

Universidad de Buenos Aires
Facultad de Ciencias Económicas
Escuela de Estudios de Posgrado

Maestría en Gestión Estratégica de Sistemas
y Tecnologías de la Información

TRABAJO FINAL DE MAESTRIA

**La calidad del software desde la perspectiva del
análisis de requerimientos**

AUTOR: ING. VIVIANA SARANGO

DIRECTOR: DR. ALBERTO TERLATO

JULIO 2021



Universidad de Buenos Aires
Facultad de Ciencias Económicas
Escuela de Estudios de Posgrado



Agradecimientos

Agradezco a Dios porque me permitió cumplir una meta más en mi vida y en especial agradecimiento al Dr. Alberto Terlato que con gran generosidad puso a disposición toda su experiencia, consejos y valiosos aportes para la realización de este trabajo.



Resumen

Existen muchos factores que determinan el éxito de una empresa u organización, uno de ellos es contar con información certera (interna y externa) para pensar estratégicamente y definir las acciones a tomar. Actualmente las organizaciones están poniendo mucho énfasis en el software como herramienta competitiva, especialmente en su vinculación con el cliente por proporcionar ello información valiosa para el desarrollo de su negocio. Desde el lado de sus proveedores de software impera la necesidad del pleno entendimiento de las necesidades y del problema a resolver y la importancia de lograr hacerlo con la solución adoptada, porque ésta será una forma de medir la calidad de su producto y por ende resolver los requerimientos de la empresa. Relativo a ello, la ingeniería de requerimientos es uno de los procesos más importantes, por ser allí donde se define el diseño de la solución según las necesidades del cliente. Esta investigación se propone indagar sobre los desafíos que enfrentan las empresas desarrolladoras de software al momento de realizar el relevamiento de los requerimientos. Se buscará indagar sobre las influencias que genera la falta de una perspectiva sistémica, la carencia de herramientas y buenas prácticas y la presencia de formas de pensamiento grupal. Busca, asimismo, elaborar una propuesta del uso de herramientas como el modelado de procesos de negocio y definiciones, metodologías y técnicas necesarias para interpretar los requerimientos. La investigación ha contemplado el desarrollo de una exhaustiva exploración bibliográfica, la cual ha incluido fuentes primarias, libros y *papers*, junto a un trabajo de campo, de tipo cualitativo con el objetivo de capturar información de las experiencias del sector corporativo (clientes y proveedores).

Palabras claves: Requerimientos, Ingeniería de Software. Ingeniería de requerimientos, calidad software, proyectos tecnológicos.



INDICE

AGRADECIMIENTOS	2
RESUMEN.....	3
JUSTIFICACIÓN.....	11
PLANTEAMIENTO DEL TEMA/PROBLEMA	12
OBJETIVOS	13
HIPÓTESIS	14
MARCO TEÓRICO	15
METODOLOGÍA	<u>2322</u>
CAPÍTULO I. PROCEDIMIENTOS SISTÉMICOS Y EVALUACIÓN DE REQUISITOS	<u>2423</u>
1.1 CALIDAD DEL SOFTWARE	<u>2524</u>
1.2 CALIDAD SISTÉMICA.....	<u>2524</u>
1.3 GARANTÍA DE CALIDAD	<u>2625</u>
1.3.1 Importancia de los estándares de software.....	<u>2625</u>
1.3.2 Áreas Críticas en la Organización.....	<u>2726</u>
1.4 ¿QUÉ ES EL ÉXITO DE UN PROYECTO?.....	<u>2827</u>
1.4.1 Factores de éxito de proyectos de software	<u>2928</u>
1.5 ¿POR QUÉ SE NECESITAN LOS REQUISITOS?	<u>3332</u>
1.6 CLASIFICACIÓN DE LOS REQUERIMIENTOS	<u>3332</u>
1.6.1 Requerimientos de Negocio.....	<u>3433</u>
1.6.2 Requerimientos de los stakeholders	<u>3433</u>
1.6.3 Requerimientos funcionales	<u>3433</u>
1.6.4 Requerimientos No-funcionales	<u>3433</u>
1.6.5 Requerimientos de Transición.....	<u>3433</u>



1.7	CARACTERÍSTICAS DE LOS REQUERIMIENTOS.....	3635
1.8	IMPORTANCIA DE LA DEFINICIÓN FORMAL DE REQUERIMIENTOS.....	3736
1.9	CONCLUSIÓN DEL CAPÍTULO	3938
CAPÍTULO II. TÉCNICAS Y METODOLOGÍAS DE LOS REQUERIMIENTOS ...		4039
2.1	INTRODUCCIÓN	4140
2.2	TÉCNICAS DE LEVANTAMIENTO DE REQUERIMIENTOS:.....	4241
2.2.1	Lluvia de ideas	4342
2.2.2	Descomposición funcional	4342
2.2.3	Reglas de negocio	4342
2.2.4	Prototipos.....	4342
2.2.5	Etnografía u observación.....	4443
2.2.6	Entrevista	4443
2.2.7	Análisis de interfaz:	4443
2.2.8	Análisis de documentos.....	4544
2.2.9	Grupo de enfoque (focus group)	4544
2.2.10	Taller de requerimientos	4544
2.2.11	Modelado de procesos.....	4645
2.2.12	Descubrimiento de escenarios	4645
2.3	ROLES EN EL LEVANTAMIENTO DE REQUERIMIENTO.	4746
2.3.1	El autor	4746
2.3.2	Los Expertos.....	4746
2.3.3	El Moderador	4746
2.3.4	El escritor de la minuta.....	4746
2.3.5	Participantes	4847
2.3.6	Autorizante	4847
2.4	METODOLOGÍAS EN LA GESTIÓN DE LOS REQUERIMIENTOS.....	4847
2.5	INFLUENCIA DE LAS METODOLOGÍAS EN LA GESTIÓN DE LOS PROYECTOS.....	4948
2.6	CONCLUSIÓN.....	5251



CAPÍTULO III. BUENAS PRÁCTICAS Y MAPAS MENTALES	5352
3.1 INTRODUCCIÓN	5453
3.2 QUE ES UNA BUENA PRÁCTICA	5453
3.2.1 Mejores Prácticas	5453
3.2.2 Atributos esenciales para determinar Mejores Practicas	5655
3.2.3 Guía de escritura de buenas practicas	5857
3.2.4 Requisito como proceso de buenas practicas	5958
Paso 5: Gestión de requisitos:	6059
3.3 MAPAS MENTALES EN LA INGENIERÍA DE REQUERIMIENTOS	6059
3.3.1 Ley de mapas mentales y recomendación de mapas mentales	6261
3.3.2 Mapa mental grupal	6261
3.3.3 MP Colaborativo para Facilitar el Proceso de ingeniera de Requerimientos	6463
3.4 CULTURA ORGANIZACIONAL	6564
3.4.1 Niveles de cultura organizacional	6665
3.4.2 Modelo y cambio cultural	6665
3.5 CONCLUSIÓN	6766
CAPÍTULO IV. EL RASGO DE FUERTE PRESENCIA DE PENSAMIENTO GRUPAL	6968
4.1 INTRODUCCIÓN	7069
4.2 COHESIÓN GRUPAL	7170
4.3 GROUPTHINK	7170
4.4 PRIORIZACIÓN ORGANIZATIVA COMO "PENSAMIENTO DE GRUPO"	7372
4.5 PENSAMIENTO GRUPAL EN LA INGENIERÍA DE SOFTWARE	7574
4.6 GROUPTHINK EN INGENIERÍA DE SOFTWARE	7675
4.6.1 Sobreestimación	7675
4.6.2 Mentalidad cerrada	77
4.6.3 Presión hacia la uniformidad	7877
4.7 MÉTODOS PARA ABORDAR EL PENSAMIENTO GRUPAL	78
4.7.1 Asignar papel de crítico	7978



4.7.2	Los líderes evitan hacer sugerencias	7978
4.7.3	Evaluación independiente	7978
4.7.4	Hacer sugerencias anónimas	7978
4.8	CÓMO EVITAR EL PENSAMIENTO GRUPAL	8079
4.9	CONCLUSIÓN.....	8281

CAPITULO V. PARTICIPACIÓN DEL CLIENTE Y HERRAMIENTAS DE RELEVAMIENTO

.....		8483
5.1	INTRODUCCIÓN	8584
5.2	LA IMPORTANCIA DE LA PARTICIPACIÓN DEL CLIENTE EN LOS PROYECTOS DE SOFTWARE.....	8685
5.2.1	Mayor claridad de los requisitos del cliente y priorización puntual. ..	8685
5.2.2	Menos desperdicio financiero y de recursos.....	8685
5.2.3	Refinamiento más cuidadoso del software a través de demostraciones y pruebas.	8685
5.3	PROBLEMAS COMUNES CON LA PARTICIPACIÓN DEL CLIENTE DURANTE UN PROYECTO DE SOFTWARE	8786
5.4	QUE ES CUSTOMER EXPERIENCE	8887
5.4.1	La experiencia del cliente móvil es una prioridad.....	8887
5.4.1	La frustración del cliente provocara deserción.....	8988
5.5	HERRAMIENTAS DE RECOPIACIÓN DE REQUISITOS.....	9089
5.6	¿POR QUÉ SON IMPORTANTES LAS HERRAMIENTAS DE RECOPIACIÓN DE REQUISITOS?	9089
5.7	LAS 10 MEJORES HERRAMIENTAS DE RECOPIACIÓN DE REQUISITOS ..	9190
5.8	¿CÓMO ELEGIR LA MEJOR HERRAMIENTA PARA SU EQUIPO?	9594
5.9	CONCLUSIÓN.....	96

CAPÍTULO VI. LABOR DE CAMPO Y CONCLUSIONES

6.1	INTRODUCCIÓN	99
-----	--------------------	----



6.2	HIPÓTESIS	100
6.3	INVESTIGACIÓN DE CAMPO	100
6.3.1	<i>Introducción</i>	100
6.3.2	<i>Composición del colectivo estudiado</i>	101
6.3.3	<i>Principales conclusiones del trabajo de campo</i>	102
6.4	CORROBORACIÓN DE HIPÓTESIS	103
6.5	CONCLUSIONES	114 112
6.6	FUTURAS LÍNEAS DE INVESTIGACIÓN	115 114
BIBLIOGRAFÍA		117 115
ANEXO		¡ERROR! MARCADOR NO DEFINIDO. 124



Listado de Tablas

TABLA 0.1 ESTADÍSTICAS DE PROYECTOS.	<u>1746</u>
TABLA 0.2 COMPARACIÓN ENTRE LA ORGANIZACIÓN CLÁSICA Y LA SISTÉMICA	<u>2220</u>
TABLA 1.0.1. FACTORES DE FALLA O CANCELACIÓN EN LOS PROYECTOS DE SOFTWARE	<u>3029</u>
TABLA 1.0.2. FACTORES DE ÉXITO DE PROYECTOS DE SOFTWARE	<u>3130</u>
TABLA 1.0.3. USUARIOS Y DESARROLLADORES	<u>3837</u>
TABLA 2.0.1 INFLUENCIA DE LAS METODOLOGÍAS EN LA GESTIÓN DE PROYECTOS	<u>4948</u>
TABLA 2.0.2 DIFERENCIAS ENTRE METODOLOGÍAS AGILES Y TRADICIONALES	<u>5049</u>
TABLA 2.0.3. VENTAJAS ENTRE METODOLOGÍAS AGILES Y TRADICIONALES	<u>5150</u>
TABLA 4.1 CONDICIÓN DE GROUPTHINK	<u>7274</u>
TABLA 4.2. CARACTERÍSTICAS DE GROUPTHINK	<u>7274</u>
TABLA 4.3 TÉCNICAS GRUPALES	<u>8180</u>
TABLA 5.1 COMPOSICIÓN DE LA MUESTRA	<u>10287</u>



Listado de Figuras

FIGURA 0.1 TRIÁNGULO DE HIERRO	<u>1746</u>
FIGURA 1.0.1. REPORTE DEL CAOS PARA LOS AÑOS 2011-2015	<u>3234</u>
FIGURA 1.0.2. CLASIFICACIÓN DE REQUERIMIENTOS	<u>3534</u>
FIGURA 3.3.1 MAPA MENTAL DE INGENIERA DE REQUERIMIENTOS	<u>6160</u>
FIGURA 5.1 CAUSAS DE FALLAS DE LOS PROYECTOS	<u>10489</u>
FIGURA 5.2 PROYECTOS EXITOSOS Y FRACASADOS	<u>10590</u>
FIGURA 5.3 PROYECTOS EXITOSOS	<u>10590</u>
FIGURA 5.4 COSTO DE LA CORRECCIÓN DE FALLOS ORIGINADOS EN REQUISITOS	<u>10694</u>
FIGURA 5.5 METODOLOGÍAS EN LOS PROYECTOS	<u>10792</u>
FIGURA 5.6 METODOLOGÍA ÁGIL VS TRADICIONAL	<u>10893</u>
FIGURA 5.7 TAMAÑO DE LOS PROYECTOS	<u>10994</u>
FIGURA 5.8 UTILIZACIÓN DE BUENAS PRÁCTICAS	<u>11095</u>
FIGURA 5.9 UTILIZACIÓN DE MAPAS MENTALES	<u>11095</u>



Justificación

Las organizaciones constantemente trabajan por obtener ventajas competitivas usando las variadas herramientas que se han desarrollado para enfrentar a la competencia. Al respecto, Ghosh & Daugherty (2015), sostiene que el software es ahora un factor determinante de la diferenciación y la innovación. Es una puerta hacia nuevos servicios y flujos de ingresos, experiencias continuas para clientes y expansión hacia nuevos mercados. Para tener éxito en medio de esta disrupción, las compañías deben responder cambiando la manera en que diseñan, desarrollan y utilizan el software.

Este trabajo final de maestría busca sacar conclusiones entre dos aspectos de la gestión de desarrollo de software para empresas. Por un lado, en la necesidad que tienen éstas de desarrollar ventajas competitivas -en un contexto cada vez más complejo- a partir de la calidad, entre otras, la de su software, a fin de diferenciarse de sus competidores evitando competir por precio, publicidad y promoción. Por el otro, desde el lado de los proveedores de software, cómo estos pueden desarrollar ventajas competitivas en sus propios procesos de análisis de requerimientos para hacer más eficientes los costos, los tiempos de entrega y la calidad, cumpliendo con las necesidades de los clientes.

La investigación busca justificación en el propósito de ayudar a las empresas proveedoras de software a afrontar la problemática del levantamiento de información dentro de sus clientes.

Desde la perspectiva del autor y su experiencia en la industria del software este trabajo pretende responder a múltiples problemas encontrados en proyectos a lo largo de su experiencia en los últimos años.



Planteamiento del tema/problema

Según Brooks (1995) la parte más difícil de construir un sistema software es decidir qué construir. Ninguna otra parte del trabajo afecta más negativamente al sistema final si se realiza de manera incorrecta. Ninguna otra parte es más difícil de rectificar.

El mismo autor señala que en la actualidad muchos proyectos de software fracasan porque no se realiza un estudio previo de los requisitos del usuario, no se hace una definición completa del alcance del proyecto o no se realiza el modelado del negocio antes de desarrollar el software. Esto significa que el analista no se embebe en el problema; aunque tenga claro que el sistema debe desarrollarse para dar soporte a los procesos de la organización. Plantear el tema desde sus propias conjeturas -endogámicas- puede introducir el riesgo de que los requisitos identificados no correspondan a las necesidades para lo que se debe crear.

Es importante entonces que no existan requerimientos indefinidos o mal definidos. Este trabajo, busca abordar la visión del colectivo en estudio, analizando las metodologías y herramientas de software, brindando lineamientos para poder ejercer un control de la gestión del levantamiento de requerimientos mediante procesos para lograr especificaciones correctas que describan con claridad, sin ambigüedades y en forma compacta las necesidades del cliente, y por ende alineadas a las necesidades del negocio. Abordará las siguientes cuestiones:

¿Cómo lograr determinar que metodología y prácticas utilizar al momento de realizar el relevamiento de requerimientos en la organización?

¿Cómo lograr evaluar la estabilidad y capacidad de los procesos actuales en el negocio?



Objetivos

El objetivo principal de este trabajo final de maestría será evaluar el proceso de levantamiento de requerimientos de software bajo tres perspectivas de análisis, la visión sistémica, utilización de buenas prácticas y herramientas e inexistencia del pensamiento grupal, considerando a estos factores como variables explicativas del desempeño de las empresas productoras de software en el proceso de determinación de requerimientos, vinculando sus resultados al tiempo de entrega, adecuación a las necesidades del cliente y calidad. Complementariamente esta investigación el autor pretende desarrollar una propuesta para la gestión de requerimientos.

Objetivos Específicos

1. Analizar el estado actual de las principales metodologías y estándares existentes a nivel internacional en la gestión de requerimientos enmarcados en una visión sistémica que permita un mejor discernimiento y decisión al área de TI.
2. Definir la vinculación de la calidad del proceso de relevamiento de requerimientos a la utilización de buenas prácticas y mapas mentales de apoyo.
3. Analizar la presencia del pensamiento grupal en los equipos de trabajo que impida alcanzar buenos resultados a nivel de calidad, adecuación, tiempos y costos del proyecto de software.
4. Determinar la importancia de la participación del cliente en el relevamiento y examinar las principales herramientas que existen en la actualidad, permitiendo maximizar el nivel de comunicación, calidad y resultados del proyecto.



Hipótesis

En este trabajo de maestría se intenta demostrar lo siguiente:

H1. La falta de un procedimiento sistémico de recopilación y evaluación de requisitos para el desarrollo de software puede llegar a introducir fallas de calidad, alcance y agregar tiempos y costos a un proyecto de software.

H2. La falta de herramientas y metodologías de gestión de proyectos en el desarrollo de software puede llegar a introducir fallas de calidad, alcance y agregar tiempos y costos a un proyecto de software.

H3. La no utilización de buenas prácticas de ingeniería de requerimientos y mapas mentales de apoyo al proceso de análisis y diseño de software puede introducir fallas de calidad, alcance y agregar tiempos y costos a un proyecto de software.

H4. Los rasgos de presencia de pensamiento grupal en el grupo de diseño pueden conspirar contra la obtención, en el análisis y diseño de software, de buenos resultados a nivel de calidad, adecuación y tiempos y costos del proyecto de software.

H5. La falta de participación del cliente en el proyecto y el uso de herramientas obsoletas que no se adaptan a los cambios disruptivos de la transformación digital que se está atravesando actualmente, puede acarrear fallas en el triángulo de hierro.



Marco teórico

Pressman (2010) estima que, del total de proyectos software grandes, ¹ un 28 % fracasan, un 46 % caen en severas modificaciones que los retrasan y sólo un 26 % son totalmente exitosos. Cuando un proyecto fracasa, rara vez es debido a fallas técnicas, la principal causa gira alrededor de la falta o carencia de aplicación de una buena metodología o proceso de desarrollo al momento de relevar la información. Entre otras, se puede observar, desde hace pocas décadas, una fuerte tendencia a mejorar las metodologías o procesos de desarrollo, o crear nuevas y concientizar a los profesionales de la informática a su utilización adecuada.

Dicho autor sostiene ser común para el desarrollo de software de mediano porte ² que los equipos humanos involucrados apliquen «metodologías propias», normalmente un híbrido de los procesos anteriores y a veces con criterios propios.

Otros de los aspectos a tener en cuenta se alinean con la dirección del recurso humano. La forma en que como resultado de ello las personas puedan o no involucrarse, comprometerse. Los factores intrínsecos deben ser comprendidos e incorporados a la agenda. Es común observar en las empresas una excesiva apelación a motivadores extrínsecos como pulsares del desempeño. Las empresas deben fijar metas medibles, realistas y realizables, las personas deben poder alcanzarlas; la organización y los jefes deben preocuparse por el desarrollo de las personas para hacer posible que éstos alcancen los objetivos trazados. Las metas y resultados alcanzados deben ser reconocidos, los comportamientos destacados no pueden quedar sin distinguirse (Terlato, 2014).

Herzberg (1987) propone que el trabajo debe ser enriquecido, esto es deben reemplazarse factores de carga horizontal, orientados a fraccionar las tareas para lograr foco y eficiencia, por otros de carga vertical alineados a desarrollar el autocontrol y empoderamiento, asimismo incrementar los niveles de responsabilidad de la función (consistentes con la formación y el desarrollo), delegación de derechos decisorios, y

¹ YAMAL CHAMOUN, define proyecto como un conjunto de esfuerzos temporales, dirigidos a generar un producto o servicio único. Donde el tiempo de esfuerzo sea mayor a 2 años, aunque este es relativo

² YAMAL CHAMOUN, Los proyectos que son menores a dos años, aunque es relativo



asignación progresiva de tareas cada vez más complejas.

Ingeniería de Requerimientos

Requerimientos de Software

Según el Standards Coordinating Committee of the Computer Society of the IEEE (1990), diccionario estándar de terminología de Ingeniería de Software, los requerimientos de software pueden definirse como:

- (1) Una condición o capacidad requerida por un usuario para resolver un problema o alcanzar un objetivo.
- (2) Una condición o capacidad que debe ser poseída por un sistema o componente de éste para satisfacer un contrato, un estándar, una especificación u otro tipo de documento formalmente impuesto.
- (3) Una representación documentada de una condición o capacidad según 1 y 2.

Según la metodología que se utilice, existen varias clasificaciones de los tipos de requerimientos una de ellas es la que distingue a los requerimientos entre funcionales y no funcionales (Sandoval Carvajal, y otros, 2008).

Requerimientos funcionales. Involucra la declaración de los servicios que proveerá el sistema, de la manera en que éste reaccionará a entradas particulares y de cómo se comportará en situaciones determinadas. Pueden declarar lo que el sistema no debe hacer: requerimientos no funcionales. Son aquellos que no se refieren directamente a las funciones específicas que entregará el sistema, sino a las propiedades emergentes de éste tales como la fiabilidad, el tiempo de respuesta, la capacidad de almacenamiento, etc. En general estos apuntan al sistema como un todo y no a rasgos particulares, por lo tanto, una falla de un requerimiento no funcional, podría inutilizar un sistema (Sandoval Carvajal & García Vargas, 2008)

El desarrollo de presente trabajo estará focalizado en los requerimientos funcionales únicamente.



El triángulo de hierro

Kerzner (2013) señala que para que un proyecto sea denominado exitoso debe cumplir con lo que él llama triángulo de hierro, el cual implica lograr cumplir con el tiempo previsto, el costo proyectado y el alcance requerido, tal y como se ilustra en la figura 1. El informe CHAOS, de Standish Group (2009), es el informe más famoso sobre el fracaso de los proyectos en el sector de las tecnologías de la información (TI). Con él comienzan innumerables charlas de ingeniería del software en numerosas instituciones. Este suele realizarse cada dos años y su última edición es de abril de 2009. Dentro de su contenido puede que los proyectos software a 2009 presentaban una tasa de éxito del 32%, frente al 35% del informe de 2006 y al 16% del de 1994.

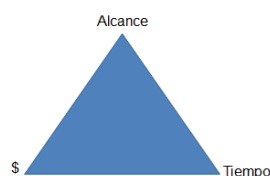
Tabla 0.1 Estadísticas de Proyectos.

	1994	1996	1998	2000	2002	2004	2006	2009
Éxito	16%	27%	26%	28%	34%	29%	35%	32%
Deficientes	53%	33%	46%	49%	51%	53%	46%	44%
Fracasos	31%	40%	28%	23%	15%	18%	19%	24%

Fuente: Informe CHAOS 2009

El 44% de los proyectos fueron deficientes (con retraso, por encima del presupuesto y/o con menos de los requisitos esperados) mientras que el 24% fracasaron, se cancelaron o se finalizaron, pero el producto nunca se usó.

