial Como optimizar el control de la gestión, el rendimiento y los costes. Harvard Business Review Press. Profit editorial.

Kim, Behr, Spafford (2013). The Phoenix Project: A Novel About IT, DevOps, and Helping Your Business Win. It Revolution Press

Kniberg, H (2007) Scrum y XP desde las trincheras Cómo hacemos Scrum. InfoQ

McAfee, Westerman, Bonnet (2014) Leading Digital: Turning Technology into Business Transformation. Harvard Business Review Press.

Mell, P. y Grance, T (2011) The NIST Definition of Cloud Computing. National Institute of Standards and Technology. NIST Special Publication

Porter, M. (1985). Competitive Advantage. Creating and Sustaining Superior Perfomance

Ries, E (2012) El método Lean Startup: Cómo crear empresas de éxito utilizando la innovación continua. Deusto.

Schallmo, Daniel y Williams, Christopher (2018) Digital Transformation Now!: Guiding the Successful Digitalization of Your Business Model

Willis, J (2016). The DevOps Handbook: How to Create World-Class Agility, Reliability, and Security in Technology Organizations. It Revolution Press

Zaigham et al. (2014). Cloud Computing: Concepts, Technology & Architecture. Pearson

Glosario de Términos

AWS: Amazon web services

CEO: Chief Executive Officer

CIO: Chief Information Officer

Data Breach: Acceso por partes no autorizadas a datos de acceso restringido. También se suelen usar los términos violación, filtrado o escape de datos

Data Center: centro de cómputos

EA: Enterprise Architect

GCP: Google Cloud Platform

IaaS: Infrastructure as a Service

On-premise: en el local, en las propias instalaciones.

PaaS: Platform as a Service

PM: Project Manager

PO: Product Owner

SaaS: Software as a Service

SLA: Service level Agreement

SLI: Service level Indicator

SLO: Service level Objective

TI: tecnología de la información