

TAREA 1 INGENIERÍA DE SOFTWARE

Juan David Bejarano Taborda

Pontificia Universidad Católica de Chile
Magister en Ciencia de Datos
Ingeniería de Software
Mayo 2023

Ingeniería de Software Tarea 1

01 mayo 2023

1) Descripción general de un proyecto en que hayas participado en lo posible recientemente (preferiblemente, un proyecto de desarrollo de software). Descríbelo resumidamente desde el punto de vista del dominio (o ámbito), propósito, usuarios, equipo de desarrollo (¿roles?), tiempo de desarrollo (estimado, real), resultado (entre exitoso, ..., y hasta fracaso) y por qué lo evalúas así.

Respuesta/

Hace un par de años liderada el equipo de BI que se encargaría de generar insights para la nueva estrategia de transformación digital que estaba implementando la compañía en la cual trabajaba. La iniciativa de transformación digital era el principal macroproyecto de compañía, el cuál involucraba una transformación cultural, pero también la ejecución de proyectos más pequeños, dentro de lo cuales la mayoría implicaba desarrollo de software.

Propósito

El propósito del proyecto era automatizar la atención de un millón de llamadas por año que en ese momento se atendían a través de operadores humanos en el call center a través de la creación de 5 canales digitales (Whatsapp, Chatbot, una Aplicación móvil, el IVR y un nuevo portal web). Salvo el IVR, no existía atención a través de ninguno de los otros canales, por lo que se requería un gran esfuerzo de desarrollo de software. El macro-proyecto tenía un CAPEX de \$2M USD y un plazo de ejecución de 3 años, con medición de avances y de hitos semestrales.

Estructura organizacional y Roles

Squads: El corazón de la estrategia. Los “squads” como los llamábamos eran equipos interdisciplinarios autónomos, que dependiendo del alcance tenían entre 6 y 12 personas (desarrolladores). Cada squad tenía un “scrum master” y un “Product Owner” y se trabajaba bajo metodología scrum.

Tribus: Teníamos 5 tribus, las cuales correspondían a cada uno de los canales digitales que se buscaba desarrollar. El objetivo de las tribus era coordinar el esfuerzo de los squads que conformaban dicha tribu. Existía un líder de tribu que denominábamos “sponsor”, y era el máximo responsable del desarrollo del canal. A su vez, cada tribu tenía 3 agile coaches, quienes se encargaban de acompañar la metodología scrum, así como de validar que las ceremonias, los artefactos y los equipos se estuviesen desempeñando de manera correcta. Tengamos en cuenta que, en este punto, para la organización la metodología scrum era nueva, y la mayoría de las personas (incluidos los scrum master y product owner) nunca habían trabajado bajo este marco, por lo que los agile coaches tenían un rol preponderante en la estrategia.

Chapter: Algunas áreas técnicas (como la mía) nos denominábamos chapters. Correspondían a equipos de personas con conocimientos muy especializados (Analítica y BI por ejemplo), que prestábamos en algunos casos asesoría y metodología, pero también recursos a los squads y a las tribus en tareas específicas. Los chapters eran mucho más parecidos a equipos de IT

tradicionales, con un gerente y distintos desarrollados. Yo era el líder del chapter de BI y Analítica. Dicho chapter desplegaba capacidades como desarrolladores en el modelo Scrum a distintas tribus, con objetivos diferentes.

Tiempos y resultados

Cada una de estas tribus se enfrentaba a desafíos muy complejos, especialmente en la parte tecnología y metodológicamente. No estuve en esa compañía hasta el final de los 3 años, por lo que no podría hablar del resultado final. Pero voy a describir uno de los proyectos en los que trabajamos. Recuerdo particularmente un gran desafío en la tribu de WhatsApp, donde toda la información que se extraía del chatbot estaba en formato JSON. Sin embargo, nadie del equipo era experto en bases de datos No estructuradas y semi-estructuradas. Teníamos un mini-proyecto que correspondía a construir un sistema que lograra convertir dichos archivos JSON en bases de datos estructuradas que respondieran preguntas de negocio. El plazo era 2 meses. Para solucionarlo, tuvimos la necesidad de contratar un experto en el tema, así como herramientas para manejo de este tipo de información (Bases de datos NO SQL). Por supuesto esto supuso un sobre costo y un incumplimiento de las fechas pactadas. El hito correspondiente a la extracción y almacenamiento de información de WhatsApp logró llegar a buen término después de 5 meses de retraso y un sobre costo cercado a los \$14.000 USD.

En cuanto a si fue exitoso o no, podría decirse que se logró resolver, pero no dentro de tiempo y el costo presupuestado. Este tipo de situaciones pueden resultar muy habituales en los proyectos de desarrollo de software. Para alguien que este consciente que esto puede pasar y que es normal que pase, podría definir el proyecto como exitoso. Pero dependerá de la mirada que cada quién le de.

2) Supuestos (dos o tres) a partir de los cuales se llevó adelante el proyecto; para cada uno [Supuesto: Algo considerado real o verdadero sin la seguridad de que lo sea]; para cada supuesto: - cuál fue el supuesto y por qué se hizo (p.ej., había alguna evidencia / se le preguntó la opinión a un experto / si no se hacía el supuesto, entonces el proyecto no se podía llevar a cabo / etc.) - ese supuesto, ¿resultó cierto? En caso afirmativo, ¿de qué manera se benefició el proyecto gracias a que el supuesto era cierto? En caso negativo, ¿qué implicaciones tuvo en el desarrollo/resultado del proyecto el hecho de que el supuesto no se cumplió?