Figura 0.1 Triángulo de Hierro



Fuente: Kerzner (2013)



No todos los gerentes están convencidos de que su equipo necesita hacer un mejor trabajo en el desarrollo y la gestión de requisitos, o que esforzarse en ello dará sus frutos. Sin embargo, numerosos estudios de la industria indican que los problemas de requisitos son una causa generalizada a la angustia del proyecto. Los informes de CHAOS de The Standish Group (2009), indican que tres de los factores que contribuyen en mayor medida a que los proyectos fracasen pasan por la falta de cuestionamientos y comentarios del usuario y requisitos y especificaciones incompletos y cambiantes.

Numerosos estudios han examinado los efectos de los errores en los requisitos en proyectos de software. Siempre encuentran que casi la mitad de los defectos descubiertos se originaron como errores de requisitos. El resultado típico es la brecha en las expectativas, una diferencia entre lo que los desarrolladores crean y lo que los clientes realmente necesitan y esperan. Claramente, cualquier dominio que sea la causa raíz de aproximadamente la mitad de los problemas en proyectos de software merece la atención (Wiegers, <https://www.jamasoftware.com/>, 2012).

Buenas Prácticas

Un conjunto de principios de diseño y buenas prácticas a implantar en el ciclo de vida de desarrollo de software (SDLC), tendrá el propósito de detectar, prevenir y corregir los defectos de seguridad en el desarrollo y adquisición de aplicaciones, de forma que se obtenga software de confianza y robusto frente a ataques maliciosos, que realice solo las funciones para las que fue diseñado, que esté libre de vulnerabilidades, ya sean intencionalmente diseñadas o accidentalmente insertadas durante su ciclo de vida y se asegure su integridad, disponibilidad y confidencialidad (Figuerola, 2016).

Sawyer (1997) el cual define la madurez de un proceso de requerimientos como el grado en que una organización tiene su proceso de Ingeniería de requerimientos basado en buenas prácticas.

Sommerville y Sawyer (1997) sugieren un conjunto de directrices detalladas para la Elicitación de Requisitos:

- Evaluar la viabilidad comercial y técnica para el sistema propuesto.
- Identificar las personas que ayudarán a especificar los requisitos y a comprender sus



prejuicios organizacionales.

- Definir el entorno técnico por ejemplo la arquitectura computacional, el sistema operativo en el que funcionara el sistema o producto.
- Identifica las “restricción de dominio” es decir, las características del entorno empresarial específico para el dominio de aplicación que limitan la funcionalidad o el rendimiento del sistema o producto que se desarrolla.
- Definir uno o más métodos de Elicitación de Requisitos, por ejemplo: entrevistas, grupos focales, reuniones de grupo.
- Solicitar la participación de muchas personas para que se definan los requisitos desde diferentes puntos de vista: asegurarse de identificar la base lógica de cada requisito que se registra.
- Identificar requisitos ambiguos como candidatos a prototipar.
- Crear escenarios de uso para ayudarles a los clientes/usuarios a identificar mejor los requisitos.

Pensamiento Grupal

Pensamiento de grupo (*groupthink* en inglés) es un término acuñado por el psicólogo Irving Janis en 1972 para describir el proceso por el cual un grupo puede tomar decisiones malas o irracionales. En una situación de pensamiento de grupo, cada miembro intenta conformar su opinión a la que creen que es el consenso del grupo. En un sentido general, esto parece ser una manera muy racional de afrontar la situación. Sin embargo, deriva en una situación riesgosa en la cual el grupo establece el conjunto de creencias a las cuales sus integrantes individualmente deben acoplarse y considerar aconsejable. Janis (1972) lo define como “Un modo de pensamiento que las personas adoptan cuando están profundamente involucradas en un grupo cohesivo, cuando los esfuerzos de los miembros por unanimidad hacen caso omiso de su motivación para valorar realísticamente cursos de acción alternativos” (p. 155).

Es decir, las personas afectadas por el pensamiento grupal deciden de acuerdo a las creencias del grupo y no sus propias creencias. Esto provoca que en muchas ocasiones se tomen decisiones erróneas, con todas sus consecuencias. Decisiones que podrían haberse evitado con una simple opinión en disidencia, alguien que creía que la decisión a tomar era incorrecta y



proponía un camino alternativo. El pensamiento grupal anula la divergencia. Quien lo hace puede ser presionado, incluso descalificado (Janis, 1972).

Perspectiva Sistémica

El pensamiento sistémico facilita el contribuir a la guía personal y dirección de otros con eficacia. A nivel empresarial permite comprender la complejidad de los procesos y descubrir la forma para mejorarlos; a su vez crear y dirigir equipos, que al fin y al cabo funcionan como sistemas de tipo complejo que no pueden ser gobernados por sistemas de menor complejidad. (Senge, 1990).

Según Goleman la mente sistémica (1998, Pag 350) “es la capacidad para reconocer los sentimientos propios y los de los demás, motivarnos a nosotros mismos, para manejar acertadamente las emociones, tanto en nosotros mismos como en nuestras relaciones humanas”

Análisis de Requerimientos de Sistemas

Simultáneamente al desarrollo de la ingeniería en sistemas en la década de 1950 emergió la vertiente de pensamiento metodológico conocido como “Análisis de Sistemas”, un desarrollo asociado especialmente con la corporación RAND, que se dedica a la consultoría de negocios. En un reporte de RAND, Hitch (1955) proporciona un informe que tiene muchas similitudes, con la metodología de la ingeniería de sistema, y que estaba emergente al mismo tiempo. Los elementos esenciales se describen así:

- a. Un objetivo u objetivos que se desea alcanzar
- b. Técnicas alternativas o conductos mediante los cuales se puede alcanzar el objetivo.
- c. Los costos o recursos que requiere cada sistema
- d. Un modelo o modelos matemáticos
- e. Un criterio que relaciona los objetivos y los costos o recursos para elegir la alternativa óptima.

El enfoque de sistemas significa que el modo de abordar los objetos y fenómenos no puede ser aislado, sino que tiene que verse como parte de un todo. No es la suma a los elementos sino un conjunto de elementos que se encuentran en interacción de forma integral que produce



Universidad de Buenos Aires
Facultad de Ciencias Económicas
Escuela de Estudios de Posgrado



nuevas cualidades con características diferentes cuyo resultado es superior al de los componentes que lo forman un provocan un salto de calidad. (Olivera, 2010)



Una organización en aprendizaje

Ciertas empresas tradicionales, jerárquicas, no confían en la capacidad, el compromiso y la responsabilidad de los empleados, por lo que gestionan implementando sistemas de control y obediencia. Los jefes mandan y los colaboradores obedecen. (Terlato, 2019). Trabajar desde esta perspectiva puede ser ineficiente para trabajar en un proceso de identificación de requerimientos toda vez que podría llevar a los analistas a trabajar sólo en aquello que les ha sido especialmente consignado, aunque creyeran que eso pueda ser parcial o incorrecto.

Tabla 0.2 Comparación entre la organización clásica y la sistémica

Organización Clásica	Organización Sistémica
✓ Énfasis exclusivamente individual y el cargo de la organización ✓ Relación del tipo autoridad-obediencia ✓ Adhesión rígida a la delegación y responsabilidad compartida. ✓ División del trabajo y supervisión jerárquica rígidas. ✓ Toma de decisiones centralizada ✓ Control rígidamente centralizado ✓ Solución de conflictos mediante represión arbitrariamente y hostilidad. ✓ Jerarquía Estricta: Por que las decisiones dependen de una sola persona ✓ Dogmatismo: Porque se basan en procedimientos y conceptos invariables.	✓ Énfasis de las relaciones inter e intergrupales ✓ Confianza y credibilidad recíprocos. ✓ Interdependencia y responsabilidad compartida ✓ Participación y responsabilidad multigrupal ✓ Toma de decisiones descentralizada ✓ Responsabilidad y control ampliamente compartidos ✓ Solución de conflictos a través de negociaciones o solución de problemas ✓ Asunción de responsabilidades: cada persona asume una determinada responsabilidad ✓ Flexibilidad intelectual

Fuente: (Rosnay, 1975)



Metodología

El estudio corresponderá a una investigación de tipo exploratoria y cualitativa, apoyando en mecánica del caso instrumental y con captura de información a partir de entrevistas, que busca comprender la fenomenología, las variables y sus relaciones.

Serán entrevistados personas del sector, directores y analistas a cargo de la determinación de los requerimientos. Se trabajará con una guía de preguntas proyectivas y situacionales, donde el entrevistado opinará a partir del colectivo que representa.

El objetivo de esta labor será comprender el complejo mundo de la experiencia vivida desde el punto de vista de las personas que la viven y trabajan en estas tareas y empresas (Taylor y Bogdan, 1984).

Se pretende conocer el contexto y proceso donde se produce el análisis de los requerimientos en las empresas estudio buscando encontrar evidencias.

Se abordará un análisis de las diferentes situaciones y escenarios que se presentan dentro del proyecto al momento de relevar la información, asimismo proyectos exitosos y otros que no cumplieron con los requerimientos del triángulo de hierro.



Universidad de Buenos Aires
Facultad de Ciencias Económicas
Escuela de Estudios de Posgrado



Capítulo I. Procedimientos Sistémicos y Evaluación de Requisitos



1.1 Calidad del Software

La calidad ha sido una preocupación desde hace mucho en los negocios, y debe serlo también para los ingenieros de requisitos en el análisis y diseño de sistemas de información. El principal objetivo de los ingenieros del software es producir un sistema, aplicación o producto de alta calidad, para lo cual emplean métodos y herramientas efectivas dentro del contexto de un proceso maduro de desarrollo del software y además deben desarrollar mediciones que den como resultado sistemas de alta calidad. Para obtener esta evaluación, el ingeniero debe utilizar medidas técnicas, que evalúan la calidad con objetividad (Osorio & Castro , 2011,pp. 65-69).

Según Pressman (2005) la calidad es la concordancia del software producido con los requerimientos explícitamente establecidos y con los estándares de desarrollo prefijados, con los requerimientos implícitos no establecidos formalmente, que desea el usuario. Otra definición que presentan Vega, Rivera & García (2008) en su libro y que es propuesta por la organización internacional de estándares (ISO/IEC DEC 9126,2002), señala que “La totalidad de características de un producto de software que tienen como habilidad, satisfacer necesidades explícitas o implícitas” (p. 150).

1.2 Calidad Sistémica

Humphrey (1997) indica que la calidad del proceso garantiza la calidad del producto y consecuentemente no se pueden desligar estos dos aspectos; tener modelos separados capaces de medir individualmente la calidad de un producto o de un proceso de software, no garantiza la relación sistémica que debe estar presente entre ellos. Esto se debe a que la naturaleza de los sistemas no puede ser dividida en partes, sino que debe existir una interdependencia y colaboración entre las partes (proceso y producto) para que el mismo sea visto como un todo (Perez , Griman , & Mendoza, 2004).



1.3 Garantía de calidad

Humphrey (1997) considera que la calidad es definida como el grado de relación que tiene el producto para satisfacer las necesidades del usuario. Un software que cumple con todos los requisitos con su usuario y procesos que se ejecuten correctamente, garantiza una buena Calidad.³

Para ello se debe considerar diferentes tipos de estándar:

Estándares de producto. Se aplican sobre el producto software que se comienza a desarrollar. Incluye los de documentación, codificación.

Estándares de proceso. Define los procesos que deben seguirse durante el desarrollo de software. Incluyen definiciones relativas a la especificación, diseño y validación, así como una descripción de los documentos que deben escribirse durante los procesos.

1.3.1 Importancia de los estándares de software

Humphrey (1997) define que los estándares de un producto guardan relación estrecha con los estándares de un proceso. Los estándares de producto por su parte se aplican a la salida del proceso de software, mientras que los de proceso incluyen actividades que garantizan que se sigan los estándares de producto.

Los estándares de software:

- a) Se basan en el conocimiento de la mejor práctica de la empresa.
- b) Proveen el marco de trabajo alrededor del cual se implementa el proceso de garantía de la calidad.
- c) Permite que una persona continúe con facilidad el trabajo iniciado por otra y el utilizar buenas prácticas por todo el personal de la organización reduce el esfuerzo de aprendizaje cuando se inicia un nuevo proyecto.

³ Sommerville señala: El uso de aseguramiento de calidad a lo largo del proceso es una forma de minimizar los riesgos y ayuda a asegurar que el sistema resultante es lo que se necesita y quiere, y que mejorará evidentemente algunos aspectos del desempeño del negocio



1.3.2 Áreas Críticas en la Organización

En un informe presentado por Project Management Institute⁴ (PMI, 2014) señala que, por cada dólar gastado en proyectos y programas, un 5,1 por ciento se desperdicia debido a la mala gestión de requisitos. Poner en términos más llamativos, esto equivale a 51 mil dólares desperdiciados por cada \$ 1 mil millones gastados. eso es un gran valor potencial que cae a través de las grietas en el mundo y no a proyectos. Para cambiar el rumbo y permitir que los proyectos sean más exitosos y los resultados vayan de acuerdo al negocio, las organizaciones - según esta institución- deben centrarse mucho más la atención en tres áreas críticas que pueden mejorar en gran medida la eficacia de sus capacidades de gestión de requisitos:

Personas - Las organizaciones deben poner el recurso necesario para aplicar correctamente la gestión de requisitos para recomendar soluciones a los proyectos y programas, al mismo tiempo, también deben reconocer la gestión de requisitos para recomendar soluciones a los proyectos y programas, y a reconocer y desarrollar las habilidades necesarias para realizar estas funciones.

Procesos - Las organizaciones deben estandarizar y formalizar sus procesos a nivel de proyectos y programas, para asegurarse que están aplicando consistentemente buenas prácticas de gestión de requisitos para todas sus iniciativas.

Cultura. - Las organizaciones deben hacer ver a las directivas el carácter urgente de esta necesidad, de modo que el equipo ejecutivo y los patrocinadores valoren por completo la práctica como una competencia crítica de los proyectos y programas, y brinden el apoyo y el compromiso debido, necesario para sobresalir en toda la organización.

⁴ PMI El Project Management Institute es una organización estadounidense sin fines de lucro que asocia a profesionales relacionados con la Gestión de Proyectos.



1.4 ¿Qué es el éxito de un proyecto?

Éxito, para Real academia española (2017), es el “resultado feliz de un negocio o de una actuación”, también puede ser “la buena aceptación que tiene alguien o algo”, además puede significar “el fin o terminación de un negocio o asunto”.

Sin embargo, para los investigadores y estudiosos, es difícil encontrar un consenso. Siendo algunas veces un término ambiguo o con significados diferentes. Esta ambigüedad puede estar afectado por la perspectiva de los interesados, por cuestiones temporales, por el tipo de proyecto u organización autores como Nara y Besteiro, citados por Balaguera Reina, Cabrera Angulo y Cuadros Guataquira (2017), han contribuido a evitar esta confusión definiendo al éxito de la gerencia de proyectos cuando se produce el cumplimiento de la triple restricción, de alcance, tiempo y costo.

Shenhar, Levy & Dvir, citados en Ruiz Arias (2015), señalan que el éxito del proyecto es probablemente la idea discutida con más frecuencia en el campo de la gestión de proyectos, y sin embargo también es la que tiene menos consenso tiene en su definición. A pesar de esto, hace más de dos décadas que los investigadores trabajan para identificar las variables que determinan el éxito en la gestión.

Prabhakar, citado por Ruiz Arias (2015), afirma que ni los profesionales ni los académicos han llegado a estar de acuerdo sobre lo que constituye el éxito del proyecto. Para este autor es un concepto bastante difícil de definir. Y esto puede explicarse por qué el éxito del proyecto puede tener un significado diferente para cada persona dependiendo de su punto de vista o experiencia. De esta forma puede tener una percepción diferente de aquello que es un proyecto exitoso o no, como así lo señalan Liu & Walker, citados por Ruiz Arias (2015).

Pinto & Slevin, citados por (Ruiz Arias, 2015), ofrecen dos razones para la indefinición del concepto. La primera -coincidente con la anterior -que todavía no está claro cómo medir el éxito del proyecto, ya que las partes que están involucradas en el mismo perciben el éxito o el fracaso de un proyecto de manera diferente y por lo tanto valorarán los resultados de manera distinta. La segunda es que las listas de los factores que influyen en el éxito o en el fracaso varían mucho en la bibliografía escrita sobre el tema. En esa misma línea Shenhar, Tishler, Lipovetsky, & Lechler, citados por Ruiz (2015), sugieren tres razones para esa falta de definición: por un lado, el enfoque universalista utilizando en la mayoría de los estudios de gestión de proyectos; por otra parte el hecho de considerar que todos los proyectos poseen una



idea subjetiva similar de lo que son los factores que determinan el éxito. Finalmente que el número de variables de gestión sea limitado y examinado por investigaciones previas no aplicables a cada caso concreto.

Para despejar esta ambigüedad del concepto es imprescindible a criterio de estos autores- diferenciar entre dos tipos de éxito, el éxito interno y el externo. El primero hace referencia a la organización, al equipo de trabajo, y por lo tanto a toda la gestión del proyecto desde el inicio de la vida del producto (el proyecto), la ejecución y entrega del mismo al usuario final. El segundo se refiere a los clientes o usuarios finales, siendo de tal importancia la satisfacción de estos, e involucra responder a sus requerimientos y que en la vida total del producto no tenga defectos ni fallos.

Munns & Bjeirmi, citados por Ruiz (2016), postulan que esta ambigüedad seguirá existiendo si no se establece distinción entre el éxito del proyecto y el éxito de la gestión del proyecto. El éxito del proyecto sería aquel que puede observarse a largo plazo con un proyecto terminado (éxito externo), mientras que, por el contrario, el éxito de gestión del proyecto es algo más concreto y se refiere a la planificación y el control en un contexto a corto plazo de la elaboración y ejecución de un proyecto (éxito interno).

Saenz Arteaga (2012), proponen que es importante, hacer una distinción entre lo que es “éxito de la gestión de proyectos” y los “proyectos exitosos” para poder buscar relaciones relevantes al concepto en la investigación.

Autores como Atkinson, & Badot, Westerveld, citados por Saenz Arteaga (2012), mencionan que el éxito de los proyectos ha sido largamente considerado como la capacidad de ajustarse a las restricciones características que se impone al proyecto como el tiempo, costo y calidad. Salir victorioso es para estos autores cumplir con este denominado triángulo de hierro.

1.4.1 Factores de éxito de proyectos de software

Sorprende que la industria del software es una actividad caracterizada por los problemas de calidad en sus productos y servicios. La investigación más citada sobre el estado de los proyectos de software es el famoso “Reporte del Chaos” que publica The Standish Group (1994) y que sirve de referencia para los investigadores. The Standish Group clasifica a los proyectos según su finalización en tres tipos:

- Exitoso (*Successful*) – El proyecto se completó a tiempo y dentro del presupuesto, con



todas las características y funciones requeridas.

- Deficiente (*Challenged*) – El proyecto se completó y es operacional, pero utilizó recursos más allá del presupuesto, concluyó más allá del tiempo estimado y/o cuenta con pocas de las características y funciones que fueron especificadas inicialmente.
- Fracasado (*Failed*) – El proyecto es cancelado antes de completarse o es implementado, pero nunca utilizado por los usuarios.

Este reporte citado sostiene que el 31.1 % de los proyectos terminaron cancelándose es decir fracasaron y el 52.7% culminaron por encima de su presupuesto y fuera de los plazos establecidos es decir con deficiencias. Solo el 16.2% se pueden clasificar como exitosos. De lo apuntado queda resaltado que el porcentaje de proyectos fallidos o cancelados suma el 84.1% del total de proyectos. Es necesario por tanto revisar las razones de ello.

Tabla 1.0.1. Factores de Falla o Cancelación en los Proyectos de Software

Factores de Deficiencias o cancelación	%
a) Requerimientos incompletos	13.1
b) Deficiencia en el involucramiento del usuario	12.4
c) Deficiencia de recursos	10.6
d) Expectativas no realistas	9.9
e) Deficiencia en soporte ejecutivo	9.3
f) Cambios en los requerimientos y especificaciones	8.7
g) Deficiencia en la planeación	8.1
h) Ya no se necesita más	7.5
i) Deficiencia en administración de TI	6.2
j) Desconocimiento en tecnología	4.3
k) Otros	9.9

Fuente: The Standish Group, 1994

En la Tabla 1.0.1. se observa que, las dos primeras causas de fallo o fracaso de los proyectos de software son: Los requerimientos incompletos y la falta de involucramiento



del usuario. Pero adicionalmente los factores d), f), y h) en alguna medida, también están relacionados con las deficiencias en el establecimiento de los requerimientos del software.

De otro lado, también es importante revisar los factores que influyen en el éxito de un proyecto. The Standish Group. (1994) presentó los principales de estos factores que se detallan en la Tabla 1.0.2. donde se observa que la participación del usuario, el apoyo de los ejecutivos y la declaración clara de los requisitos forman parte de los tres factores más importantes para el éxito de un proyecto.

Tabla 1.0.2. Factores de Éxito de Proyectos de Software

Factores de Éxito	%
Participación del usuario	15.9%
Apoyo de gestión ejecutiva	13.9%
Declaración clara de requisitos	13.0%
Planificación adecuada	9.6%
Expectativas realistas	8.2%
Hitos más pequeños del proyecto	7.7%
Personal competente	7.2%
Propiedad	5.3%
Visión clara y objetivos	2.9%
Trabajo duro, personal enfocado	2.4%
Otro	13.9%

Fuente: The Standish Group ,1994

En la Tabla 1.0.2. The Standish Group ha actualizado el “Reporte del Chaos”, en desde 1994, hasta el 2018, la última versión publicada. En estos informes observa un incremento en los porcentajes de los proyectos de software exitosos y una reducción en los proyectos defectuosos y cancelados. En el Reporte del Chaos del año 2015 publicado por InfoQ (2015), muestra un avance en el porcentaje de éxito de los proyectos respecto al año 2014, con un 29% de todos los proyectos (entregado a tiempo, dentro del presupuesto, con las características y funciones necesarias), el 52% fueron comprometidos, cuestionados o sufrieron cambios (fuera



de tiempo, por encima del presupuesto, y/o con menos de las características requeridas y las funciones) y el 19% fracasaron (cancelado antes de su finalización o entregado y nunca se usa). Estas cifras representan un avance de 1% en proyectos exitosos respecto al estudio anterior. Los proyectos que tuvieron algún tipo de problema se redujeron en 3% respecto al estudio anterior y hubo un incremento de 2% en proyectos fracasados, también teniendo como referencia el estudio anterior. Basándose en las estadísticas contenidas en el informe se puede indicar que no existe mayor variación respecto a los informes anteriores (2011 al 2015). Pero es oportuno mencionar que hubo un incremento en el porcentaje de proyectos fallidos. Estos datos pueden observarse en la Figura 1.0.1.

Figura 1.0.1. Reporte del Caos para los años 2011-2015

MODERN RESOLUTION FOR ALL PROJECTS					
	2011	2012	2013	2014	2015
SUCCESSFUL	29%	27%	31%	28%	29%
CHALLENGED	49%	56%	50%	55%	52%
FAILED	22%	17%	19%	17%	19%

Fuente: InfoQ (2015)

Como se ha visto en el “Reporte de Chaos”, publicado por The Standish Group (2014), los tres principales pilares para el éxito de los proyectos de software son: el respaldo de la dirección, la participación del usuario y los objetivos de negocio claros, y como se verá más adelante, conocer los objetivos del negocio lo cual permita elaborar buenos requisitos.

La situación actual de la industria del software indica que aquel problema llamado “Crisis del Software” que venía afectando desde mediados del siglo pasado a los grandes proyectos de software aún no ha sido superado. Esto se evidencia porque conforme avanza la tecnología y las opciones metodológicas también se presentan mayores exigencias a nivel de usabilidad, portabilidad, seguridad, interoperabilidad y agilidad en la entrega de los productos. Convirtiéndose en factores que un encargado de proyecto debe controlar para asegurar el éxito de los proyectos (Canchari, 2018).



1.5 ¿Por qué se necesitan los requisitos?

El Instituto Nacional de Tecnologías de la Comunicación de Madrid (INTECO, 2008) en su informe destaca que los requisitos poseen influencia en los siguientes factores:

- **Costo.** El costo de una buena recolección de requisitos y análisis del sistema a desarrollar es menor que el resultante de tener requisitos pobres, es decir, el emergente de reparar productos deficientes o de poca calidad, los proyectos cancelados y de haber perdido un negocio o cliente.
- **Procesos de ingeniería de requisitos.** El fundamento básico de cualquier software recae sobre su proceso de ingeniería de requisitos. El éxito o fallo del software depende casi siempre de cómo y cuánto bien se hayan capturado, entendido y usado los requisitos como base para el desarrollo. La ingeniería de requisitos es la fase de la ingeniería del software donde se definen las propiedades y la estructura del software. La ingeniería de requisitos comprende el desarrollo y gestión de requisitos de forma tal de:
 - ✓ Entender los requerimientos del negocio, identificar los requisitos del usuario y trasladar ambos a requisitos de sistema/software.
 - ✓ Gestionar los cambios de requisitos y mantener la consistencia entre éstos y otros productos de trabajo del proyecto.

1.6 Clasificación de los Requerimientos

La Guía sobre Los fundamentos del conocimiento del Análisis de Negocio⁵ (BABOK, 2015) contiene una descripción de prácticas generalmente aceptadas en el campo temático. El contenido incluido ha sido verificado por profesionales, encuestas a la comunidad y respuestas a consultas con expertos reconocidos en el campo. la Guía BABOK ha sido reconocida alrededor del mundo como una herramienta clave para la práctica del Análisis de Negocio y se ha convertido en un estándar ampliamente aceptado para la profesión. A continuación, detallaremos la clasificación que propone el BABOK:

⁵ BABOK Una guía para el análisis empresarial es un estándar para la práctica del análisis empresarial.



1.6.1 Requerimientos de Negocio

Comprende las definiciones a nivel de las metas, objetivos o necesidades de la empresa. Describen las razones del porque un proyecto ha sido iniciado, los objetivos que el mismo debe alcanzar y las métricas que serán utilizadas para medir su éxito. Los requerimientos del negocio describen necesidades de la organización como un todo y no los de grupos de usuarios o *stakeholder* dentro de ella. Son desarrollados y definidos a través del análisis empresarial.

1.6.2 Requerimientos de los stakeholders

Comprenden las definiciones de los *stakeholder*. Describen sus necesidades y su forma de operar con una solución. Los requerimientos de éstos sirven como puente entre los requerimientos del negocio y las diversas clases de requerimientos de la solución. Son definidos a través del análisis de requerimientos.

1.6.3 Requerimientos funcionales

Estos describen el comportamiento e información que la solución debe administrar. Describen capacidades que el sistema deberá poder llevar a cabo en términos de su comportamiento u operación, asociado con acciones de respuesta específicas a las aplicaciones de tecnologías de la información.

1.6.4 Requerimientos No-funcionales

Estos capturan las condiciones que no están directamente relacionadas con el comportamiento o funcionalidad de la solución, por el contrario, describen condiciones del medio ambiente en las cuales la solución deberá permanecer eficaz o cualidades que el sistema debe tener. Son también conocidos como requerimientos de calidad o suplementarios. Esto pueden incluir requerimientos relacionados con capacidad, velocidad, seguridad, disponibilidad, la arquitectura de la información y la presentación de la interfaz de usuario.

1.6.5 Requerimientos de Transición

Estos describen las capacidades que la solución debe tener con el fin de facilitar la transición del estado actual de la empresa hacia el estado futuro deseado. Serán requeridos cuando la transición sea llevada a cabo. Se diferencia de otros tipos de requerimientos porque son siempre de naturaleza temporal y porque no podrán ser desarrollados hasta que ambas, la solución existente y la nueva sean definidas. Habitualmente cubren la conversión de datos existentes, brechas en las habilidades que deben ser atendidas y otros cambios relacionados



para alcanzar el estado futuro deseado. Son desarrollados y definidos a través de la Evaluación y validación de la solución

Lo anteriormente expuesto se puede representar, de una mejor manera, en base a un modelo detallado en la Figura 1.2. Un concepto de requerimientos, apiramidado, partiendo de un enfoque de negocio hasta llegar a un enfoque de solución en donde cada uno de ellos tiene un objetivo bien definido.

Figura 1.0.2. Clasificación de Requerimientos



Fuente: Caballero, A. (2015). Psicología con Alfonso. Recuperado de <http://gabrielalmeida.com.mx/617/entendiendo-el-babok-en-espanol-capitulo-1-introduccion-1-3-3-requerimientos>



1.7 Características de los requerimientos

Los requerimientos permiten que los desarrolladores expliquen cómo han entendido lo que el cliente pretende del sistema. También indican a los diseñadores qué funcionalidad y qué características va a tener el sistema resultante. Además, indican al equipo de pruebas qué demostraciones llevar a cabo para convencer al cliente de que el sistema que se le entrega es el que solicitó. Las características de los requerimientos mencionados en el estándar IEE830⁶ son desarrollado por Pfleeger (2008), a partir de la siguiente taxonomía:

- ✓ **Deben ser correctos.** Tanto el cliente como el desarrollador deben revisarlos para asegurar que no poseen errores.
- ✓ **Deben ser consistentes.** Dos requerimientos son inconsistentes cuando es imposible satisfacerlos simultáneamente.
- ✓ **Deben estar completos.** El conjunto de requerimientos está completo si todos los estados posibles, cambios de estado, entradas, productos y restricciones están descritos en alguno de los requerimientos.
- ✓ **Deben ser realistas.** Todos los requerimientos deben ser revisados para asegurar que son posibles.

El mismo estudio se plantea la pregunta de si ¿Cada requerimiento debe describir algo que es necesario para el cliente?

Propone al respecto de que los requerimientos deben ser revisados para conservar sólo aquellos que inciden directamente en la resolución del problema del cliente. Por tanto:

- ✓ **Deben ser verificables.** Poderse preparar pruebas que demuestren que se han cumplido los requerimientos.
- ✓ **Deben ser rastreables.** Se puede rastrear cada función del sistema hasta el conjunto de requerimientos que la establece.

⁶ El estándar **IEEE 830**-1998 para el SRS(en inglés) o ERS (Especificación de requerimientos de software) es un conjunto de recomendaciones para la especificación de los requerimiento o requisitos de software el **cuál** tiene **como** producto final la documentación de los acuerdos entre el cliente y el grupo de desarrollo



1.8 Importancia de la definición formal de requerimientos

El análisis y especificación de requerimientos puede parecer una tarea relativamente sencilla, pero las apariencias engañan. Puesto que el contenido de comunicación es muy alto, abundan los cambios por mala interpretación o falta de información. El dilema con el que se enfrenta un ingeniero de software puede ser comprendido repitiendo la sentencia de un cliente anónimo: "Sé que crees que comprendes lo que piensas que he dicho, pero no estoy seguro de que entendiste lo que yo quise decir." (Pfleeger, 2002, p.116).

En la Tabla 1.0.3 se ilustra el conflicto que encontró Scharer (1990) cuando los desarrolladores y los usuarios se limitan a ver el problema desde su particular punto de vista sin tomar en cuenta la situación del otro.

Existen muchos contribuyentes a la determinación del conjunto de los requerimientos. Cada uno tiene una visión particular del sistema de cómo debe funcionar, a menudo estas visiones son conflictivas. Una de las muchas habilidades de un analista de requerimientos es la capacidad para comprender cada punto de vista y capturar los requerimientos de una manera que refleje los intereses de cada participante.



Tabla 1.0.3. Usuarios y Desarrolladores

Como ven los desarrolladores a los usuarios.	Como ven los usuarios a los desarrolladores.
Los usuarios no saben lo que quieren	Los desarrolladores no comprenden las necesidades operacionales.
Los usuarios no pueden articular lo que quieren.	Los desarrolladores ponen demasiado énfasis en la técnica.
Los usuarios tienen muchas necesidades motivadas políticamente.	Los desarrolladores pretenden decirnos como hacer nuestro trabajo.
Los usuarios lo quieren todo bien y ahora	Los desarrolladores no pueden traducir nuestras necesidades claramente establecidas a un sistema exitoso.
Los usuarios son incapaces de priorizar sus necesidades.	Los desarrolladores dicen “no” todo el tiempo.
Los usuarios rehúsan tomar responsabilidades por el sistema	Los desarrolladores siempre están por encima del presupuesto
Los usuarios son incapaces de proporcionar un enunciado utilizable de las necesidades.	Los desarrolladores siempre están atrasados.
Los usuarios no están comprometidos con los proyectos de desarrollo de sistemas.	Los desarrolladores piden a los usuarios tiempo y esfuerzo, aún en detrimento de sus obligaciones primarias importantes.
Los usuarios no tienen voluntad de colaborar.	Los desarrolladores establecen estándares no realistas para la definición de los requerimientos.
Los usuarios no pueden mantener el cronograma.	Los desarrolladores son incapaces de responder rápidamente a los legítimos cambios de las necesidades.

Nota: Fuente: (Scharer, 1990).



1.9 Conclusión del capítulo

La industria de software viene desarrollándose en los últimos años y con ello la necesidad de producir software de calidad. Como se puede ver en el informe de CHAOS los proyectos en los últimos años están mejorando, pero el porcentaje de proyectos no exitosos aún es muy alto, esto se debe a que existen muchos componentes para el éxito de la implementación del proyecto todos ellos interrelacionados y es evidente que la gestión de requisitos es uno de los más críticos, y cuando se hace mal, las consecuencias pueden ser graves en términos de calidad, tiempo de entrega y costo.

Es impórtate entonces identificar que son los requisitos y porque necesitan claros desde el inicio del proyecto, ya que curiosamente la queja más repetida entre los desarrolladores que codifican bajo supuestos tayloristas, es la de que los usuarios no saben lo que quieren o que no piensan en los requisitos antes de demandarlos, ni tampoco evalúan el impacto que suponen los cambios.

Por el contrario, la protesta de los usuarios va en el sentido contrario, poniendo de manifiesto la rigidez y las barreras que encuentran a la hora de gestionar el cambio en los sistemas de software.



Universidad de Buenos Aires
Facultad de Ciencias Económicas
Escuela de Estudios de Posgrado



Capítulo II. Técnicas y Metodologías de los Requerimientos



2.1 Introducción

De la organización del proceso de levantamiento de requerimientos depende el éxito y los buenos resultados en la implementación de las aplicaciones (Terstine, 2015), el generar una estructura que organice las respuestas y necesidades sistémicas de los interesados, garantizan la proyección y el modelado de los requerimientos (Pérez, Salamando y Valencia, 2012), haciendo posible que la efectividad aumente en las metodologías propuestas y, por consiguiente, aumente la satisfacción en los clientes o usuarios. Es clave para el éxito del desarrollo de software la correcta identificación y administración de los requerimientos. Cuando se realiza su levantamiento se debe establecer un límite a las prestaciones que solicita el cliente, ya que muchas veces ello comprende pedidos que no son viables, ya sea por restricciones de tiempo o presupuesto. Y aunque la tecnología puede solucionar la mayoría de los problemas que se les puedan presentar, esto no significa que ésta a sea la única variable que intervenga en la solución (Masooma Yousuf et al., 2015).

Es por ello, que en algunas ocasiones se tiene que hacer ver al cliente que no es posible realizar todo lo que solicita. De esta manera se podrá entregar el proyecto en tiempo y sin superar el presupuesto establecido, ya que, si desde un principio no se muestran estas restricciones y se dejan para fases posteriores, se pone en riesgo el desarrollo del sistema. Adicionalmente, se debe dejarse en claro que la gente que se verá perjudicada no es únicamente el cliente, ya que los desarrolladores del proyecto muchas veces terminan cubriendo parte del presupuesto faltante contra su costo para poder cubrir algunas características que no fueron analizadas bien en la primera fase del desarrollo (Lopez & Cuello de Ávila, 2016).

Otro punto adicional pero muy importante, es cuidar la imagen de la empresa que está desarrollando el proyecto, ya que, por querer ganar un cliente, puede prometerse el desarrollo de los requerimientos careciendo de una buena planeación, y cuando se alcanza la fecha límite para el proyecto y no se cubran todos los requerimientos, el cliente no estará satisfecho con la solución, el tiempo y presupuesto se verán afectados, Esto podrá hacer que se termine perdiendo un cliente para en futuros desarrollos, además de afectar la imagen de la empresa desarrolladora. Para ello los analistas encargados de realizar el levantamiento de requerimientos deben de establecer los alcances que se desean de manera razonable, tomando en consideración el tiempo, el presupuesto y el talento con el que se cuenta.



Asimismo, se debe convencer al cliente que, si algún requerimiento que se solicita, no se puede lograr con las restricciones actuales, ofreciendo una alternativa que satisfaga la necesidad, con características diferentes (Amaya, 2013).

El reto principal de la empresa desarrolladora es mostrar al cliente que siempre tiene la razón (estrategia clara de venta), sin embargo, se le deben ofrecer alternativas que satisfagan sus necesidades cuando no tenga claro lo que quiere (Delgado y Erly, 2008).

2.2 Técnicas de Levantamiento de Requerimientos:

Para tener un mejor resultado antes de realizar una sesión de levantamiento de requerimientos, se debe preparar la sesión de trabajo, en la que se debe investigar a qué se dedica el cliente, cuál es el alcance que tiene su empresa, es decir, si es local, si tiene sucursales, el número de personas que trabaja en ella, entre otras. Lo anterior ayudará a tener una idea global de lo que el cliente va a solicitar, y poder aprovechar el tiempo de la sesión en resolver cuestiones que sean relevantes para el sistema (Amaya, 2013).

En lugar de preguntar al cliente cuál es su objetivo para el proyecto, es mejor preguntar cuáles son sus problemas, ya que de esta forma se podrá obtener más información que sea relevante para el levantamiento de los requerimientos, y se abre una comunicación más simple entre ellos, sin la necesidad de tener que hablar de cuestiones técnicas, ni otras que tengan que ver con la interfaz de la solución. Muchos clientes cuando están en esta etapa, en lo primero que piensan es cómo quieren la interfaz, y no le dan importancia a los requerimientos relevantes. Así cuando se entrega el sistema, el cliente pide agregar nuevas funcionalidades porque en el levantamiento de requerimientos no se descubrieron. Esto ocasiona problemas de presupuesto y tiempo en el desarrollo (Terstine, 2015).

Lopez & Cuello de Ávila (2016) mencionan que las sesiones de levantamiento de requerimientos deben consistir en la propuesta de soluciones para cada una de las necesidades del cliente, ya que el objetivo de éstas únicamente busca realizar el levantamiento. Hay que evitar cometer el error de proponer algo que sea complicado ya que el cliente podría dar por hecho que se va a llevar a cabo. Algo muy importante es identificar son los requerimientos



críticos del sistema. Estos son las definiciones más importantes para el cliente. Si se enfoca sólo en las cuestiones técnicas complejas existe el riesgo de que no se termine a tiempo el sistema.

A continuación, se mencionan un conjunto de técnicas para el levantamiento de requerimientos propuesta en la guía BABOK® en la emisión del año 2009.

2.2.1 Lluvia de ideas

Esta técnica no tiene gran dificultad de aplicación. Permite generar ideas, a partir de una lluvia de propuestas de los participantes. El personaje más importante en esta técnica es el moderador, ya que él es el responsable de que la gente hable y de que todos participen sin que opción alguna sea descartada. Algunas herramientas para facilitar esta técnica son: los mapas mentales y diagramas de contexto, sin embargo, algo importante que se debe tener en cuenta es no realizar el mapa o el diagrama en el transcurso de la reunión, ya que se podría perder el foco central del tema, perdiendo la atención de la gente.

2.2.2 Descomposición funcional

Ayuda a descomponer procesos o áreas en partes más pequeñas, ya que es más fácil comprender función por función, que entender un todo. De esta manera se facilita la comprensión de lo que el cliente solicita.

2.2.3 Reglas de negocio

Estas son las políticas o restricciones, que rigen sobre la forma de cómo funciona la organización. Sobre éstas se debe basar el desarrollo del sistema de información.

2.2.4 Prototipos

Cuando los desarrolladores o usuarios no están seguros acerca de los requisitos, la construcción de un prototipo, aun cuando sea parcial, es posible, para tener conceptos más tangibles.

Esta técnica detalla los requisitos de la interfaz de usuario y se unifica con otros obtenidos en las entrevistas, escenarios y reglas de negocio. Los beneficios que esta técnica ofrece es permitir que los usuarios visualicen cómo será el sistema mediante bosquejos o



modelado de alto nivel del sistema, haciendo más cómoda y simple la explicación al usuario en referencia a cómo será el sistema. Además, un prototipo puede ser desechable, convirtiéndose en un medio barato para descubrir y confirmar los requerimientos que van más allá de la interfaz, tales como procesos, datos y reglas de negocio. Dependiendo de la complejidad del sistema, la creación de prototipos para el levantamiento de requerimientos puede llevar un tiempo considerable y puede que se estanque en las especificaciones de diseño, dejando a un lado los requerimientos que el sistema debe cumplir (Karl Wieggers, 2013).

2.2.5 Etnografía u observación

La observación es una técnica que permite entender los requerimientos organizacionales. El analista tiene que adentrarse en el entorno laboral donde se utilizará el proyecto a desarrollar y observar el trabajo diario, anotando las tareas reales en las que los involucrados (Stakeholders) se desarrollan.

Estas técnicas generalmente son relativamente costosas, pero son altamente instructivas ya que ilustran muchas tareas de los usuarios y los procesos (Fairley & Bourque, 2014).

2.2.6 Entrevista

Tienen como objetivo obtener información de los involucrados, ya sea llevada adelante con una o un grupo de personas, en las cuales el que el entrevistador se encarga de realizar preguntas pertinentes acerca del sistema y documenta las mismas. Esta técnica es simple, se apoya del usuario(s) y/o cliente del sistema, lo que permite mantener una mejor relación con ellos. No debe ser utilizada como un medio para llegar a un consenso entre un grupo de personas, ya que se realiza de manera individual a cada uno de los involucrados. También se debe considerar que dependiendo del nivel de claridad proporcionado durante la entrevista la documentación resultante puede estar sujeta a la interpretación del que realizó la entrevista, por lo que la recopilación y análisis de los datos de la misma puede ser un proceso complejo y costoso (Karl Wieggers, 2013a).

2.2.7 Análisis de interfaz:

Esto se utiliza para identificar las interfaces entre las soluciones y/o componentes de la solución y definir cómo van a interactuar estos componentes. Esta identificación temprana



de las interfaces proporciona una visión de alto nivel para la planificación del desarrollo. Sin embargo, no proporcionará información sobre otros aspectos del sistema ya que el análisis no se realiza integrando los componentes internos (Karl Wieggers, 2013b).

2.2.8 Análisis de documentos

Se utiliza cuando el objetivo es reunir detalles de los sistemas ya existentes, incluyendo las reglas de negocio, entidades y atributos que deben ser incluidos en el nuevo sistema o que necesitan ser actualizados para el sistema actual. La ventaja está en que no se parte desde cero y se aprovechan los elementos existentes para descubrir y/ o confirmar los requerimientos. Sin embargo, se corre el riesgo de que la documentación no esté actualizada o no sea válida. La localización de información relevante puede ser tediosa y consumir bastante tiempo (Karl Wieggers, 2013c).

2.2.9 Grupo de enfoque

Esta técnica permite obtener ideas u opiniones acerca de un producto o servicio. Aquí los participantes comparten sus impresiones, preferencias y necesidades, guiados por un moderador. Esto ahorra tiempo ya que, en menos sesiones de entrevista, se puede alcanzar el objetivo. Pero puede dificultarse a la hora de programar el grupo para la misma fecha y hora de la reunión (Karl Wieggers, 2013d).

2.2.10 Taller de requerimientos

Puede utilizarse para identificar, definir, priorizar y alcanzar el cierre sobre los requisitos del sistema. Un taller de requerimientos es una de las maneras más eficientes de encontrar éstos de una forma rápida, ya que permite promover la confianza, un entendimiento mutuo y una buena comunicación entre los involucrados en el sistema. Puede ser una buena opción para obtener requerimientos detallados en un periodo de tiempo relativamente corto, permitiendo que los interesados colaboren y obtengan un conocimiento de los requerimientos. Un gran beneficio de esta técnica son los costos, a menudo son más bajos, en comparación de la realización de múltiples entrevistas. Sin embargo, un inconveniente es la disponibilidad de las partes involucradas, ya que puede hacerse difícil la programación de los talleres. El éxito de esta técnica depende en gran medida de la experiencia y conocimiento de los



participantes. Un aspecto que se debe de considerar es el número de éstos, ya que involucrar demasiados puede ocasionar que el proceso del taller se realice lentamente, mientras que, por lo contrario, la obtención de aportaciones de pocos puede provocar que se dejen pasar requerimientos que son importantes para los usuarios. Un riesgo adicional es perderse en largas discusiones o divagar sin llegar a los requerimientos concretos (Karl Wieggers, 2013e).

2.2.11 Modelado de procesos

Se utiliza para entender cómo se realiza el trabajo que involucra múltiples funciones y departamentos dentro de una organización. El proceso describe como varias personas o grupos colaboran en un periodo de tiempo para realizar el trabajo. Este tipo de procesos son repetibles y pueden tener varios caminos para su finalización. El beneficio de esta técnica es la facilidad con la que el usuario entiende los conceptos que se plasman en el modelado. Sin embargo, éstos pueden llegar a ser extremadamente complejos y difíciles de manejar si no se estructuran cuidadosamente, haciendo imposible que un usuario lo entienda.

2.2.12 Descubrimiento de escenarios

La forma más básica para averiguar lo que la gente hace en su trabajo actual o futuro es pedirles que lo describan como si lo estuvieran haciendo, o como les gustaría hacerlo.

Existen tres contextos principales para descubrir los escenarios: entrevistas, observación y talleres. Todas estas técnicas que ayudan a identificar un escenario tienen efectos muy diferentes sobre el proyecto. La técnica que se utiliza en determinados contextos claramente difiere debido a la cantidad de gente con la que se va a tratar.

El escenario es una historia adaptada y estructurada para el uso en la ingeniería. El propósito de un escenario es comunicar una situación, por lo general a medida que evoluciona a través del tiempo, en una serie de pasos. Los escenarios comunican requisitos de forma muy efectiva, ya que explican de una manera natural las necesidades en términos humanos. Por ejemplo: un usuario tiene que realizar una tarea continuamente en un solo proceso, el analista observa o se entrevista con el usuario para identificar dentro de ese escenario la tarea repetitiva que puede ser automatizada (Ian & Ljerka, 2009).



2.3 Roles en el levantamiento de requerimiento.

Amaya (2013) sostiene que en el levantamiento de requerimientos se tienen diferentes roles, los cuales deberán estar presentes para que se realice de manera adecuada esta etapa.