Respuesta/

A continuación, voy a describir algunos de los supuestos sobre los que se trabajó el macro-proyecto:

1. Los clientes estarán dispuestos a utilizar canales digitales de autogestión en lugar de llamar al centro de llamadas.
2. Los canales digitales de autogestión serán fáciles de usar y comprender para la mayoría de los clientes.
3. Los canales digitales de autogestión serán capaces de aumentar la capacidad de trabajo que es atendido en el call center.
4. La automatización reducirá el tiempo y los costos asociados con la gestión de llamadas.

5. Los empleados del centro de llamadas estarán dispuestos y capacitados para manejar otros tipos de tareas una vez que se automatice la gestión de llamadas.
6. Los sistemas informáticos y de tecnología necesarios para la automatización de las llamadas serán implementados sin problemas y sin retrasos significativos.
7. No se producirán interrupciones significativas en el servicio al cliente durante la transición de un centro de llamadas a canales digitales de autogestión.
8. La información personal del cliente estará segura y protegida durante todo el proceso de automatización.
9. La automatización de llamadas mejorará la satisfacción del cliente y, por lo tanto, la lealtad del cliente hacia la empresa.

Estos supuestos parecen lógicos. Y no fueron únicamente ideas propias. Estos supuestos se encontraban alineados con el socio estratégico que se contrató para soportar este macro-proyecto: McKinsey, una de las consultoras más grandes y prestigiosas del mundo. Existe evidencia de muchas compañías que han ahorrados millones de dólares anualmente automatizando los canales digitales de autogestión. Nos mostraron decenas de casos de uso en Norteamérica y Europa sobre la automatización de la atención con excelentes resultados en experiencia y satisfacción.

Sin embargo, vivimos en una realidad latinoamericana que difiere completamente de lo que puede ocurrir en otras latitudes. A medida que íbamos iterando en el desarrollo de los canales de atención, nos dimos cuenta de que muchas personas no estaban dispuestas adoptar estos canales digitales y preferían seguir llamando al call center. Algunas de ellas porque lo encontraban difícil (supuesto 2) pero otro simplemente por resistencia a la edad, por motivos de edad y afinidad tecnológica, entre muchas otras razones. Al menos los 2 primeros años, los supuestos de disminución de llamadas no se alcanzaron.

Así mismo, los supuestos de que la satisfacción del cliente en estos canales se iba a mantener tampoco se cumplieron. Las encuestas de satisfacción por canal mostraban que la gente prefería ser atendida por un ser humano, y la satisfacción de los canales auto asistidos dejaba mucho por desear.

Los supuestos presupuestales y de tiempo, como mencioné en el primer punto, en muchas ocasiones no se cumplieron. Como resultado de todo esto, el proyecto tuvo que cambiar de supuestos. Las llamadas que esperaban ahorrar, probablemente no se darían. Ciertas transacciones no era convenientes automatizar y era mejor mantenerlas dentro del servicio de atención telefónico. Si bien el proyecto siguió adelante, y se podía generar valor, la evidencia e iteración mostraba que era necesario seguir bajo nuevos supuestos, o supuestos ajustados.

3) Requisitos del proyecto: - ¿cómo fue el proceso de levantamiento de los requisitos? - ¿cómo resultó la experiencia de definición de requisitos con sus usuarios? - ¿cómo documentaron los requisitos? - ¿cambiaron los requisitos al transcurrir el tiempo?

Respuesta/

Los requisitos funcionales eran levantados primero por el líder de la tribu, quién a su vez los refinaba y socializaba con los Product Owner de cada squad. Para definirlos, se invitaban muchas veces usuarios externos, quienes planteaban sus necesidades. Luego, los requisitos eran almacenados y declarados en el “Product Backlog”.

Una vez allí, y luego de definir las prioridades, se realizaba la planificación del sprint. Los requisitos eran documentados en las “historias de usuario”. A la vez, se documentaba algo conocido como el “Definition of Done (DOD)”, que declaraba al final de cada sprint cómo se vería un incremento correcto. Se podría decir que en el DOD se establecía el plan de pruebas para el incremento. En esta ceremonia, los desarrollados manifestaban qué podía ser desarrollado y qué no, y se establecía la estimación para ser desarrollado.

Este era un proceso que de hecho funcionaba muy bien para los requisitos funcionales. Sin embargo, no se era igual de diligentes con los requisitos no funcionales del desarrollo. Un ejemplo de esto fue el lanzamiento del módulo de facturación a través de la Web. Todos los requisitos funcionales salieron a producción de acuerdo con lo definido. Sin embargo, no se tuvo en cuenta la facilidad de uso para el cliente como uno de los requisitos no funcionales. A pesar haber cumplido todos los requisitos funcionales, 2 meses después la compañía se dio cuenta que la satisfacción de los clientes con la solución era muy baja, no se habían diseñado los mecanismos para medirlo ni para recibir feedback, y fue necesario un reproceso del desarrollo. Estos requisitos no contemplados tuvieron que agregarse en una segunda fase, después de haber invertido muchas capacidades en el primer lanzamiento.