2.3.1 El autor

Es el analista de negocios, responsable de recolectar los requerimientos, puede ser uno o más, el objetivo de este rol es el de escuchar a los involucrados en el sistema, para identificar sus necesidades.

2.3.2 Los Expertos

Estas personas son especialistas en determinadas soluciones, es decir, conocen muy bien un tema en particular, en específico el tema que se va a tratar en la sesión de requerimientos, deberá de realizar preguntas al cliente y analizar la información para poder determinar la viabilidad de lo que se solicita. El experto puede decir si se puede o no construir cierto módulo del sistema.

2.3.3 El Moderador

Es la persona que tiene como objetivo hacer que la sesión se lleve a cabo de forma correcta y revisando todos los puntos programados. Debe ser neutral, es decir, no debe estar del lado del cliente, ni de la empresa que ofrece la solución al problema, ya que no se debe involucrar en las discusiones que se presenten en la sesión. Puede incluso no conocer el proyecto a fondo, o los objetivos que se pretenden alcanzar. Esta persona será la responsable de mantener el foco en el tema de la reunión, es decir, que no se pierda de vista el objetivo, es quien pondrá la disciplina e invitará a los involucrados a que se integren en las conversaciones.

2.3.4 El escritor de la minuta

Esta persona es quien tendrá que generar la minuta, los documentos de los requerimientos, así como documentar los asuntos que surgieron en la sesión y que no fue posible dar una solución.



2.3.5 Participantes

Estos son todos los participantes en la reunión, pueden ser el cliente o los usuarios que van a estar involucrados en el sistema, presentarán y discutirán comentarios de lo que realizan en el negocio.

2.3.6 Autorizante

Esta persona tiene las facultades para aprobar los requerimientos que se presenten en la sesión.

2.4 Metodologías en la Gestión de los Requerimientos

Las metodologías imponen un proceso disciplinado sobre el desarrollo de software con el objetivo de hacerlo más predecible y eficiente, donde predecir no significa perder la capacidad adaptativa, no significa evitar la introducción de cambios en los requisitos, ni evitar que los nuevos surjan sino definir un camino reproducible para obtener resultados confiables. Definen, además, una representación que permite facilitar la manipulación de modelos, la comunicación e intercambio de información entre todas las partes involucradas en la construcción de un sistema (Gacitúa, 2003).

Hoy en día existen numerosas propuestas metodológicas que inciden en distintas dimensiones del proceso de desarrollo. Un ejemplo de ellas son las propuestas de metodología de desarrollo de software tradicionales, centradas específicamente en el control del proceso.

En febrero del 2001, en una reunión celebrada en Utah en los EE. UU. comienza a emplearse el término ágil para caracterizar las metodologías ágiles. En esa reunión un grupo de 17 expertos esbozaron los valores y principios que deberían permitir a los equipos desarrollar software rápidamente y respondiendo a los cambios que puedan surgir a lo largo del proyecto, se pretendía ofrecer una alternativa a los procesos de desarrollo de software tradicionales. Surge entonces la Alianza Ágil, una organización, sin ánimo de lucro, dedicada a promover los conceptos relacionados con el desarrollo ágil de software y ayudar a las organizaciones para que adopten dichos conceptos.



Las metodologías tradicionales están pensadas para el uso exhaustivo de la documentación durante todo el ciclo del proyecto mientras que, las metodologías ágiles ponen vital importancia en la capacidad de respuesta a los cambios, la confianza en las habilidades del equipo y mantener una buena relación con el cliente.

A pesar de que aportan una elevada simplificación, las metodologías ágiles no renuncian a las prácticas esenciales para asegurar la calidad del producto (Palacio, 2006).

2.5 Influencia de las metodologías en la gestión de los proyectos

Con el surgimiento una nueva forma de gestionar proyectos, así que, se hizo necesaria una distinción entre la metodología que se llevaban aplicando hasta entonces respecto de la nueva. Esta última se autodenomino ágil, por lo tanto, obligo a dar un nombre al modelo de gestión de proyectos que hasta entonces era único.

Un dato llamativo del INFORME DE CHAOS 2015, resulta de analizar los grados de resultado, para proyectos de diferentes tamaños utilizando metodologías tradicionales y ágiles. Como puede verse en la Tabla 2.0.1, cuando se aplican metodologías ágiles, los grados de éxito aumentan en diferentes tipos de proyectos (según tamaño) (InfoQ, 2015).

Tabla 2.0.1 Influencia de las metodologías en la gestión de proyectos

TAMAÑO	METODOLOGIA	EXITOSO	DISCUTIDO	FRACASO
Grande	Ágil	18%	59%	23%
	Tradicional	3%	55%	42%
Mediano	Ágil	27%	62%	11%
	Tradicional	7%	68%	25%
Pequeño	Ágil	58%	38%	4%
	Tradicional	44%	45%	11%
TOTAL	Ágil	39%	52%	9%
	Tradicional	11%	60%	29%

Nota: Fuente: (InfoQ, 2015).



Este estudio demuestra que las metodologías ágiles consiguen más del triple de porcentaje de éxitos respecto a las metodologías tradicionales, usadas en la gestión de proyectos de forma tradicional o predictiva.

2.5 ¿Qué tipo de metodología usar, Tradicional o Ágil?

García (2018) indica que no se puede afirmar categóricamente que una metodología sea mejor que otra, ya que los resultados del éxito o fracaso del proyecto dependerá no solamente se si se han elegido una metodología adecuada para su desarrollo, sino también que se hayan empleado correctamente las herramientas de gestión, realizado una buena gestión por parte del jefe de proyectos y que el equipo haya estado motivado y concentrado en el trabajo.

En la Tabla 2.0.2 este autor expone las principales diferencias que existen entre la metodología Tradicional vs la metodología Ágil.

Tabla 2.0.2 Diferencias entre metodologías Ágiles y Tradicionales

Metodologías Ágiles	Metodologías Tradicionales
Basados en heurísticas provenientes de prácticas de producción de código	Basadas en normas provenientes de estándares seguidos por el entorno de desarrollo
Especialmente preparados para cambios durante el proyecto	Cierta resistencia de los cambios
Impuestas internamente (por el equipo)	Impuestas externamente
Proceso menos controlado, con pocos principios	Proceso mucho más controlado, con numerosas políticas/normas
No existe contrato tradicional o al menos es bastante flexible	Existe un contrato prefijado
El cliente es parte del equipo de desarrollo	El cliente interactúa con el equipo de desarrollo mediante reuniones
Grupos pequeños (<10) integrantes y trabajando en el mismo sitio	Grupos grandes y posiblemente distribuidos
Pocos artefactos	Mas artefactos
Pocos Roles	Mas Roles
Menos énfasis en la arquitectura del software.	La arquitectura del software es esencial y se expresa mediante modelos.

Nota: Fuente: Tomado de <http://issi.dsic.upv.es/archives/f-1069167248521/actas.pdf>

2.6 Ventajas y Desventajas

Para dar una mejor visión al momento se seleccionan una metodología u otra al momento de gestionar un determinado proyecto, se expone en la tabla 2.3 las ventajas y desventajas que se consideran tanto para las metodologías tradicionales como las metodologías Ágiles (García, 2018).

Tabla 2.0.3. Ventajas entre metodologías Ágiles y Tradicionales

Tradicional		Ágil	
Ventajas	Desventajas	Ventajas	Desventajas
Registra toda la información.	Tras cada fase, no se puede volver hacia atrás	Cambios sobre la marcha	Producto final puede ser muy diferente de lo que inicialmente se pretendía.
El cliente sabe exactamente cómo será su producto final.	Dependencia de los requisitos iniciales. Si están mal planteados, fracasará el proyecto.	Añadir nuevas funcionalidades	Existe el riesgo de entrar en un ciclo de entrega de prototipos y nunca cerrar el proyecto.
Se conoce el tamaño, costo y fechas del	Si aparece un error de requisito o si se necesita hacer un cambio, el proyecto debe comenzar de nuevo.	Se realizan pruebas tras cada sprint	El desarrollo depende del equipo, un cambio de empleado supone necesitar tiempo para su adaptación.
Las rotaciones de empleados tienen un impacto mínimo en el proyecto, al estar todo rigurosamente documentado.	Las pruebas se retrasan hasta el final.	Producto estable tras cada entregable	Debido a las modificaciones el coste del proyecto puede dispararse.
	La planificación no tiene en cuenta las necesidades cambiantes del cliente.	Cliente integrado en el equipo	

Fuente: (García, 2018)



2.6 Conclusión

En este capítulo se ha concentrado en la revisión de las técnicas y metodologías que se pueden aplicar al momento de gestionar los requerimientos. Las técnicas apuntan a obtener respuestas por parte del cliente y los usuarios, que ayuden a comprender las necesidades del negocio y de esta manera dimensionar con mayor certeza el tiempo, la calidad y presupuesto que se necesite para llevar a cabo el proyecto.

En cuanto la metodología a seleccionar en una organización, para el desarrollo de diferentes tipos de proyectos o programas de proyectos, la selección no debe ser hecha de forma empírica o por la popularidad de determinado grupo de metodologías, pues, esa práctica puede afectar en la calidad del producto, en el tiempo de desarrollo, y en el costo del mismo. Como se expuso la metodología tradicional enfatiza en la planificación inicial y detallada para todos los proyectos, independientemente si los requisitos son conocidos o no. Se resalta la larga planificación inicial para asegurar la fijación de la variable como tiempo, costo, alcance, etc.

Sin embargo, un mundo VICA (volátil, incierto, complejo y ambiguo) las metodologías ágiles son adaptativos, permiten "convertir lo complejo en sencillo" y sacan buen partido del pensamiento lateral. No hay que confundir lo "sencillo" con lo "simple". Sencillo quiere decir mantener el nivel de complejidad manejable y productivo, eso es la agilidad. La agilidad no es prisa, es adaptación al cambio, transversalidad, esa sencillez; es invertir más tiempo y cuidado al principio, con el objetivo de prototipar e iterar cuanto antes, ofreciendo al cliente entregas más pequeñas y más productivas.

La buena noticia es que estas dos metodologías pueden convivir juntas, cada una tienen puntos a favor y en contra dentro de determinados tipos de organizaciones. Una metodología 'híbrida' podría ser una buena opción adaptando a lo que vendría siendo la realidad de una organización, ya que en los tiempos que corren, las empresas puramente rígidas o puramente flexibles no existen como tal.



Universidad de Buenos Aires
Facultad de Ciencias Económicas
Escuela de Estudios de Posgrado



Capítulo III. Buenas prácticas y Mapas Mentales



3.1 Introducción

Se ha demostrado que una gran proporción de los problemas en el desarrollo de software pueden estar relacionados con la ingeniería de requisitos (Hofmann y Lehner, 2001, pp. 58-66). Las fallas durante el proceso de Ingeniería de Requerimientos, tienen un impacto negativo significativo en el proceso de desarrollo general (Hall et al., 2002, pp. 153-160). Las fallas en los requisitos de reelaboración pueden suponer el 40% del costo total del proyecto. Si los errores de requisitos se descubren tarde en el proceso de desarrollo, por ejemplo, durante el mantenimiento, su corrección puede costar hasta 200 veces más que corregirlos durante las primeras etapas del proceso de desarrollo (Niazi y Shastry, 2003, p.402-407). Por lo tanto, los requisitos adecuados son esenciales para garantizar que se produzca el sistema que el cliente espera y que se eviten esfuerzos innecesarios.

De lo anterior se puede ver que se ha demostrado que los requisitos faltantes, deficientes y cambiantes tienen un impacto negativo en el éxito del proyecto. Por tanto, se puede argumentar que las buenas prácticas de ingeniería de requisitos que minimicen o gestionen estos problemas deberían tener una influencia positiva en el éxito del proyecto. Davis & Zowghi (2006) definen una buena práctica de requisitos “una práctica de requisitos que reduce el costo del proyecto de desarrollo o aumenta la calidad del producto resultante cuando se utiliza en situaciones específicas” (p.3).

3.2 Que es una Buena Práctica

Aunque desde distintos ámbitos y perspectivas se reconoce la utilidad e importancia de estudiar las buenas prácticas, definir no es una tarea sencilla, Abdoulaye (2003, p. 3) describe las buenas prácticas como “ejemplos exitosos de cambio en modos de hacer que mejoran un estado de cosas existente y que comportan una serie de criterios o estándares a los que se atienen”.

Es una técnica, método, proceso, actividad, incentivo o recomendación, que es más efectivo que cualquier otro conocido, para entregar cierto resultado (Armijo, 2004).

3.2.1 Mejores Prácticas

Wieggers, K. E. (2003) señala que en el desarrollo de software, una mejor práctica es un método bien definido que contribuye a una implementación exitosa del proyecto software. Las



organizaciones crean mejores prácticas para que les ayuden a resolver problemas pudiendo disminuir la tasa de fallo de sus proyectos.

a) Desarrollo de Requisitos

- Documentar el alcance y visión del proyecto: permitirá tener un mejor entendimiento de los requisitos y asegurará que todas las personas involucradas en el proyecto trabajen hacia la misma meta.
- Mantener un glosario del proyecto: facilitará una comunicación efectiva asegurando un entendimiento unánime.
- Uso de técnicas de obtención de requisitos de usuario: para facilitar esta tarea.
- Involucrar a toda la gente implicada: asegura una validación temprana del entendimiento de los requisitos.
- Desarrollo incremental de requisitos: puede minimizar la cantidad de re-trabajo del proyecto.
- Captura de requisitos usando casos de uso: será más fácil gestionar los requisitos y hacer un seguimiento de los mismos.
- Validar requisitos: para mejorar el éxito de los proyectos es crítico que se validen los requisitos de forma adecuada.
- Verificar requisitos: para asegurar que los requisitos proporcionan una base adecuada para llevar a cabo el diseño, la construcción y las pruebas.

b) Gestión De Requisitos

A continuación, se muestran una serie de mejores prácticas relacionadas con la gestión de requisitos:

- Priorizar requisitos: para determinar aquellos que se deberían cumplir en la primera versión o producto y aquellos que pueden llevarse a cabo en sucesivas versiones.
- Establecer líneas base de los requisitos: para asegurar que cualquier modificación en los requisitos que cambie la línea base se trata como cambios de alcance.
- Comunicación abierta: para asegurar que la información relacionada con los requisitos se comunica de forma consistente. Una comunicación abierta también implica comunicar a la gente correcta y al conjunto mínimo de personas.



- Gestión de cambios de los requisitos: es esencial gestionar estos cambios de forma efectiva y eficiente.
- Uso de herramientas para la gestión de requisitos: para facilitar la gestión de requisitos.
- Mantener trazabilidad de requisitos: para llevar un seguimiento de la vida de un requisito.
- Establecer un plan de mejora de procesos para la ingeniería de requisitos: para cumplir con las necesidades actuales y futuras de forma más eficiente y con mayor calidad.
- Formar a los analistas de requisitos: para asegurar que los analistas de requisitos tienen el conocimiento, entre otros aspectos, de cómo escribir buenos requisitos, etc.

3.2.2 Atributos esenciales para determinar Mejores Practicas

Parker (2015) menciona que, en una única organización, un equipo puede necesitar el perfeccionamiento de requisitos con revisiones detalladas por participante, a diferencia de otro, que puede empezar con una idea simple y repetir y crear prototipos hasta contar con los conocimientos suficientes para empezar a generar la solución.

a) Facilidad de uso

La implementación de la solución se debe configurar fácilmente para adoptar el lenguaje utilizado en la organización. No debe ser un requisito que el léxico cambie para adaptarse a la solución. La formación, implementación razonable y localización se deben terminar en un plazo de dos a cuatro semanas.

La solución debe ser fácil de aprender Los nuevos usuarios, independientemente de su función, pueden aprender rápidamente a iniciar sesión en la solución y navegar por la interfaz del cliente. En este contexto, los revisores de requisitos no técnicos, que comprenden los requisitos de la empresa, pueden empezar a trabajar con una descripción general de la página. El resto de los usuarios de la solución, a excepción de los administradores o gestores de proyectos formados para admitir la implementación, deberían poder utilizar el producto después de dos a cuatro horas de formación.



La solución debe ser fácil de usar La solución seleccionada para la gestión y generación del informe de requisitos debe facilitar la tarea de todos los participantes y poder aumentar, al mismo tiempo, la posibilidad de los mismos de acceder a la información. El usuario general no debería tardar más de una hora en entender cómo realizar funciones específicas con la solución.

Las propiedades y visualización deben ser configurables Debe ser posible para los usuarios o administradores modificar la cantidad de información visualizada.

Todos los participantes deben tener acceso a documentos Debe ser posible publicar documentos en formato empresarial para su revisión y aprobación. No debería ser necesario que los usuarios utilicen la herramienta para ver la información que incluyen.

Debe haber métodos para organizar los requisitos

Debe ser posible crear contenedores (carpetas) como método para organizar requisitos según el componente, subtipo o para crear listas de usuarios o grupos específicos.

Se deben admitir las funciones de los usuarios Debe ser posible definir funciones organizativas o de proyecto y asignar usuarios a dichas funciones.

b) El sistema debe ser flexible

Los requisitos se deben almacenar como objetos individuales Los requisitos de todos los tipos se deben almacenar como objetos individuales. Se pueden enlazar a otros objetos del proyecto, pero no ser propiedad de estos. Se puede recopilar cualquier combinación de estos objetos para su visualización o publicación.

Los requisitos se deben poder organizar en jerarquías y descomponer hasta el nivel atómico Requisitos de todos los tipos se deben poder dividir en requisitos cada vez menores hasta que alcancen el tamaño atómico que representa únicamente un requisito específico. Las jerarquías de requisitos descompuestos se deben poder manejar individualmente, en nivel atómico, y en la jerarquía y el nivel de jerarquía intermedio con acción en cascada hasta los requisitos descompuestos.

Los tipos de requisitos deben reflejar los estándares empresariales Se debe asignar un nombre y describir los tipos de requisitos según las necesidades y los procesos de la organización, en lugar de los del proveedor de la herramienta. No debe haber ningún requisito en la solución que incluya un tipo de requisito de uso inmediato.



c) La propiedad debe ser configurable

Se deben poder definir y asignar propiedades a tipos de requisitos, más allá de los valores predeterminados definidos por el producto. Los tipos de datos usados en la definición de estas propiedades también deben ser flexibles. Por ejemplo, deben incluir campos de texto (cortos y largos), listas desplegables, campos numéricos y campos de fecha. Debe ser posible asociar gráficos o adjuntos a los tipos de datos.

Las propiedades tienen que utilizar términos conocidos. El lenguaje utilizado para definir los requisitos y sus propiedades debe reflejar el léxico de la organización y se debe hacer referencia a él de forma coherente en la solución. Debe ser posible que varios equipos de proyecto de la misma organización definan tipos de requisitos y utilicen términos que cumplan las necesidades de cada uno. No se debe requerir que ningún grupo lleve el bagaje de los otros.

Debe haber un seguimiento, debe ser posible crear una vista de seguimiento, con mecanismos para seleccionar los tipos de requisitos que se van a seguir. La información recopilada se debe visualizar en un formato que sea fácil de leer y de analizar.

3.2.3 Guía de escritura de buenas practicas

A continuación, se presenta una guía para la escritura de los requisitos, con un enfoque en las metodologías ágiles (Ambler, 2009).

- ✓ Participación activa de los stakeholders: Los stakeholders deben proveer información en tiempo y forma para poder tomar decisiones y deben participar activamente a lo largo del proceso de desarrollo.
- ✓ Visión arquitectónica: Al iniciar un proyecto ágil se deberá llevar a cabo modelado arquitectónico de alto nivel para poder identificar la viabilidad técnica de la solución.
- ✓ Especificaciones Ejecutables: Preferentemente priorizar las especificaciones ejecutables a la documentación estática. Especificar los requerimientos en la forma de casos de pruebas ejecutables y el diseño como pruebas unitarias.
- ✓ Modelado iterativo: al comenzar cada iteración habrá cierta carga de modelado como parte de la planificación de la iteración.
- ✓ Bueno suficiente: Los modelos y documentación generada deben ser buenos y suficientes.



- ✓ Requerimientos priorizados: los equipos de desarrollo implementan las historias de usuario según su prioridad. Estas prioridades son especificadas por los stakeholders.
- ✓ Visión de Requerimientos: Al iniciar un proyecto nuevo se deberá invertir tiempo en identificar el alcance del proyecto y en crear una lista inicial de requerimientos priorizados.
- ✓ En proyectos grandes y complejos resulta difícil saber por dónde comenzar a identificar las historias de usuario. Una alternativa es comenzar por identificar los roles y los objetivos que persiguen estos roles con el software.
- ✓ Las historias de usuario pueden descomponerse en partes más pequeñas.
- ✓ Escribir historias de usuario cerradas: una historia cerrada es aquella que finaliza con el cumplimiento de un objetivo y que permite a los usuarios sentir que han alcanzado algo.
- ✓ Evitar incluir características de la interfaz de usuario en la descripción de la historia de usuario.
- ✓ Hacer foco en las áreas más importante: prestar mayor atención a las cosas que estarán sucediendo en el futuro cercano.
- ✓ Utilizar otras formas de especificación de requerimientos si fuera necesario.
- ✓ Escribir en voz activa.
- ✓ Dejar que los clientes sean quienes escriban las especificaciones en la medida de lo posible.
- ✓ No enumerar las historias de usuario.

3.2.4 Requisito como proceso de buenas practicas

Kevin Parker (2015) en su informe de mejoras prácticas de ingeniería de requisitos hace referencia al proceso debería seguir, a continuación, mencionare brevemente los pasos citados ya que algunos de ellos ya se encuentran mencionados en capítulos anteriores.

Paso 1: Ideación y captura de demanda: Un elemento esencial de este proceso es la recopilación de ideas de una forma coherente y rastreable. La ideación es un concepto más moderno y no formalizado en muchas organizaciones. En algunas empresas, lo realiza el comité de revisión de control de cambios (CCRB) o el comité de asesoramiento en materia de cambios (CAB) y muchos otros títulos similares.



Paso 2: Gestión de cambios: Aquí es donde se produce la clasificación de las ideas de modo que las más importantes, las que tienen restricciones temporales y las que son esenciales para el control tienen la máxima prioridad. Se realiza un estudio de la viabilidad, financiación, obtención de recursos y proceso de programación de las ideas antes de aprobarlas para su desarrollo. Algunas peticiones urgentes y de emergencia obtienen la aprobación inmediata si el impacto de las mismas en la empresa es grande.

Paso 3: Obtención de requisitos: Implica la recopilación y registro de los requisitos de los participantes y otras fuentes. Se utilizan varias técnicas, entre otras, entrevistas, análisis de documentos, grupos específicos, talleres y, más recientemente, creación de prototipos, simulación y ludificación. La creación iterativa de prototipos define los requisitos en menos tiempo de lo que se tardaría en preguntar y volver a preguntar al usuario empresarial.

Paso 4: Validación de requisitos: Confirma que el conjunto de requisitos cumple las necesidades y objetivos empresariales. Este paso existe para garantizar que la información recopilada es suficiente para crear la solución deseada, y que cumple con restricciones organizativas tales como estándares de conformidad y control.

Paso 5: Gestión de requisitos: A menudo, esta actividad sigue procedimientos muy formales y es el proceso en el que los requisitos se mejoran, versionan, rastrean, supervisan, priorizan, asignan y, en definitiva, gestionan.

El proceso de gestión de requisitos es esencial para las mejores prácticas de desarrollo de aplicaciones. Sin el conocimiento de lo que se supone que se va a desarrollar, la probabilidad de que coincida con la necesidad empresarial es casi nula. No importa cómo se defina el término, o dónde se encuentre su organización en el espectro metodológico, se deben entender las expectativas del cliente (o del cliente potencial) antes de comenzar la construcción. El “cómo” se puede trabajar en desarrollo, pero el “qué” debe entenderse antes de la aprobación de la financiación.

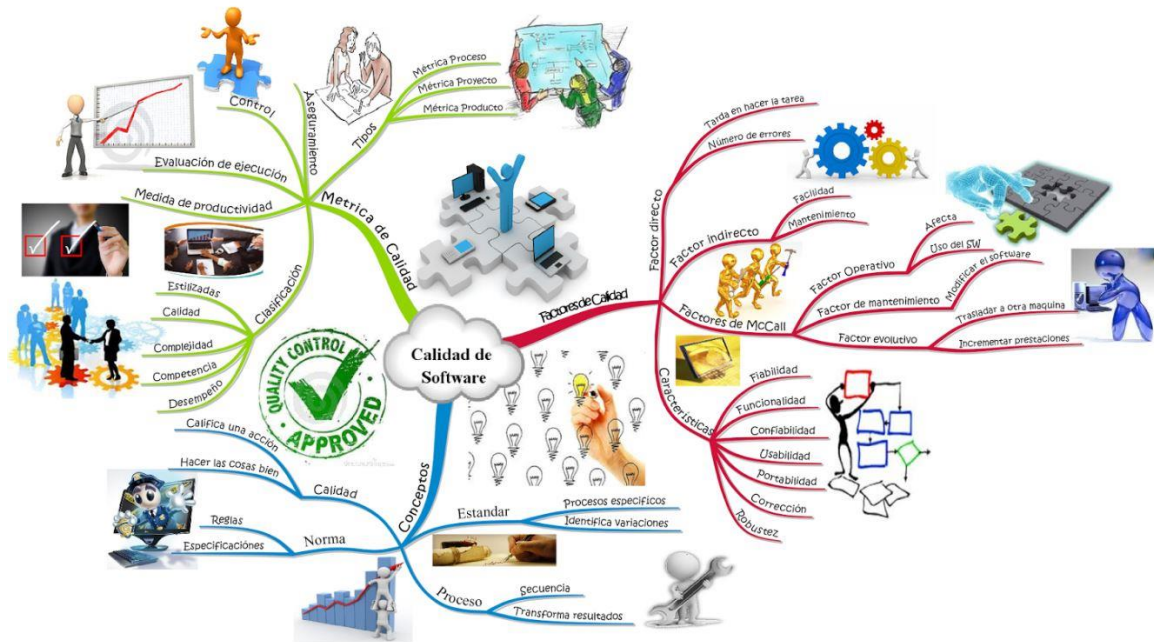
3.3 Mapas Mentales en la Ingeniería de requerimientos

El mapa mental es una técnica que utiliza la ilustración gráfica para expresar pensamientos e ideas basadas en el concepto de Pensamiento Radiante; una función natural de la mente humana. El término pensamiento radiante fue introducido por primera vez por Tony Buzan para reflejar la forma en que nuestro cerebro estructura y procesa la información.

Cada bit de información que ingresa a nuestro cerebro puede considerarse como un punto central que luego puede irradiar hasta millones de direcciones. Y cada dirección tiene su propia variedad infinita de vínculos y conexiones (Buzan, et al., 1995).

El mapa mental es capaz de soportar el potencial uso equilibrado del cerebro derecho e izquierdo (Morris, et al., 1998). Como lo fundó en la década de 1960 el Dr. Roger W. Sperry, cada parte del hemisferio cerebral tiene su propia forma de pensar, aprender y expresar ideas (Sperry, 1981). El lado izquierdo de los hemisferios es más dominante en palabras y procesos en un orden más lógico y secuencial. Por el contrario, el lado derecho es más visual y opera de manera intuitiva, holística y aleatoria. La mayoría de las personas tienden a dominar un lado del cerebro, pero lo ideal es mantenerlo en equilibrio para que se pueda utilizar todo el poder o potencial del cerebro. Un mapa mental se puede representar como se muestra en la siguiente Figura 3.1.

Figura 3.0.1 Mapa Mental de Ingeniería de Requerimientos



Fuente: Jose, B. (2017). Psicología con Alfonso. Recuperado de <http://gabrielalmeida.com.mx/617/entendiendo-el-babok-en-espanol-capitulo-1-introduccion-1-3-3-requerimientos>



3.3.1 Ley de mapas mentales y recomendación de mapas mentales

Siguiendo el concepto, existen reglas para crear el Mapa Mental para que uno pueda maximizar sus beneficios. Se clasifican en leyes de técnica y leyes de distribución (Buzan, et al., 1995).

Leyes de la técnica

- Use énfasis: use imágenes centrales, incorpore imágenes cuando sea apropiado a través del mapa mental, use diferentes colores, use variaciones de tamaño para las palabras clave, imágenes y ramas y use el espaciado apropiado.

- Use asociaciones: use flechas para la conexión, use códigos de colores para categorizar y agrupar.

- Sea claro: facilite la captura de la imagen general, las asociaciones, las palabras clave y la categorización y clasificación.

- Desarrolle un estilo personal.

Leyes de distribución

- Use jerarquía y categorización.

- Utilice órdenes numéricos (cuando corresponda) para indicar la priorización o los órdenes cronológicos.

3.3.2 Mapa mental grupal

Es una forma para que grupos de individuos combinen, colaboren y multipliquen sus capacidades creativas personales (Buzan, et al., 1995). Se sabe y se acepta que trabajar en grupo es mejor que trabajar individualmente. Se aplica lo mismo al mapa mental grupal. Las capacidades y las funciones del mapa mental individual se multiplican en el mapa mental grupal. Por ejemplo, cuando se usa el mapa mental grupal en una sesión de lluvia de ideas o en la captura de ideas, la capacidad individual aumenta con la retroalimentación instantánea de los miembros del grupo. Se puede generar más información, puntos de vista, ideas creativas y opiniones desde el mapa mental grupal.

Aparte de eso, el mapa mental grupal fomenta el intercambio de información y el trabajo en equipo (Morris, et al., 1998). Todos los miembros del equipo tendrán la sensación de haber



realizado aportes que los animarán a comprometerse con la ejecución del trabajo (Hughes, 2008).

Las principales aplicaciones del mapa mental grupal son la creatividad conjunta, el recuerdo combinado, la resolución y el análisis de problemas grupales, la toma de decisiones grupales, la gestión de proyectos grupales y la capacitación y educación grupales (Buzan, et al., 1995).

Etapas Principales de un Mapa Grupal

Según Tony Buzan en su libro "The Mind Map Book", hay siete etapas principales en el proceso del mapa mental grupal.

1) Definir el tema: el tema y el objetivo deben estar claramente definidos y comprendidos por todos los miembros del equipo. Además, toda la información relevante a ser considerada en la discusión debe ser transmitida a todos los miembros del grupo.

2) Lluvia de ideas individual: cada miembro debe tener al menos una hora para construir su propio mapa mental individual.

3) Discusión en grupos pequeños: luego, los grupos se dividirán en grupos pequeños para intercambiar y compartir sus ideas. Los miembros deben mantener una actitud totalmente positiva y de aceptación.

4) Todas las ideas y comience a construir el mapa mental grupal.

5) Incubación: un proceso en el que la búsqueda de ideas tiende a una actividad analítica y verbal ininterrumpida hasta que se logra un resultado.

6) Segunda reconstrucción y revisión: repetir los pasos 2, 3 y capturar los resultados de los pensamientos recién considerados e integrados. El resultado de esta actividad es otro mapa mental grupal que se puede utilizar para comparar como preparación para la etapa final.

7) Análisis y toma de decisiones: toma decisiones, establece objetivos, planifica los dispositivos y edita en consecuencia.



3.3.3 MP Colaborativo para Facilitar el Proceso de ingeniería de Requerimientos

Muchos investigadores se han dedicado a encontrar formas de mejorar los procesos de IR⁷ principalmente debido al conocimiento de la IR como uno de los factores cruciales que contribuye al éxito del proyecto de software. A partir del análisis de la literatura, hemos comprendido que el proceso de IR se trata más de personas, comunicación y colaboración. El proceso de IR no puede abordarse únicamente desde el punto de vista técnico, sino también teniendo en cuenta su contexto social para el que creemos que es más crucial. (Eric T. Blue, 2006). Hay bastantes herramientas de mapas mentales que se utilizan en la gestión de proyectos, pero muy pocas en el proceso de IR. Un ejemplo obvio que muestra un uso práctico muy convincente del mapa mental en la obtención de requisitos es el de Kenji Hiranabe (2006). En su artículo, describió algunos beneficios e ilustró los ejemplos con plantillas específicas en el uso del mapa mental en el proceso de IR. También proporciona una forma de mapear mapas mentales en diagramas UML⁸.

a) Facilitar la comunicación

La herramienta de mapa mental colaborativo puede servir como un mecanismo para capturar los requisitos de una manera más estructurada, rápida, clara y persuasiva. La comunicación entre las partes interesadas será más eficiente. Permitirá la comunicación (donde no sería posible de otra manera) y traerá múltiples perspectivas o información en un solo lugar. La información recopilada se almacenará de manera centralizada para garantizar la coherencia de los datos. Además de los requisitos reales, también puede ayudar a capturar los atributos del requisito (prioridad, fuente, persona responsable, etc.) que ayudarán a comprender mejor el requisito (Hiranabe, 2006).

⁷ **UML** siglas en inglés, *Unified Modeling Language*. es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad; está respaldado por el *Object Management Group* (OMG).

⁸ **IR** El proceso de recopilar, analizar y verificar las necesidades del cliente para un sistema es llamado “Ingeniería de Requerimientos”.



b) Promover el trabajo en grupo

Permitir que un grupo de personas trabaje de manera bien organizada, estructurada, distribuida y coordinada. Además, se puede aprovechar el espíritu de equipo y la confianza entre los miembros del equipo (Hiranabe, 2006).

c) Ahorre tiempo y dinero

El aumento de la globalización ha exigido organizaciones de desarrollo de software de múltiples sitios (Damian, et al., 2003). La demanda puede ser satisfecha por herramienta de mapa mental colaborativo que puede ayudar a ahorrar tiempo y costos. No será necesario que el analista vuele sobre el continente solo para capturar los requisitos.

d) Solicitud de requisitos

Al recopilar los requisitos, puede ayudar en la "lluvia de ideas estructurada". A diferencia de la lluvia de ideas típica, el mapa mental colaborativo fomenta la generación de ideas creativas y el pensamiento "fuera de la caja" formado de manera estructurada a través de una forma natural de pensar. Además del método de lluvia de ideas, también puede ilustrar el proceso o los procedimientos del sistema futuro y categoriza las listas de deseos de los clientes (Hiranabe, 2006).

e) Modelado y análisis de requisitos y validación de requisitos

En el modelado y análisis de requisitos, el mapa mental colaborativo puede ayudar a categorizar los requisitos (por ejemplo, requisitos funcionales frente a requisitos no funcionales), refinar, revisar, priorizar y validar los requisitos. Y, si surgen problemas en esta fase, se puede utilizar la herramienta de mapa mental colaborativo para facilitar la toma de decisiones (Hiranabe, 2006).

3.4 Cultura organizacional

En 1980, el profesor de administración estadounidense Edgar Schein desarrolló un modelo de cultura organizacional para hacer la cultura más visible dentro de una organización. También indicó qué pasos deben seguirse para lograr un cambio cultural. Según Edgar Schein,



existen mecanismos directos e indirectos dentro de las organizaciones. El modelo de cultura organizacional de Schein está directamente influenciado por mecanismos directos. Esto incluye comportamiento, opiniones, estatus y nombramientos ejemplares. Los mecanismos indirectos no influyen directamente en la cultura organizacional, pero son determinantes. Esto incluye la declaración de misión y la visión de una empresa, pautas formales, identidad corporativa, rituales y diseño.

3.4.1 Niveles de cultura organizacional

Edgar Schein dividió la cultura organizacional en tres niveles diferentes:

1. Artefactos y símbolos
2. Valores propuestos
3. Supuestos subyacentes básicos

En la práctica, los tres niveles del modelo de cultura organizacional de Schein a veces se representan como un modelo de cebolla, ya que se basa en diferentes capas. La capa exterior es bastante fácil de adaptar y cambiar. Cuanto más profunda sea la capa, más difícil será ajustarla. Profundamente incrustados en el corazón de la cebolla encontramos los supuestos. Alrededor del núcleo encontramos los valores. Los artefactos y símbolos se pueden encontrar en las capas externas de la cebolla y se pueden cambiar más fácilmente. Entre esta capa y la capa en la que están incrustados los valores, puede haber otra capa en la que nos encontremos con los llamados “héroes”; personas que juegan o han jugado un papel importante en la organización y que son admiradas. El núcleo de la cebolla se compone de suposiciones. Se trata de "cómo funciona el mundo". De acuerdo con todas las personas que pertenecen a la organización y provienen de experiencias y percepciones. Estos se han convertido en parte en suposiciones inconscientes y se consideran evidentes por sí mismos, por lo que no es necesario discutirlos.

3.4.2 Modelo y cambio cultural

El modelo de cultura organizacional de Schein proporciona puntos de referencia para crear un cambio cultural. Según Edgar Schein, es sensato tener conversaciones con tantos empleados como sea posible para descubrir los antecedentes y aspectos subyacentes de la cultura organizacional. Estos podrían ser una base para el cambio cultural. Las personas deben ser conscientes de que el cambio cultural es un proceso de transformación.; el comportamiento



debe desaprenderse primero antes de que se pueda aprender un nuevo comportamiento en su lugar. Cuando surge una diferencia entre la cultura deseada y la predominante, se deben realizar intervenciones culturales. La responsabilidad recae en la alta dirección respaldada por un departamento de personal. Esto requiere un enfoque integral. Un nuevo logotipo, un estilo corporativo o una formación de orientación al cliente no serán suficientes. Es importante que se midan los resultados y que se recompense el buen desempeño.

3.5 Conclusión

Para algunas organizaciones es difícil dar forma a la disciplina de gestión de requisitos, y por ello, los grupos suelen sobrecargar el proceso en detrimento del proyecto. Los equipos de proyecto suelen pasar más tiempo discutiendo acerca de la adecuada estructura del caso de uso o la diferencia entre un usuario y un requisito empresarial, que acerca del contenido de cualquiera de ellos. Puesto que dichos argumentos no hacen avanzar el proyecto, los miembros del equipo continuarán y asumirán que las preguntas abiertas se abordarán durante el desarrollo. Por lo que es esencial la implementación de buenas prácticas en la gestión de requisitos, estas prácticas son empleadas con el fin de que las actividades realizadas por cada miembro de un equipo de trabajo, y esté tengan concordancia con las actividades realizadas por los demás miembros. Esto se realiza mediante el seguimiento de los preceptos correspondientes, que, al ser implementadas, generan un proyecto comprensible por quienes conforman el equipo de trabajo. El no seguimiento de los preceptos puede generar retrasos e incluso fracaso en un proyecto.

Las mejores soluciones de gestión de requisitos son fáciles de aprender, fáciles de usar, accesibles y transparentes. Además, cuentan con excelentes funcionalidades de seguimiento y creación de versiones e informes. Estas funcionalidades trabajan conjuntamente para proporcionar la base de una solución de prestigio y el inicio de un proceso de requisitos de mejores prácticas.

Tiene mucha razón al decir que: “así como las competencias distintivas de una empresa pueden ser difíciles de imitar por otras, sus mejores prácticas pueden ser difíciles de replicar internamente”. El proceso de propagar las mejores prácticas a través de una organización está lleno de desafíos, pero es un esfuerzo que bien vale la pena realizar.



El uso de los mapas mentales permite lograr el objetivo propuesto y resultó ser una herramienta estratégica eficaz para explicar la estructura de un problema e identificar los puntos de vista y los requisitos críticos en el proceso. Se puede utilizar para permitir, a través de conceptos mentales, individuales y colectivos, obtenidos a partir de declaraciones escritas y entrevistas, la identificación y estructuración de los problemas existentes de los requisitos para el desarrollo de productos. Los resultados obtenidos pueden tomarse como base para posibles acciones correctivas. Esta investigación es relevante por presentar un aporte a la ingeniería de requerimientos, mostrando la eficiencia de los mapas mentales para identificar los puntos críticos de los requisitos durante el proceso de desarrollo del producto, ya que, en general, un requisito bien diseñado reduce el costo de desarrollo, evitando retrabajos y asegurando la satisfacción de los grupos de interés.



Universidad de Buenos Aires
Facultad de Ciencias Económicas
Escuela de Estudios de Posgrado



Capítulo IV. El Rasgo de Fuerte Presencia de Pensamiento Grupal



4.1 Introducción

Se cree ampliamente que el éxito o el fracaso de un proyecto de software depende en gran medida del personal que trabaja en él. El personal calificado que trabaja en conjunto puede producir software confiable dentro del presupuesto. Pero la falta de la combinación adecuada de personas puede producir un fracaso. Este factor en la creación de software, generalmente denominado *Peopleware*⁹, justifica la investigación sobre cómo se comportan las personas en los grupos de ingeniería de software (Brown, 2014).

Para resolver la falla de un equipo puede ser un problema de tipo complicado. Requerirá tal vez el análisis de su diseño industrial, software y hardware, posiblemente implique el concurso de especialistas, ajenos a la empresa y reunir información sistematizada e insumir un largo tiempo de análisis. Pero la solución existe, es conocida, debe ser localizada y descubierta, aunque sea con esfuerzo (Terlato, 2011, p.8).

Argumentamos que la ingeniería de requisitos, como base para el despliegue de tecnología de la información en las organizaciones, está fallando en gran medida porque la toma de decisiones en el proceso de ingeniería de requisitos (RE) empatiza con el espíritu del “pensamiento grupal” que “alimenta la indeterminación de control” ineficiente. A esto lo podemos denominar GroupThink (Janis, 1982; Janis, 1972), es autor menciona que es un fenómeno en el que la cohesión del grupo puede afectar negativamente la capacidad del grupo para tomar buenas decisiones. Algunos autores han declarado que los grupos de ingeniería de software sufren de GroupThink (Kontio, et al, 2004; Salinger, et al., 2008). Pero estos artículos no entran en detalles sobre las técnicas de diagnóstico de GroupThink descritas por Janus (Janus, 1982).

La mayor parte de la investigación en GroupThink se realiza en forma de estudios de casos (Janus, 1982). Estos estudios analizan grupos específicos que tomaron malas decisiones y, a través del análisis, determinan que padecen GroupThink. Algunos ejemplos comunes son la decisión de la administración Kennedy de apoyar la invasión de Bay of Big¹⁰; La decisión

⁹ **Peopleware** es un término utilizado para designar uno de los tres aspectos centrales de la tecnología de computadores, siendo los otros dos: hardware, software

¹⁰ **Bay of Big** también conocida como invasión de playa Girón(Cuba), para formar un gobierno provisional que reemplace al de Fidel Castro y buscar el apoyo de la Organización de los Estados Americanos y el reconocimiento de la comunidad internacional. La acción acabó en fracaso en menos de 65 horas.



de la NASA de lanzar el transbordador espacial Challenger y la decisión de la administración Nixon de encubrir Watergate. En lugar de centrarse en un grupo específico y un solo proceso de toma de decisiones, ese grupo son los equipos de desarrollo de software. Después de décadas de investigación en Peopleware, hemos aprendido mucho sobre cómo funcionan estos equipos.

4.2 Cohesión Grupal

GroupThink¹¹ fue introducido por primera vez por Irving Janis en 1972 y es un fracaso que muestran algunos grupos que les impide tomar buenas decisiones. Muchos factores pueden llevar a GroupThink, sin embargo, es principalmente un problema con la cohesión del grupo.

La cohesión grupal es un fenómeno que se ha observado en grupos desde el inicio del estudio de la dinámica de grupo. Aunque hay algunas opiniones diferentes, comúnmente se piensa en la cohesión grupal como "el grado en que los miembros de un grupo desean permanecer en el grupo". (Cartwright y Zander, 1968) La cohesión grupal alta puede producir muchos beneficios, entre ellos: capacidad para retener miembros; la participación y lealtad de los miembros; la propiedad del producto y el sentimiento de seguridad de los miembros (DeMarco y Lister, 1987).

La cohesión del grupo debe ser un objetivo de todo gerente de proyecto de desarrollo de software. La cohesión tiene muchos efectos positivos en el desarrollo de software y miembros prominentes de la comunidad del software la promueven como una forma de incrementar el desarrollo (Demarco y Lister, 1987). La cohesión parece, lógicamente, una característica de grupo positiva. La cohesión, en cualquier grupo, ayuda a reducir la ansiedad, aumenta la autoestima y retiene a los miembros (Cartwright y Zander, 1968). Dado que el desarrollo de software actual es una ocupación de alta ansiedad y de alta rotación, se cree que la cohesión del grupo es una solución a muchos problemas.

4.3 GroupThink

El pensamiento grupal es un término que se usa para describir el "modo de pensar en el que se involucran las personas cuando están profundamente involucradas en un grupo interno

¹¹ **GroupThink** Pensamiento de grupo es una expresión acuñada por el psicólogo Irving Janis en 1972 para describir el proceso por el cual un grupo puede tomar decisiones malas o irracionales.

cohesionado, cuando el esfuerzo de los miembros por la unanimidad anula su motivación para evaluar de manera realista cursos de acción alternativos". (Janus, 1972) Los miembros del grupo se concentran en mantener la cohesión del grupo y evitar conflictos. Este fenómeno ocurre todo el tiempo y generalmente pasa desapercibido. Los miembros generalmente desconocen el pensamiento grupal cuando ocurre. La Tabla 4.1 muestra las condiciones primarias que generan GroupThink (Janus, 1972). Es fácil ver que muchos grupos de ingeniería de software presentan estas condiciones.

Tabla 0.1 Condición de GroupThink

Alta cohesión grupal
El grupo tiene aislamiento del juicio de elementos fuera del grupo, mientras que el proceso de grupo está en marcha
Líder del grupo promueve sus propias ideas

Fuente: (Brown, 2014)

Después de examinar las condiciones que pueden causar GroupThink, Janus presenta las características de GroupThink. Al analizar un grupo utilizando estas características, se puede determinar si el grupo sufre de GroupThink. La Tabla 4.2 muestra estas tres características.

Tabla 0.2. Características de GroupThink
Características de GroupThink

Características de GroupThink
Sobreestimaciones del grupo
Mentalidad cerrada
Presión hacia la uniformidad

Fuente: (Brown, 2014)

GroupThink es una condición que ocurre en muchos grupos. Es causado fundamentalmente por la cohesión del grupo, pero dado que hay muchos beneficios en la cohesión, los métodos de cohesión pueden prevenir GroupThink sin eliminar la cohesión. Se han realizado muchas investigaciones sobre el tema y los métodos que lo previenen. Más adelante abordaremos estos métodos.



4.4 Priorización organizativa como "pensamiento de grupo"

Terlato (2014) hace referencia en su tesis doctoral, lo citado por Bernard en su libro en el año 1938, que los ejecutivos tienen la obligación de inculcar a los trabajadores un compromiso ético que trascienda su propio interés. Reconoce que una organización comprende la cooperación de individuos que trabajan reunidos en forma de grupos.

En interpretaciones tradicionales, investigadores como Janis (1972) consideran el pensamiento grupal como una tendencia de búsqueda de concurrencia que puede obstaculizar los procesos de toma de decisiones colectivas y conducir a malas decisiones que, a su vez, inducen fracasos. Esta conceptualización tradicional del pensamiento grupal está vinculada a la toma de decisiones grupales que producen resultados desfavorables vinculados a una serie de factores antecedentes como el liderazgo (Leana, 1985), la cohesión del grupo (Callaway y Esser, 1984), la amenaza externa (Turner, Pratkanis, Probasco & Leve, 1992), y así sucesivamente. Desde una perspectiva práctica, la conceptualización tradicional del pensamiento grupal se ha aplicado en organizaciones en varios dominios gerenciales, incluida la toma de decisiones (Miranda, 1994), el liderazgo (Hughes, Ginnett y Curphy, 1993) y la gestión de equipos organizacionales (Kayser, 1994). En estos dominios, el pensamiento de grupo se ha considerado un proceso grupal perjudicial (Miranda, 1994). La consecuencia de considerar el pensamiento grupal como perjudicial para la toma de decisiones es una búsqueda para desarrollar estrategias organizacionales para evitar el pensamiento grupal en las organizaciones (Quinn, Faerman, Thompson y McGrath, 1990).

Groupthink es un anatema para la toma de decisiones organizacionales. Esta afirmación encuentra fuerza en las inferencias del hecho de que la evidencia empírica que demuestra que el pensamiento grupal tiene implicaciones negativas extrapoladas del análisis de casos políticos (Janis, 1972; Moorhead, Ference y Neck, 1991) o de entornos de laboratorio que involucran a estudiantes universitarios (Driskell y Salas, 1991; Leana, 1985). Por lo tanto, en esta sección, usamos el pensamiento grupal como una metáfora estructurante, no solo para demostrar que el pensamiento grupal, tal como se percibe, no brinda apoyo en los entornos de toma de decisiones organizacionales (como las decisiones de priorización de TI), sino también para mostrar que el



pensamiento grupal tal como se practica en entornos organizativos es actualmente defectuoso, por lo tanto, fallas repetidas en las tareas de ER¹².

Wiegers vincula la idea de un requisito con la de una propiedad de una solución (TI) que proporciona valor a un interesado en particular (Ruhe G, Eberlein A, Pfahl D., 2002). Inevitablemente, cuando se mencionan partes interesadas, el conflicto que surge de la supuesta multiplicidad de intereses, preferencias y limitaciones técnicas de las partes interesadas implican que los requisitos en el diseño se negocian dentro de un entorno basado en valores y deben resultar en la maximización del valor comercial (Lehtola et al, 2009). La implicación es que los requisitos en diseño que surgen a través de la negociación dentro de un contenido basado en valores no pueden ser todos igualmente importantes (Boehm, 2003), por lo tanto, la necesidad inherente de priorización y selección de requisitos (Sawyer, 2000). Este análisis permite dos inferencias: la primera es que el proceso de llegar a requisitos "negociados" implica que la noción de "pensamiento grupal" es parte integral del proceso de toma de decisiones organizacionales. Negociar implica no solo un proceso que involucra a grupos de personas; pero también fomenta la visión orientada al proyecto de que la negociación surge de la necesidad de minimizar el conflicto entre las partes interesadas (Muganda, Asmelash y Samali, 2012).

La segunda inferencia se deriva de la primera: que debido a que el proceso de ER ya tiene fallas, cualquier enfoque de modelado adoptado en la priorización y selección de requisitos (ya sea cualitativo o cuantitativo) resultará en alguna falla. Por lo tanto, para que el pensamiento de grupo apoye las decisiones relativas a la priorización y selección de requisitos, argumentan Muganda, Asmelash y Samali que la base debe ser que los diseñadores tengan una evaluación crítica de aquellos procesos en la organización que se consideran modulares y / o integrales y dependientes. sobre si la organización tiene la capacidad y / o los conocimientos necesarios para implementar la solución de TI. Que, sobre la base de esta categorización, se toma una decisión de priorización y selección de ER. Simchi-Levi, Kaminsky y Simchi-Levi (2008) caracteriza un producto modular como uno basado en la preferencia del cliente que

¹² **ER** Especificación de requerimientos es un documento que describe las características que debe cumplir.



comprende la combinación de varios componentes independientes e intercambiables con interfaces estándar. Por lo tanto, un proceso modular serían aquellos procesos que dan como resultado la realización de productos modulares y comprenderían una combinación de procesos independientes pero intercambiables. Por otro lado, un proceso integral es aquel en el que las funcionalidades del proceso están estrechamente relacionadas y se diseñan desde una perspectiva de arriba hacia abajo (Simchi-Levi, Kaminsky & Simchi-Levi, 2008). Por lo tanto, el desafío para la organización de TI¹³ es determinar si los requisitos que se obtienen son procesos de focalización que son integrales o modulares (Muganda, Asmelash y Samali, 2012).

4.5 Pensamiento grupal en la ingeniería de software

Varias publicaciones hacen referencia a GroupThink cuando se habla de dinámicas culturales o de grupo en ingeniería de software y tecnología de la información (Salinger, et al., 2008; Gallivan & Srite, 2005). Dado que cualquier grupo tiene el potencial de exhibir GroupThink, es fácil concluir que el grupo de desarrollo de software podría ser víctima de él. Pero estas publicaciones no intentan generalizar el típico grupo de ingeniería de software y concluyen que GroupThink se exhibe en ellas. Una excepción a esto es un artículo publicado por Schiano y Weiss (Schiano & Weiss, 2006). Estos autores afirman que GroupThink fue la causa de la crisis del año 2000 y actualmente es la razón por la que muchas organizaciones tienen una seguridad deficiente en tecnología de la información.

La investigación de Schiano y Weiss (Schiano & Weiss, 2006) establece correlaciones entre la investigación existente sobre las causas del problema Y2K¹⁴ y las características de GroupThink. Continúa estableciendo correlaciones similares entre las actitudes hacia la seguridad y GroupThink. A diferencia de investigaciones anteriores en GroupThink, Schiano y Weiss no concluyen que un grupo específico exhibiera GroupThink, sino que generalizaron grupos que trabajaban en estas áreas.

¹³ **TI** Tecnologías de la Información

¹⁴ **Y2K** El problema del año 2000, también conocido como efecto 2000, error del milenio, problema informático del año 2000 o por el numerónimo Y2K (en el que Y=year o año, 2=dos y K=kilo o 1.000), es un *bug* o error de software.



4.6 GroupThink en Ingeniería de Software

Antes de que podamos abordar los métodos para prevenir GroupThink, necesitamos determinar si GroupThink realmente ocurre en muchos grupos de ingeniería de software. Esto se puede hacer mediante una encuesta de la literatura existente en Peopleware.

Diversos investigadores han estudiado las características comunes de los equipos de desarrollo de software. Estas características se pueden utilizar para demostrar que Groupthink causa muchas de las fallas de decisión en el desarrollo de software. El equipo de desarrollo de software típico es cohesionado con respecto a los grupos en general.

4.6.1 Sobreestimación

La primera característica de GroupThink es la sobreestimación del grupo. La sobreestimación de la capacidad del grupo se puede ver en la literatura existente de Peopleware.

Una sobreestimación común que exhiben los equipos se refiere a predecir cuándo se completará el software. Es bien sabido que los ingenieros de software no predicen las fechas de finalización (Yourdon, 1997). Hay dos formas de describir un proyecto que tarda más de lo estimado: el equipo de desarrollo sobreestimó su capacidad o subestimó la complejidad del proyecto. Ambas explicaciones son equivalentes. Un estudio de Michiel Van Genuchten (Van Genuchten, 1991) muestra que las dos razones más comunes por las que los proyectos de software sobrepasan su agenda son, que dedican más tiempo a otros trabajos como el mantenimiento y que subestiman la complejidad del proyecto (Van Genuchten, 1991). Sobreestimar su capacidad para realizar otros trabajos como el mantenimiento provocó que el 27% de los proyectos para llegar tarde. Estos hallazgos muestran que los desarrolladores de software sobreestiman su capacidad para realizar mantenimiento y desarrollar software.

Recientemente, la investigación ha proporcionado una imagen más clara de esta sobreestimación. Jorgensen y Grimstad (Jorgensen & Grimstad, 2012) realizaron una investigación mediante la contratación de más de 350 desarrolladores de software que trabajan en la industria. Se pidió a los desarrolladores que estimaran una tarea breve de desarrollo de software. También se les entregaron tres cuestionarios para medir diferentes características de los participantes. El primer cuestionario mide el autoconstrucción, que es si le dan importancia a la interdependencia o la independencia. El segundo cuestionario midió el estilo de



pensamiento de los sujetos, holístico o analítico. La pregunta final midió su necesidad de cognición, que es si disfrutaban resolviendo problemas. Estas son tres características comunes que se utilizan en los estudios culturales.

Jorgensen y Grimstad (Jorgensen y Grimstad, 2012) encontraron que los individuos que obtuvieron puntajes altos en la prueba de interdependencia tenían más probabilidades de dar una estimación más baja para las tareas de desarrollo de software. Estos resultados apoyan investigaciones previas que concluyen que los desarrolladores de software hacen un mal trabajo de estimación. Otra literatura apoya la correlación entre el deseo de cohesión grupal y la interdependencia (Nisbett, 2003). Teniendo en cuenta que los desarrolladores que valoran la interdependencia también valoran la cohesión del grupo, esta investigación apoya firmemente la correlación entre GroupThink e Ingeniería de software.

Otro resultado interesante de Jorgensen y Grimstad (Jorgensen & Grimstad, 2012) fue que los gestores, desarrolladores con mucha experiencia y desarrolladores con maestría también dan estimaciones más bajas. Estos individuos se consideran líderes del grupo. Este hallazgo es importante porque una de las soluciones para prevenir el GroupThink que se describirá más adelante en este documento es que los líderes no dan opiniones.

Cerpa y Verner (Cerpa & Verner, 2009) realizaron un estudio de proyectos de software fallidos. Su encuesta analizó los factores de los proyectos de software fallidos. Descubrieron que en el 81,4% de los proyectos fallidos se subestimaba la complejidad del proyecto. Esto respalda la afirmación de que muchos ingenieros de software sobreestiman sus habilidades.

Estos estudios muestran una sobreestimación constante de la capacidad y una subestimación del desarrollo de software que abarcan varias décadas. Aunque se sabe comúnmente que las estimaciones de desarrollo de software suelen ser muy bajas, los desarrolladores de software no han podido compensar esto.

4.6.2 Mentalidad cerrada

La segunda característica de GroupThink es la mentalidad cerrada. En las primeras etapas del desarrollo de software, el equipo de desarrollo propone cómo implementar las funciones solicitadas dentro del tiempo y el presupuesto. Desarrollan un plan para hacerlo. En



muchos casos, los proyectos no logran implementar completamente los requisitos dentro de las limitaciones de tiempo y presupuesto. Sin embargo, muchas investigaciones han indicado que los miembros del equipo de desarrollo de software son algunos de los últimos en reconocer este fracaso inevitable (Boehm & DeMarco, 1997).

El estudio de Cerpa y Verner (Cerpa & Verner, 2009) mencionado en el apartado anterior también encontró que en el 75,7% de los proyectos no se siguieron las mejores prácticas de gestión de riesgos. Los riesgos no se identificaron, controlaron ni gestionaron adecuadamente. Una observación de su estudio es que incluso en los casos en que el proyecto obviamente fracasó, los miembros del equipo no admitirían que fracasaron (Cerpa & Verner, 2009). En la mente de los miembros del equipo de estos proyectos, tuvieron éxito.

4.6.3 Presión hacia la uniformidad

Hay varias formas en que los grupos de desarrollo de software muestran presión interna hacia la uniformidad. Hay muchas formas de crear software y, en última instancia, el grupo debe decidir un método. Janus (1972) escribe que hay dos formas de identificar esta presión: la autocensura y la presión de los miembros del grupo para que estén de acuerdo.

La autocensura se ve cuando los esfuerzos de desarrollo de software se retrasan. A menudo, los contratiempos no se informan a la dirección cuando se dan cuenta por primera vez (McConnel, 1996).

La presión sobre los miembros del grupo se ve en muchas fases de desarrollo. Hay presión sobre los miembros de un equipo de desarrollo para que no asuman roles críticos (Carr, 1997). La gerencia recompensa a los miembros que están de acuerdo con las ideas del grupo y castiga a los que señalan los riesgos.

4.7 Métodos para abordar el pensamiento grupal

La investigación en GroupThink ha desarrollado una serie de métodos para prevenir GroupThink sin eliminar la cohesión del grupo. La comunidad de ingenieros de software ha adoptado algunos de estos métodos. Esto también refuerza el argumento de que los equipos de ingeniería de software sufren de GroupThink. Las causas de GroupThink no solo existen en la



ingeniería de software, sino que también las soluciones para GroupThink aumentan la calidad del software (Brown, 2014).

4.7.1 Asignar papel de crítico

Un método para prevenir GroupThink es asignar un papel de crítico. Muchas personas tienen dificultades para plantear problemas con la idea de alguien. Esto se considera demasiado conflictivo. Asignar un papel de crítico reduce este estrés. El crítico debe plantear problemas potenciales con la idea. Ese es su papel. La mayoría de los métodos de desarrollo de software han incluido esta solución en forma de revisiones de diseño y código. Muchas investigaciones han indicado que las revisiones de este tipo son beneficiosas (Constantine, 1995; Kemerer y Paulk, 2009).

4.7.2 Los líderes evitan hacer sugerencias

Los investigadores que estudian GroupThink creen que los grupos pueden tomar mejores decisiones si los líderes del grupo evitan hacer sugerencias (Janus, 1972). Cuando los líderes presentan sugerencias, otros miembros del grupo se apresuran a estar de acuerdo con ellas. Esto evita que el grupo considere completamente todos los enfoques posibles. Se ha realizado poca o ninguna investigación sobre esto en el campo de la ingeniería de software. Esta podría ser una posible área de investigación futura.

4.7.3 Evaluación independiente

Un tercer método para prevenir GroupThink es la evaluación independiente. Esto también se utiliza en ingeniería de software. El campo de la Verificación y Validación Independiente (IV&V) se basa en esta filosofía. IV&V es utilizado por muchas organizaciones, incluida la NASA. Existe una variedad de investigaciones que indican que la IV&V reduce los costos y aumenta la calidad (Akella & Rao, 2011).

4.7.4 Hacer sugerencias anónimas

Puede haber otra forma de prevenir GroupThink sin eliminar las aportaciones de los líderes del grupo. Si la entrada del grupo es anónima, los miembros del grupo no apoyan automáticamente las opiniones de sus superiores. No saben qué aportes vinieron de los líderes del grupo. La investigación sugiere que esto puede mejorar la toma de decisiones en grupo (Kontio, et al., 2004).



4.8 Cómo evitar el pensamiento grupal

Muganda, Asmelash y Samali (2012) estos autores indican que el desafío para cualquier líder de equipo o grupo es crear un entorno de trabajo en el que sea poco probable que ocurra el pensamiento grupal. También es importante comprender los riesgos del pensamiento grupal: si hay mucho en juego, debe hacer un esfuerzo real para asegurarse de tomar buenas decisiones.

Para evitar el pensamiento grupal, es importante contar con un proceso para verificar los supuestos fundamentales detrás de las decisiones importantes, para validar el proceso de toma de decisiones y para evaluar los riesgos involucrados. Para decisiones importantes, asegúrese de que su equipo haga lo siguiente en su proceso de toma de decisiones:

- a) Explora objetivos.
- b) Explora alternativas.
- c) Fomenta el desafío de las ideas sin represalias.
- d) Examina los riesgos si se elige la opción preferida.
- e) Prueba supuestos.
- f) Si es necesario, retrocede y vuelve a examinar las alternativas iniciales que fueron rechazadas
- g) Recopila información relevante de fuentes externas
- h) h) Procesa esta información de forma objetiva.

Hay muchas técnicas grupales que pueden ayudar con esto, incluidas las "Herramientas mentales" que se enumeran a continuación en la tabla 4.3. Al utilizar una o más de estas técnicas para lograr aspectos del trabajo del grupo, variará las formas de trabajo del grupo y, por lo tanto, se protegerá del pensamiento grupal y ayudará a tomar mejores decisiones.



Tabla 0.3 Técnicas grupales

Técnicas grupales:	
Lluvia de ideas	Ayuda a que las ideas fluyan libremente sin críticas.
Recuento de Borda modificado	Permite que cada miembro del grupo contribuya individualmente, mitigando así el riesgo de que miembros del grupo más fuertes y persuasivos dominen el proceso de toma de decisiones.
Seis sombreros para pensar	Ayuda al equipo a ver un problema desde muchas perspectivas diferentes, lo que permite a la gente jugar al "Abogado del diablo".
La técnica Delphi	Permite que los miembros del equipo contribuyan individualmente, sin conocimiento de una vista grupal y con poca penalización por el desacuerdo.
Herramientas de apoyo a la toma de decisiones:	
Análisis de riesgo	Ayuda a los miembros del equipo a explorar y gestionar los riesgos.
Análisis de impacto	Garantiza que se exploren a fondo las consecuencias de una decisión.
La escalera de la inferencia	Ayuda a las personas a verificar y validar los pasos individuales de un proceso de toma de decisiones.

Fuente: (Muganda, Asmelash y Samali, 2012)



4.9 Conclusión

El GroupThink siempre es un riesgo en las organizaciones y por ende en los proyectos de TI al momento de levantar los requerimientos, el tener que llegar a un consenso en las soluciones. Contar con el apoyo de una decisión grupal, especialmente cuando no se consideran todas las alternativas potenciales para la resolución de problemas. Se convierte en una realidad cuando los individuos simplemente se ajustan a la opinión de la mayoría y dudan en sugerir sus propias soluciones a un problema.

Si se logra construir un equipo de trabajo poderoso se podría evitar GroupThink con decisiones que se basan en opiniones de los miembros que son dominantes, y de quienes se acepta la idea pese a que se pueda tener una opinión distinta. Y es que con el groupthink se crea una polarización que, al final, pretende reflejar un grupo cohesivo, ya que no existe la confianza de ser honestos y de expresar otro punto de vista. Así se crea la ilusión de unanimidad.

Construir confianza en equipo es un arduo trabajo que empieza por valorar a los demás. Debe realizarse a todo nivel y en ambos sentidos de la jerarquía organizativa, entendiendo que somos diferentes y debemos respetar las ideas, pensamientos y actitudes de los miembros del grupo. Por ejemplo, algunos asimilamos información de forma más sensorial, que implica valorar los sentidos y confiar en los hechos y la experiencia; mientras que otros la asimilamos intuitivamente, lo que implica mirar el conjunto para trabajar en las posibilidades. Al ser diferentes podemos fortalecernos con las diferencias, solo si las respetamos.

Para fortalecer la relación de un equipo se puede comenzar realizando un reconocimiento de los logros o ideas planteadas de los miembros, a fin de crear seguridad y confianza para la contribución al grupo. La idea es fortalecer el autoconocimiento de cada uno para aportar al grupo. Esto es importante y debe darse de una manera sincera y abierta.

Asimismo, se debe fomentar una interacción entre todos los miembros para no formar subgrupos en el que, por ejemplo, uno sea el dominante y no permita la participación de todos. Esto implica respeto y apertura a todos los aportes, los cuales se deben realizar en un marco conforme al tema tratado.



Existen algunos métodos y técnicas para abordar GroupThink que podrían aplicarse al momento de realizar el relevamiento de los requerimientos en el proyecto, algunas de estas se encuentran mencionadas en este capítulo. Una solución que sugiere Janus es tener un líder que no dé opiniones. Esto elimina el deseo de las personas de complacer a sus superiores estando de acuerdo con ellos. Actualmente no se ha realizado ninguna investigación en ingeniería de software sobre este enfoque. Esta podría ser una investigación futura.

Es importante que los CEO y directivos de altos mandos de las organizaciones tomen conciencia de la importancia de seguir procedimientos metódicos en el proceso de toma de decisiones, para disminuir la probabilidad de que se creen condiciones en la dinámica grupal que favorezcan la formación de fenómenos psicosociales, como el GroupThink, que conduzcan a decisiones pobres o erróneas. El aprendizaje de procedimientos metódicos para el proceso de toma de decisiones debe formar parte del instrumental básico de los proyectos de TI e inclusive de la cultura como tal de las empresas. Esto no implica hacer más lentos los procedimientos, sino asumir la necesidad de aprender a tomar decisiones complejas ágilmente. Generando que los grupos de trabajo sean poderosos ya que todos van a luchar hacia un ideal en el cual creen y no solo por apoyo al grupo dominante, creando así un groupthink con resultados positivos.



Universidad de Buenos Aires
Facultad de Ciencias Económicas
Escuela de Estudios de Posgrado



Capítulo V. Participación del Cliente y Herramientas de Relevamiento



5.1 Introducción

En épocas de reinención, de cambios coyunturales e incertidumbre, las empresas buscan constantemente continuar cumpliendo con las expectativas de sus clientes y mejorar sus servicios.

En la actualidad, se requieren organizaciones que tengan capacidad de adaptarse a los cambios, de transformarse culturalmente. Para eso, resulta necesario contar con compañías que sean flexibles e innovadoras en su cultura de trabajo. Es importante que cada día más las empresas trabajen con estructuras organizativas horizontales, dando lugar a que todos puedan proponer nuevas ideas (González, 2021).

Debemos considerar que gran parte de los proyectos que se deban relevar de hoy en adelante estarán orientados a tener una mayor participación del Cliente, además de darle una gran importancia al usuario final que consumirá las aplicaciones. Como menciona Kulbyte (2021) :

- ✓ La experiencia del cliente. - Ya que se ha convertido rápidamente en una prioridad para las empresas y para los años subsiguientes no será diferente. Pero, ¿por qué tantas empresas se centran en la experiencia del cliente y qué sucede con las empresas que eligen ignorarla?
- ✓ Las compras entre dispositivos a través de una amplia gama de canales han dificultado que las empresas mantengan la coherencia. Los procesos y las tecnologías deben cambiar para proporcionar una experiencia coherente en todas las plataformas.

Para poder gestionar de una mejor forma los requerimientos y mejor comunicación con los clientes, una buena propuesta es implementar Herramientas Innovadoras de Relevamiento. Gartner (2014) confirma la importancia de los requisitos, en su informe de investigación: “Guía de mercado para la definición de requisitos de software y soluciones de gestión”, enfatizando la necesidad de mejorarlos y brindar una buena orientación para aumentar su calidad, logrando además una mejor comunicación con el cliente.

Reconoce que los requisitos deficientes continúan contribuyendo significativamente a la reelaboración del proyecto y la fricción entre las empresas y las partes interesadas de TI. También revela lo que cree que es una de las principales razones por las que las organizaciones aún luchan con la calidad de los requisitos: las organizaciones no han adoptado las herramientas adecuadas para respaldar la definición de requisitos y las prácticas de gestión de alta calidad.



5.2 La importancia de la participación del cliente en los proyectos de software

Los beneficios de la participación del cliente durante los proyectos de desarrollo de software. En un mundo ideal, un representante del cliente estaría ubicado junto al equipo de desarrollo. Cuando esto no sea posible, la tecnología moderna proporciona amplias (y generalmente gratuitas) formas de permanecer en contacto con su equipo de desarrollo. Cualquiera que sea el medio de comunicación, Solidetech (2020) reseña los beneficios de la participación frecuente del cliente: Mayor claridad de los requisitos del cliente y priorización puntual.

Nadie conoce su negocio mejor que el own client o su Business Team (equipo de negocio). Como tal, le conviene permitir que el equipo de desarrollo que implementa los procesos comerciales centrales se beneficie de su participación constante. Esta necesidad de salvaguardar la precisión de cómo su empresa se traduce en código no solo se aplica a las primeras etapas de un proyecto. La realidad es que en el transcurso del proyecto las necesidades de su negocio pueden cambiar, o al menos puede darse cuenta de que los requisitos funcionales iniciales eran inadecuados o inviables. A través de una comunicación constante, sus necesidades cambiantes pueden adaptarse rápidamente y puede evitarse el desarrollo de funcionalidades obsoletas o innecesarias.

5.2.1 Menos desperdicio financiero y de recursos.

Al asegurarse de que el equipo de desarrollo siempre reciba comentarios oportunos sobre los requisitos, ahorra dinero y tiempo al permitir que los desarrolladores se concentren solo en lo que usted necesita. Los proyectos de software suelen sobrepasar el presupuesto, y se ha dicho que después de un proyecto de software mal gestionado, solo se utiliza realmente el 40% de la funcionalidad. Si sigue siendo un cliente comunicativo y estrechamente involucrado, puede evitar todo este desperdicio. Los ciclos de vida de desarrollo de software que siguen el enfoque ágil, que son comunes en la industria actual, intentan incorporar comentarios frecuentes de los clientes en el proceso; sin embargo, la decisión final de participar sigue siendo del cliente.

5.2.2 Refinamiento más cuidadoso del software a través de demostraciones y pruebas.

Incluso si se comunica diligentemente con su equipo de desarrollo y se asegura de que estén siempre actualizados con respecto a sus requisitos en evolución, no puede cosechar todas



las recompensas de su participación sin una interacción práctica frecuente con su software. Un buen equipo de desarrollo se asegurará de que pueda interactuar con la nueva funcionalidad a medida que se desarrolle, en lugar de hacerle esperar hasta que todo esté terminado para tener su primera mirada. Esto es esencial porque hay tres fuentes de error potencial cuando sus requisitos se expresan verbalmente:

1. Una imperfección en su expresión de la necesidad;
2. Una imperfección en la comprensión del equipo de desarrollo de su necesidad;
3. Una imperfección en la implementación de la solución por parte del equipo de desarrollo.

Las demostraciones frecuentes lo ayudan como cliente a reconocer la funcionalidad faltante como resultado de un malentendido o identificar necesidades que aún no había considerado hasta que tuvo la oportunidad de usar el software.

5.3 Problemas comunes con la participación del cliente durante un proyecto de software

Aunque hemos considerado los grandes beneficios de la participación cercana del cliente en los proyectos de software, es importante tener en cuenta al menos dos dificultades comunes:

a. Tenga cuidado de abordar los síntomas en lugar de las causas fundamentales

Con un equipo de desarrollo disponible para resolver todos sus problemas, puede parecer más fácil tratar cada ineficiencia empresarial recién descubierta como un problema separado para que el equipo de desarrollo lo resuelva. Sin embargo, todos estos problemas pueden ser el resultado de una causa subyacente común. Pensar detenidamente más allá de la solución rápida puede revelar el problema real, y abordar esto desde el principio resultará en un proyecto de software más cohesivo y se ahorrará mucho tiempo y costos. Alternativamente, un poco de escrutinio a veces puede revelar que la enorme y costosa solución que había imaginado puede reducirse a un refinamiento simple y efectivo.

b. Usted es el experto en negocios, pero la implementación es dominio de los desarrolladores

Los clientes son lógicamente la fuente de la verdad al articular los problemas / necesidades comerciales que el software puede resolver. Sin embargo, como cliente, debe obsesionarse **con qué y por qué**, y evitar pensar demasiado en **cómo**. Los desarrolladores son expertos en comprender el contexto técnico de su software, por lo que le conviene prestar



atención a sus consejos sobre la forma más factible de implementar la solución a sus necesidades.

Si desea asegurarse de obtener el mayor retorno de su inversión en software empresarial, asegúrese de trabajar con un equipo de desarrollo que priorice la participación cercana del cliente durante todo el proyecto. ¡Esté dispuesto a considerarse un participante vital en el proceso y observe cómo sucede la magia!

5.4 Que es Customer Experience

Cuando desarrollamos un software para un cliente se debe tomar en consideración todo lo indicado por el Business Team, pero hoy en día en la transformación digital acelerada que estamos atravesando debemos estar preparados, junto con nuestras aplicaciones para hacer frente a la experiencia del usuario final que es nuestro nuevo desafío.

Por lo tanto, La experiencia del cliente es la percepción que tienen sus clientes de cómo los trata su empresa. Estas percepciones afectan sus comportamientos y construyen recuerdos y sentimientos para impulsar su lealtad. (Kulbyte, 2021).

La buena noticia es que no importa en qué tipo de negocio se encuentre: se ha demostrado (una y otra vez) que mejorar la experiencia de sus clientes aumenta la retención, la satisfacción y los ingresos. (Kulbyte, 2021).

5.4.1 La experiencia del cliente móvil es una prioridad

Kulbyte (2021) en su informe estadístico indica que cuando se trata de brindar una experiencia positiva a través de diferentes canales, se espera que el servicio al cliente móvil se dispare. La razón es simple: ¡una mala experiencia móvil puede dañar seriamente su marca!

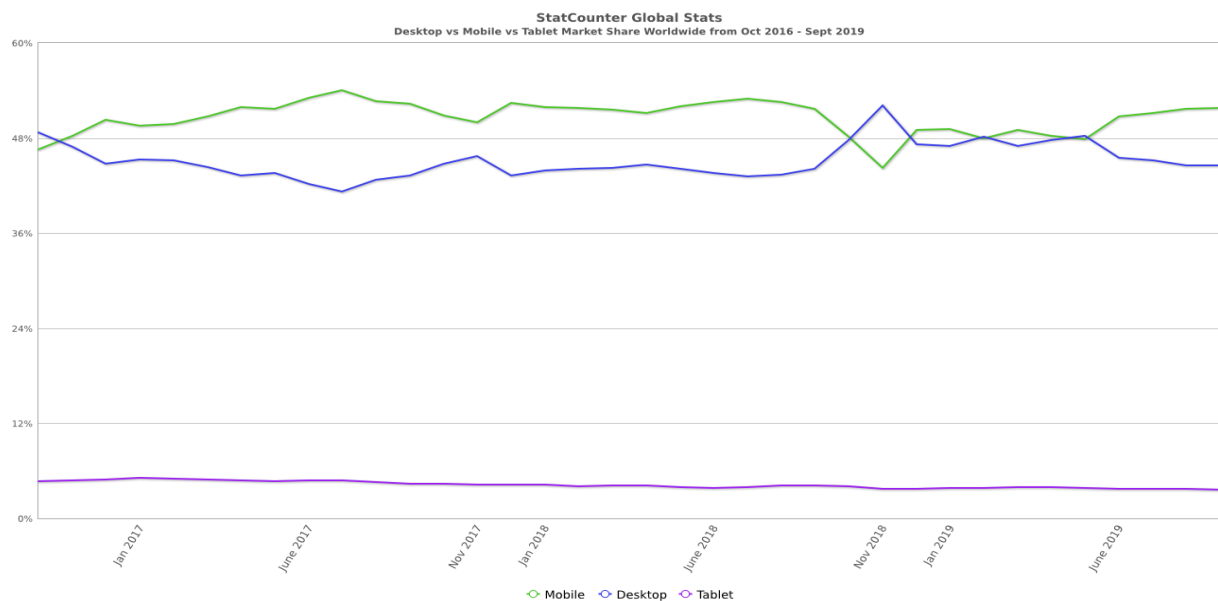
Según Kolsky (2017) comparte su visión de mejorar la experiencia del cliente con Internet de las cosas o Internet de los clientes. Internet de las cosas (IoT) cambiará los modelos comerciales y requisitos de prestación de servicios. La internet de los clientes es algo que todos los ejecutivos deben comprender e incluir como parte de su plan de transformación digital. Kolsky cree que IoT transformará los servicios al cliente y cambiará significativamente las expectativas del cliente. Las máquinas se integrarán en la vida y los negocios de los consumidores, produciendo datos sobre los que se puede actuar en tiempo real, con acciones específicas que se personalizan para ofrecer un valor significativo. En una economía hiperconectada, la expectativa será que las empresas brinden servicios proactivos basados en el análisis en tiempo real de los datos generados por el cliente (y el dispositivo o "cosa").

Por ejemplo, el 57% de los clientes no recomendarán una empresa con un sitio web mal diseñado en dispositivos móviles. Y si un sitio web no está optimizado para dispositivos móviles, el 50% de los clientes dejarán de visitarlo, incluso si les gusta el negocio.

Según Stat Counter (2020), el 52% de todo el tráfico de Internet ahora proviene de dispositivos móviles, y el uso de computadoras de escritorio tiene una tendencia a la baja.

Por tanto, no sorprende que el 84% de las empresas que afirman estar centradas en el cliente se centren ahora en la experiencia del cliente móvil, como se puede ver en la gráfica.

Figura 5.1.1. Comparativo de uso de Internet



Fuente: Counter (2020) Recuperado de <https://gs.statcounter.com/platform-market-share/desktop-mobile-tablet/worldwide/#monthly-201610-201909>

Sin embargo, para las empresas que han tardado en adaptarse a esta nueva tendencia, especialmente en lo que respecta a la atención al cliente, ya que un abrumador 90% de los clientes informa tener una mala experiencia al buscar atención al cliente en dispositivos móviles.

5.4.1 La frustración del cliente provocara deserción

Según Esteban Kolsky fundador de organización de consultoría Think Tank de estrategia de clientes en el 2017, indica que el 72% de los clientes compartirán una experiencia positiva



con 6 o más personas. Por otro lado, si un cliente no está contento, el 13% de ellos compartirá su experiencia con 15 o incluso más.

El desafío aquí radica en el hecho de que, en la mayoría de los casos, los clientes no le dicen que están descontentos. De hecho, solo 1 de cada 26 clientes insatisfechos se queja. El resto, simplemente se van, afirma Kolsky.

Pero, como muestra la investigación, es posible que sus clientes no estén contentos o, peor aún, que compartan sus malas experiencias con los demás. Las expectativas de los clientes están en su punto más alto y es un momento difícil para que las empresas las cumplan y superen.

Incluso si brinda una experiencia positiva al cliente 9 de cada 10 veces, esa única vez que no lo haga podría ser fatal.

En su informe sobre el futuro PwC en el 2018 encuestó a 15.000 consumidores y descubrió que 1 de cada 3 clientes dejará una marca que ama después de solo una mala experiencia, mientras que el 92% abandonaría por completo una empresa después de dos o tres interacciones negativas.

5.5 Herramientas de Recopilación de Requisitos

Las herramientas de recopilación de requisitos son recursos que puede utilizar para completar el proceso de recopilación de requisitos con mayor facilidad, rapidez y éxito (Minestra, 2021).

Las herramientas de gestión de requisitos que necesitan las organizaciones serán diferentes según la metodología y los objetivos de su proyecto. Del mismo modo, su enfoque de los requisitos variará. Por ejemplo, algunos equipos utilizan otra palabra para los requisitos, como "historias de usuario", "requisitos del producto" o simplemente, "características" (Aston, 2021).

5.6 ¿Por qué son importantes las herramientas de recopilación de requisitos?

Gartner (2014), reconoce que las sólidas herramientas de gestión y definición de requisitos pueden mejorar la productividad y la calidad por una buena razón.

Si bien las herramientas para desarrolladores y evaluadores son maduras y se han adoptado ampliamente, los analistas de negocios han confiado durante mucho tiempo en Microsoft Word y Excel o en herramientas de ingeniería de requisitos arcaicos para administrar



los requisitos. No han tenido otra opción, porque las herramientas de requisitos sólidas y especialmente diseñadas no existían hasta hace poco.

Hoy, los analistas de negocios tienen opciones mucho mejores. Las herramientas de requisitos modernas les permiten obtener, definir y gestionar rápidamente requisitos de alta calidad mediante un conjunto de herramientas flexible de características y artefactos.

Como explicó la experta en gestión de proyectos Kelly Suter (2018) , en su guía completa para la recopilación de requisitos: "Cuando los presupuestos son escasos, los plazos son ajustados y el alcance es cada vez mayor, la documentación de requisitos tiende a ser el primer producto que se entrega y el último que se debe considerar".

5.7 Las 10 mejores herramientas de recopilación de requisitos

A continuación, se muestran algunas de las mejores herramientas de gestión de requisitos disponibles en la actualidad. (Mistretta, 2021)

1. Jama Software

Jama Software proporciona una plataforma que funciona bien con el desarrollo y las pruebas de software. Es una opción adecuada para equipos que están construyendo sistemas y productos complejos. Algunas de las características más importantes de esta plataforma incluyen:

- ✓ Mejora los tiempos de ciclo y la eficiencia.
- ✓ Reduce la redundancia
- ✓ Asegura alta calidad
- ✓ Tiene una interfaz intuitiva basada en navegador.

La plataforma de Jama permite a los equipos alinear hardware, partes interesadas y firmware. Es lo suficientemente flexible para admitir diferentes metodologías de desarrollo. Y proporciona una captura, comunicación, requisitos, objetivos y progreso más precisos.

2 CodeBeamer ALM

CodeBeamer ALM es fácil de usar e incorpora integraciones de API. Es una buena herramienta de recopilación de requisitos , también tiene algunas opciones de personalización. Algunas áreas en las que esta plataforma sobresale incluyen:

- ✓ Gestión de riesgos
- ✓ Gestión de la línea de productos
- ✓ Prueba de preguntas y respuestas



- ✓ Seguimiento de interdependencias

CodeBeamer ALM es lo que se conoce como una plataforma de gestión de estilo de vida de aplicaciones. Su plataforma es integrada, colaborativa y flexible.

3. Accompa

Muchas organizaciones eligen la plataforma de Accompa simplemente porque es fácil de usar e implementar. Lo que le falta en características robustas, lo compensa siendo fácil de usar. Y agregan nuevas funciones de vez en cuando.

Algunas de las razones para elegir esta herramienta de recopilación de requisitos incluyen:

- ✓ Característica de trazabilidad de requisitos
- ✓ Tableros de discusión
- ✓ Alertas para colaboradores
- ✓ Vistas, campos y roles de usuario personalizados

Si su empresa requiere una plataforma más completa, es probable que desee buscar en otra parte. Sin embargo, para muchas pequeñas empresas, Accompa tiene suficientes funciones útiles para satisfacer sus necesidades.

4. Caliber

Calibre es una buena herramienta de recopilación de requisitos para guiones gráficos y simulaciones. Funciona bien para probar roles, procesos y requisitos. También tiene excelentes características visuales para los requisitos de modelado. Otras características notables de Calibre incluyen:

- ✓ Herramientas de requisitos rastreables
- ✓ Es más visualmente interactivo
- ✓ Controles de fácil acceso de usuario
- ✓ Una colección de las funciones más útiles

Calibre utiliza un sistema de arrastrar y soltar para los requisitos de lectura. En lo que respecta a las herramientas de recopilación de requisitos, es una buena herramienta independiente y es más fácil de usar que otras. Sin embargo, no se integra bien con otras herramientas de recopilación de requisitos.



5. Perforce Helix RM

Esta herramienta de recopilación de requisitos funciona bien con equipos grandes y distribuidos. A menudo es elegido por empresas que buscan una mejor trazabilidad y funciones de gestión de casos de prueba. Algunas otras razones para usar Perforce Helix RM incluyen

- ✓ Es escalable
- ✓ Maneja archivos grandes fácilmente
- ✓ Servicio de atención al cliente receptivo
- ✓ Incorpora muchas funciones útiles

El único golpe importante en esta plataforma es que, si bien es poderoso, lleva mucho tiempo configurarlo y aprenderlo. La implementación puede resultar difícil. En general, no es la herramienta de recopilación de requisitos más fácil de usar. Pero para compensar estas dificultades, Perforce Helix RM tiene algunos tutoriales en video que los usuarios pueden encontrar útiles y probablemente necesitarán.

6. Pearls

Pearls es una herramienta de recopilación de requisitos que se destaca cuando se trata de proporcionar funciones útiles de colaboración en equipo. Crea fácilmente roles de usuario, atrasos y es suficiente en la gestión de especificaciones. También es asequible y tiene una variedad de características que incluyen:

- ✓ Genera documentos de Word con un solo clic
- ✓ Comentarios y configuración de notificaciones para la actividad de los miembros
- ✓ Un sistema de precios flexible
- ✓ Genera fácilmente documentos de requisitos y especificaciones

Pearls es más adecuado para pequeñas empresas e incluso tiene un plan de inicio gratuito para tres o menos usuarios. También tiene un sistema de precios escalable para empresas más grandes. Y como la mayoría de las herramientas de recopilación de requisitos, incluye funciones para gestionar las partes interesadas y definir los objetivos del proyecto.

7. ReqSuite

ReqSuite se vende a sí mismo como altamente personalizable, pero es más exacto decir que camina por una delgada línea entre la simplicidad y la personalización. Define fácilmente los flujos de trabajo relevantes. También crea fácilmente casos de uso y requisitos reutilizables. Las razones para elegir ReqSuite como su herramienta de recopilación de requisitos incluyen:



- ✓ Muchas plantillas definibles
- ✓ Herramientas de modelado visual
- ✓ Numerosas funciones de colaboración
- ✓ Es fácil de navegar

La facilidad de uso puede ser la mejor razón para elegir ReqSuite, ya que no es demasiado complejo y los tutoriales son realmente útiles. Sin embargo, tiene menos capacidades en comparación con herramientas de recopilación de requisitos más sólidas.

8. Visure

Visure es una buena herramienta de recopilación de requisitos específicamente en el área de gestión de la configuración. Realiza un seguimiento fácil de la trazabilidad, la verificación y los defectos. Algunas de las funciones más populares que los clientes disfrutarán son:

- ✓ Es fácil de usar
- ✓ Es particularmente útil para la gestión de la trazabilidad.
- ✓ Tiene un proceso de revisión de requisitos simplificado
- ✓ Ofrece una plataforma de requisitos centralizada

Visure puede soportar una variedad de procesos del ciclo de vida desarrollo (Agile, modelo V, Waterfall y otros). Puede integrarlo con productos de Microsoft como Word y Excel. Y esta herramienta de recopilación de requisitos ofrece plantillas de cumplimiento para una variedad de estándares de seguridad internacionales.

9. Orcanos

Esta herramienta de recopilación de requisitos es especialmente útil para informes y visualización. La plataforma Orcanos crea y edita fácilmente documentos de especificaciones y requisitos de productos, aunque sus funciones de colaboración son bastante básicas. Algunas de las razones por las que los usuarios eligen Orcanos incluyen:

- ✓ Buen soporte al cliente
- ✓ Paneles de control en tiempo real
- ✓ Funciones de navegación que parecen intuitivas
- ✓ Importa y exporta fácilmente documentos de Word

Orcanos tiene mejores funciones de informes que otras herramientas de recopilación de requisitos en esta lista, pero también carece de integraciones prediseñadas. También deberá



actualizar su plan para algunas funciones adicionales como la gestión de quejas de los clientes, el control de calidad, la evaluación de riesgos y otras.

10. MindManager

Una razón por la que muchas personas eligen MindManager como su herramienta de recopilación de requisitos es la facilidad con la que puede crear un panel central de información que segmenta los datos visualmente e incluye enlaces y archivos adjuntos para expandir el contenido de la página. Algunas otras características de MindManager que los clientes encuentran útiles incluyen:

- ✓ Se puede ordenar fácilmente mediante etiquetas y filtros.
- ✓ Su diseño altamente adaptable se adapta a una variedad de necesidades
- ✓ Atención al cliente insuperable
- ✓ Permite a los equipos colaborar en tiempo real en mapas a través de MS Teams.

MindManager facilita a los equipos la colaboración en documentos de requisitos mediante la coedición (lo que permite que varias personas editen el documento a la vez). El diseño de MindManager también se puede adaptar según sea necesario para adaptarse a cómo las partes interesadas clave asimilan la información, lo cual es útil para remediar cualquier desconexión entre los miembros del equipo y las partes interesadas. Y tiene potentes herramientas de visualización y software de mapas mentales.

5.8 ¿Cómo elegir la mejor Herramienta para su Equipo?

A fin de cuentas, aprender a utilizar e implementar una herramienta de gestión de requisitos es una inversión de tiempo y energía. Antes de sumergirse, debe asegurarse de que está eligiendo la herramienta ideal para sus proyectos.

Simplifico el proceso brindando algunas de las preguntas más importantes que creo se pueden hacer durante el proceso de selección para reducir las opciones (Aston,2021).

- ✓ ¿Para qué lo necesitas?
- ✓ ¿Necesita algo que esté un paso por delante de Google Docs y una hoja de cálculo, o está buscando una herramienta de nivel empresarial que pueda admitir grandes volúmenes de datos y equipos distribuidos?
- ✓ ¿Necesita funcionalidades adicionales como gestión de tareas, gestión de riesgos o la capacidad de crear maquetas?



- ✓ ¿Utilizará el software específicamente para pruebas de software?
- ✓ ¿Quiere comprar una licencia o acceder a un software basado en la nube?
- ✓ ¿Qué tan fácil es implementar y aprender?
- ✓ ¿Cuáles son los puntos principales que cree que debería cubrir una solución de software de gestión de requisitos?
- ✓ ¿Mejorara mi comunicación con el cliente?

5.9 Conclusión

La transformación digital y la innovación del modelo de negocio resultante de los mercados disruptivos que estamos atrasando a raíz de la Pandemia Mundial, tiene el potencial de cambiar el mundo tal como lo conocemos y cambiará fundamentalmente los modelos de negocio de muchas empresas, así como las expectativas y los comportamientos tanto de los clientes, como los usuarios finales (consumidores) interactúan con estas empresas y otras partes interesadas (Fredette, Marom, Steinert Y Witters, 2021).

Por lo tanto, desde la gestión de requerimientos debemos estar a la vanguardia de la innovación, permitiendo mayor colaboración de Los usuarios finales y los clientes ya que son los que encargan de indicar las necesidades, lo que se pueden utilizar para priorizar, qué características de un producto se desarrollan primero, son los que validar la aceptación del producto y determinaran si el producto de software exitoso.

En esta tesis hemos abordado metodologías y dejamos al criterio de la organización la que consideren que más se adapte a sus necesidades, pero en este punto podría adecuarse trabajar centrándose en iteraciones cortas y al ejercer una transparencia del 100% en su trabajo, los desarrolladores ágiles también invitan a los comentarios y sugerencias de los clientes e incluso a cambios de requisitos. De esta manera, los clientes también pueden cambiar el enfoque futuro de los desarrolladores sin interrumpir el proceso de desarrollo del producto. Finalmente, con este tipo de configuración, también es mucho más fácil responder a los cambios en el mercado y los clientes y los proveedores siempre deben abordarlos de manera colaborativa.

Todo esto tiene como objetivo mejorar el producto y hacerlo más parecido al producto que el cliente imaginó originalmente.



Adicional el uso de herramientas adecuadas para la gestión de los requisitos también podrá servir de impulso para mejorar la comunicación tanto con el cliente y el ciclo de vida del proyecto.

El Business Team se benefician de la adopción de una herramienta, pueden trabajar directamente en la herramienta como colaboradores activos y desarrolladores de requisitos. Pueden recibir documentación de requisitos en formatos con los que se sientan cómodos, como documentos y páginas web. Pueden digerir información y proporcionar comentarios valiosos fácilmente de una manera que les funcione.

Los desarrolladores y probadores obtienen la ventaja de recibir requisitos de consumibles más completos en sus propios conjuntos de herramientas, donde pueden procesarlos fácilmente. También pueden proporcionar información y hacer preguntas desde sus entornos de trabajo sin necesidad de iniciar sesión en un sistema diferente. La generación automática de historias de usuario, criterios de aceptación y casos de prueba aligera sustancialmente su carga.

Una herramienta de requisitos moderna cierra la brecha entre el negocio y la TI y ayuda a los equipos a entregar productos adaptados a las necesidades del negocio.



Universidad de Buenos Aires
Facultad de Ciencias Económicas
Escuela de Estudios de Posgrado



Capítulo VI. Labor de Campo y Conclusiones



6.1 Introducción

El análisis de los requisitos es una parte vital en el desarrollo de proyectos de software, puesto que define el propósito, la dirección y el tamaño del proyecto y, por tanto, en mayor o menor medida, condicionará el éxito global del mismo. Una adecuada gestión de requisitos permite analizar, validar y verificar los mismos, facilitando el alcance de objetivos de negocio y minimizando riesgos asociados, costos innecesarios, siendo por ello un proceso de soporte estratégico.

Las organizaciones deben estar preparadas para este mundo VICA (Volátil, Incierto, Complejo y Ambiguo), considerar todos los posibles escenarios que puedan afectar el normal desarrollo de un proyecto, lo que implica identificar riesgos y prepararse para afrontarlos. Pero también aprovechar oportunidades. Para ello se requiere un cambio de paradigma, líderes enfocados a procesos para la toma de decisiones.

El cambio parte con contar un pensamiento sistémico en la gestión de los requisitos, que permite la comprensión de la dirección para llegar a los objetivos deseados y la dinámica que hay dentro de estos procesos. Esto requiere a su vez considerar las teorías asociadas a distintas disciplinas, haciendo más fácil identificar las problemáticas generadas y por consiguiente darles solución adecuada para cada caso y sus interrelaciones.

Esta investigación desea demostrar, a través de una investigación de campo, cómo los requerimientos afectan de forma significativa la calidad del software y, por ende, los costos asociados al proceso de desarrollo de software.



6.2 Hipótesis

Esta investigación se ha planteado que:

H1. La falta de un procedimiento sistémico de recopilación y evaluación de requisitos para el desarrollo de software puede llegar a introducir fallas de calidad, adecuación y agregar tiempos y costos al proyecto.

H2. La falta de herramientas y metodologías de gestión de proyectos en el desarrollo de software puede llegar a introducir fallas de calidad, adecuación y agregar tiempos y costos al proyecto.

H3. La utilización de buenas prácticas de ingeniería de requerimientos y mapas mentales de apoyo al proceso de análisis y diseño de software puede contribuir a mejorar la calidad, adecuación y reducir tiempos y costos del proyecto.

H4. El rasgo de fuerte presencia de pensamiento grupal conspira contra la obtención, en el análisis y diseño de software, de buenos resultados a nivel de calidad, adecuación y tiempos y costos del proyecto.

H5. La falta de participación del cliente en el proyecto y el uso de herramientas obsoletas que no se adaptan a los cambios disruptivos de la transformación digital que se está atravesando actualmente, puede acarrear fallas en el triángulo de hierro.

6.3 Investigación de Campo

6.3.1 Introducción

La investigación de campo realizada ha sido de tipo cualitativo y exploratoria apoyada en la mecánica del caso.

Los casos fueron elegidos por decisión razonada, utilizando como criterios de selección:

- Se buscó obtener variedad de empresas posibles, considerando tecnología, experiencia en proyectos y servicio.
- Experiencia de los encuestados en el área de proyectos
- Que las empresas cuenten con procesos sistemáticos para la gestión de los proyectos
- Usen técnicas para realizar el relevamiento de requerimientos



Fueron entrevistados 10 personas de la industria del desarrollo de software, comprendiendo ello directores y analistas a cargo de la determinación de los requerimientos dentro de los proyectos.

Se trabajó para las entrevistas con una encuesta donde se realizaron preguntas de tipo situacional y proyectivo donde el entrevistado opina a partir del colectivo que representa.

El objetivo de esta labor fue comprender el complejo mundo de la experiencia vivida desde el punto de vista de las personas que trabajan en tareas de desarrollo de software en materia de relevamiento de requisitos (Taylor y Bogdan, 1984).

Se pretendió conocer el contexto y proceso donde se produce el levantamiento y análisis de los requerimientos, buscando encontrar evidencias que se alineen con el marco teórico. Comprobar el uso de buenas prácticas de proyectos para la gestión de los nuevos proyectos, Conocer la calidad final de los proyectos, en base a los procesos que actualmente manejan, Identificar si en sus equipos de trabajo se encuentra presente el GroupThink, Comprender la fenomenología de los proyectos las variables intervinientes y sus relaciones.

Se abordó un análisis de las diferentes situaciones y escenarios que se presentan dentro de los proyectos, que estuvieron al alcance de los entrevistados, al momento de relevar la información, asimismo contrastando proyectos exitosos y otros que no cumplieron con los requerimientos dentro de lo establecido en el *triángulo* de hierro.

6.3.2 Composición del colectivo estudiado

Se realizó el estudio a partir de una encuesta que se adjunta en Anexo 1, organizada en tres secciones. La primera y segunda orientadas a permitir segmentar las conclusiones, que involucran, respectivamente, la información de la empresa y el rol de la persona encuestada dentro de la organización y sus características de principales. La tercera comprende las preguntas propias del estudio a fin de responder a objetivos de investigación y permitir la confrontación de las hipótesis.

Es trabajo de campo fue realizado entre los meses de octubre y diciembre del 2020. Se utilizó una plataforma digital de tipo Google Forms que fue respondida de forma independiente y confidencial por los encuestados. Se informó sobre el carácter confidencial del estudio y el uso académico de las conclusiones. Se trabajó con una muestra no probabilística

o incidental integrada por 10 casos, involucrando hombres y mujeres, de diferentes edades y roles.

La **Tabla 5.1** representa los atributos característicos de las empresas encuestadas.

Tabla 0.1 Composición de la muestra

Composi ción muestra Caso	1	2	3	4	5	6	7	8	9	10
Industria o sector	Seguros	Petrolera	Alimenticio	Tecnología	E-commerce	Bancaria	Salud	Marketing	Construcción	Automotor en servicios
Tipo de Actividad	Servicios	Gubernamental	Producción de bienes de consumo	Servicios	Servicios	Producción de bienes	Servicios	Producción de bienes	Producción de bienes de capital	servicios
Cantidad de empleados	250 - o más	250 - o más	250 - o más	250 - o más	250 - o más	250 - o más	250 - o más	250 - o más	250 - o más	250 - o más
Antigüedad de la empresa	5 a 10 años	Mayor a 10 años	Mayor a 10 años	2 a 5 años	Menor de 2 años	Mayor a 10 años	Mayor a 10 años	Mayor a 10 años	5 a 10 años	5 a 10 años
Años de Experiencia en Proyectos	5 a 10 años	Mayor a 10 años	Mayor a 10 años	Mayor a 10 años	5 a 10 años	Mayor a 10 años	5 a 10 años	Mayor a 10 años	Mayor a 10 años	Mayor a 10 años
Categoría	Grande	Grande	Grande	Grande	Grande	Grande	Grande	Grande	Grande	Grande
Cargo	Business Analyst	Project Manager	Functional Analyst	Functional Analyst	CEO	Project Manager	Functional Analyst	Functional Analyst	Project Manager	Sr. QA Engineer
Personas a Cargo	0	20	0	12	355	25	5	0	15	10
Área	Business Development Office	Project Management	Service Delivery	Tecnología	Strategy Management Office	Business Development	Service Delivery	Tecnología	Tecnología	Quality Assurance
Nivel de Educación	Universitarios	Postgrado	Universitarios	Universitarios	Universitarios	Postgrado	Postgrado	Universitarios	Universitarios	Universitarios

6.3.3 Principales conclusiones del trabajo de campo

Las preguntas realizadas a los encuestados fueron segmentadas de la siguiente forma:



Filtro y Caracterización: orientadas a permitir identificar las características de las empresas cuyas encuestas se respondieron y permitieron identificar el tipo de sector que pertenecen los encuestados, donde la gran mayoría pertenecen a medianas y grandes empresas.

- Nivel de educación
- Experiencia en proyectos informáticos
- Conocimiento de procesos sistémicos
- Técnicas y skills en relevamiento de requerimientos
- Metodologías más utilizadas
- Experiencia en resultados de proyectos exitosos

Se presentan en el Anexo 1. las respuestas obtenidas para cada pregunta.

6.4 Corroboración de Hipótesis

El estudio realizado ha brindado los siguientes resultados respecto a las hipótesis planteadas:

H1. La falta de un procedimiento sistémico de recopilación y evaluación de requisitos para el desarrollo de software puede llegar a introducir fallas de calidad, adecuación y agregar tiempos y costos al proyecto.

Del colectivo estudiado puede concluir que si bien es cierto la mayor parte de las empresas cuentan con procedimientos establecidos, no existe un verdadero pensamiento sistémico para el manejo de los requerimientos y evaluación de los requisitos en los proyectos de software.

La forma de especificar un sistema tiene una gran influencia en la calidad de la solución implementada finalmente en los proyectos. Es ampliamente aceptado que el costo de corregir-modificar un sistema después de su instalación, o incluso después de las primeras etapas de la fase de diseño, es mucho mayor que el costo de preparar una especificación de requerimientos inicial (Davis, 2003).

Como se puede evidenciar en la figura 1. la mayor parte de los encuestados un 44.4% coinciden que uno de los factores que afecta directamente al éxito del proyecto es el hecho de no establecer correctamente el alcance.

Figura 0.1 Causas de Fallas de los Proyectos



Fuente: Encuesta (2020)

Para corroborar lo analizado en las encuestas, se puede acceder al informe de la consultora The Standish Group, periódicamente publica el “CHAOS Report” 2015 , donde indica que las principales causas de fallas en los proyectos se encuentra que el 52% se atribuyen a deficiencias en el proceso de ingeniería de requisitos como son: una mala administración de los requisitos, trabajar con los proveedores inapropiados, poca participación de los usuarios, requisitos incompletos, incorrectos, que faltan e inclusión de requisitos no esenciales. Todo lo anterior genera un alto nivel de reproceso de actividades que deben realizarse para corregir dichas fallas. Una valoración de los costos por reproceso muestra que estos pueden llegar a representar el 30% del total del costo del proyecto.

Por otro parte CHAOS indica que encuestó a 9.236 proyectos, donde queda demostrado que sólo el 29% de los mismos finalizan con éxito. Entendiendo por proyectos exitosos, aquellos que terminan en el tiempo estimado, presupuesto establecido y funcionalidad requerida. En la siguiente figura 2. se evidencia el alto número de proyectos que fallan al tratar de alcanzar sus objetivos y queda demostrado como la etapa de requisitos influye en un alto porcentaje de estos fracasos.

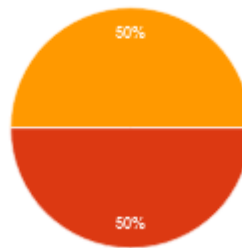
Figura 0.2 Proyectos Exitosos y Fracasados



Fuente: Informe Chaos (2015)

En el caso de la encuesta los entrevistados indicaron que solo un 50% de los proyectos se pueden considerar éxitos, como se muestra en la Figura 3.

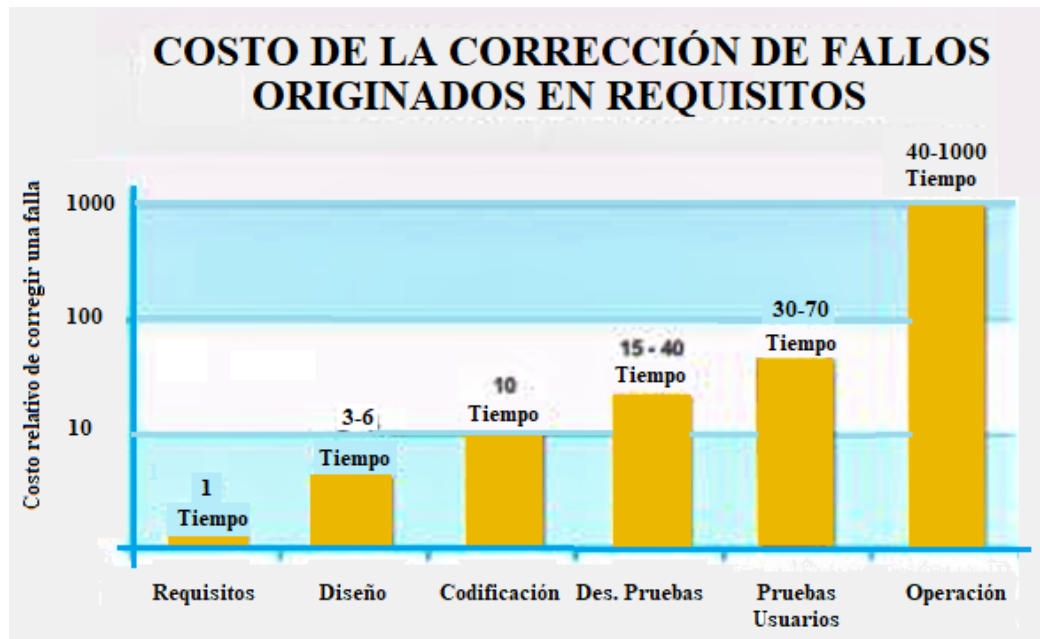
Figura 0.3 Proyectos Exitosos



Fuente: Encuesta (2020)

Davis (2003) menciona que los costos pueden llegar a aumentar dependiendo de la etapa del proyecto en la cual sean identificados, como se evidencia en la figura 4. *Costo de la corrección de fallos originados en requisitos*. La detección temprana de estas fallas ayuda a disminuir considerablemente el impacto en todos los aspectos del proyecto, incluido el costo de corrección. En contraposición, la detección tardía de una de estas fallas puede llegar a aumentar los costos de corrección hasta 1000 veces que sí se hubiese hecho al inicio del proyecto.

Figura 0.4 Costo de la corrección de fallos originados en requisitos



Fuente: Davis (2003)

Por lo tanto, este es uno de los factores que afectan directamente al éxito o fracaso de un proyecto, ya que puede generar que el mismo avance con su desarrollo e implementación sin saber cuáles son las necesidades básicas dentro de la organización perdiendo de vista el objetivo general de la misma, y cuando se hace mal, las consecuencias pueden ser graves en términos de calidad, tiempo de entrega y costo.

H2. La falta de herramientas y metodologías de gestión de proyectos en el desarrollo de software puede llegar a introducir fallas de calidad, adecuación y agregar tiempos y costos al proyecto.

De las respuestas obtenidas por el colectivo estudiado en torno a la falta de herramientas y prácticas de gestión de proyectos en el desarrollo de software, se puede identificar la presencia de cierto conocimiento de herramientas o técnicas por parte de los encuestados al parecer las más mencionadas son:

- ✓ Entrevista
- ✓ Lluvia de Ideas

✓ Cuestionarios.

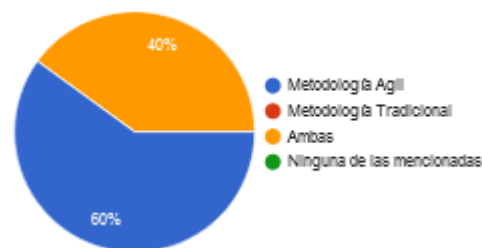
Sin embargo, las organizaciones no logran demostrar mediante sus procesos de gestión que se centren en la validación y verificación de los requisitos mediante el uso de alguna de las técnicas mencionada en este trabajo.

En un estudio realizado por Smith (2014) muestra que las organizaciones que utilizan un proceso formal para la validación de los requisitos de proyectos y en general de procesos de gestión de proyectos, los resultados obtenidos son significativamente mejor.

Fallar en obtener los requerimientos, del sistema final, desde un principio es la principal razón del aumento de costos y tiempos de entrega como se puede evidenciar en la hipótesis 1 de este trabajo. La especificación de los requerimientos de información, no es solo la actividad más importante dentro del proceso del desarrollo de sistemas sino la más difícil (Vessey & Conger, 1994).

También se recolectó información de los encuestados en cuanto a la metodología utilizada, la mayor parte de ellos se inclinan por el uso de metodologías ágiles donde al parecer han tenido mejores resultados, aunque existe un número representativo dentro de los entrevistados que usa la combinación de las metodologías Ágiles y tradicionales como se puede ver en la figura 5.

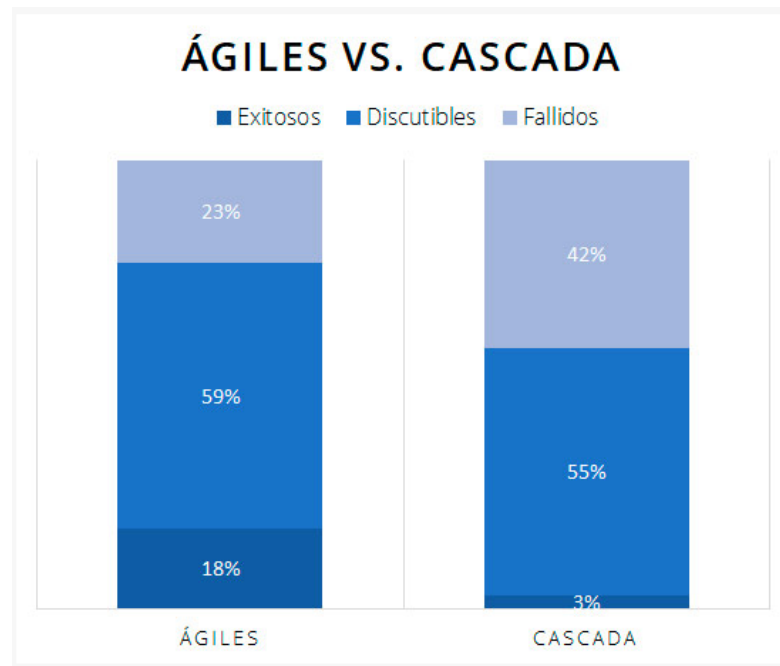
Figura 0.5 Metodologías en los Proyectos



Fuente: Encuesta (2020)

Las estadísticas del informe CHAOS, donde ilustran con datos el éxito o fracaso de los proyectos según la metodología utilizada, si se comparan ambas metodologías en proyectos grandes, los cuales desde luego son más complejos por el tiempo de duración que es más de 2 años, se puede ver que según la metodología elegida tenían mayor o menor probabilidad de éxito.

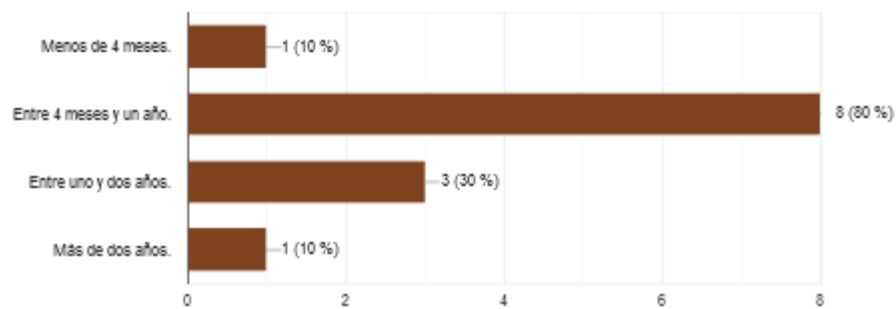
Figura 0.6 Metodología Ágil Vs Tradicional



Fuente: Rodelgo, A. (2019)

En la Figura 6, se puede ver que solo el 18% de los proyectos grandes son exitosos. En el caso de nuestra encuesta figura 7. Como se puede evidenciar el 80% de los proyectos son pequeños y por ende existe menos complejidad en cuanto las variables del triángulo de hierro, y este puede ser uno de los factores por el cual el porcentaje alto de proyectos exitosos es un 60%.

Figura 0.7 Tamaño de los Proyectos



Fuente: Encuesta (2020)

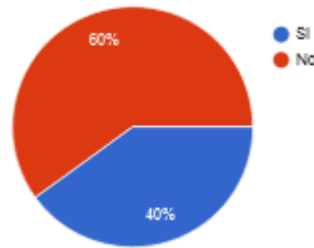
En esta lucha entre la gestión tradicional y la gestión ágil de proyectos, parece que en los últimos años va ganando camino la metodología ágil frente a la tradicional. Cuando se involucra al equipo completo y a los clientes en procesos importantes, el resultado final para estos últimos y la experiencia de trabajo para todos los involucrados es mucho más gratificante que el enfoque de estilo tradicional.

También podría estar asociado al aspecto cultural de la empresa ya que las personas siempre han utilizado metodologías tradicionales dentro de la organización y no quieren cambiar sus procesos metodológicos. Igualmente podemos concluir que las metodologías ágiles están ganando más fuerza sobre el desarrollo de proyectos lo que pensamos que hace falta es mayor capacitación a los gerentes de proyectos de cómo aplicar o utilizar adecuadamente metodologías ágiles dando confianza y de esta forma mitigar los casos de fracaso en proyectos de TI.

H3. La utilización de buenas prácticas de ingeniería de requerimientos y mapas mentales de apoyo al proceso de análisis y diseño de software puede contribuir a mejorar la calidad, adecuación y reducir tiempos y costos del proyecto.

Considerando el colectivo estudiado se puede evidenciar, como se destaca en la figura 8, que el 60% de los entrevistados no consideran el uso de las buenas prácticas como parte de los procesos o estándares de relevamiento del requerimiento en los proyectos.

Figura 0.8 Utilización de Buenas Practicas

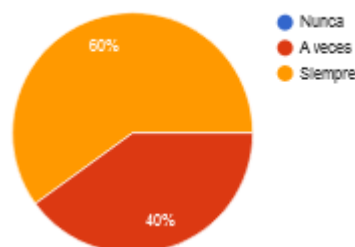


Fuente: Encuesta (2020)

Sin embargo, los estudios reflejan que las malas prácticas de requerimientos son la causa principal del fracaso de los proyectos, la prioridad de las organizaciones no debe centrarse únicamente en la calidad del producto terminado. Debe medirse la calidad a lo largo de todos los procesos que se llevan a cabo en una empresa más que todo en las que se dedican exclusivamente al desarrollo de software como es el caso de la mayoría de los encuestados que se dedican a la gestión de proyectos. Enfocarse en cada uno de los procesos implica aplicar las mejores prácticas en gestión e ingeniería de software para alcanzar una evolución hacia la cultura excelencia y por ende a la calidad del software

En el caso de los mapas mentales como apoyo a una mejor estrategia para el levantamiento de los requerimientos, la mayor parte de los encuestados, como se puede evidenciar en la figura 9. son conscientes que los mapas mentales o representación gráfica ayudan a tener una mejor visión del problema.

Figura 0.9 Utilización de Mapas Mentales



Fuente: Fuente: Encuesta (2020)



No obstante, al no ser una mayoría absoluta indicaría que aún existen falencias para tener un pensamiento sistémico como lo menciona Ontorio (2003):

Pensamiento sistémico trabaja fundamentalmente con los llamados "modelos/mapas mentales" los cuales son una representación gráfica de un proceso integral y global del aprendizaje que facilita la unificación, diversificación e integración de conceptos o pensamientos para analizarlos y sintetizarlos en una estructura creciente y organizada, elaborada con imágenes, colores, palabras y símbolos (p.40).

El uso de los mapas mentales permite lograr el objetivo propuesto y resultó ser una herramienta estratégica eficaz para identificar los puntos críticos de los requisitos durante el proceso de desarrollo del producto, ya que, en general, un requisito bien diseñado reduce el costo de desarrollo, evitando retrasos, asegurando la satisfacción de los grupos de interés y por ende mejorando la calidad del proyecto.

H4. El rasgo de fuerte presencia de pensamiento grupal conspira contra la obtención, en el análisis y diseño de software, de buenos resultados a nivel de calidad, adecuación y tiempos y costos del proyecto.

En esta hipótesis se focalizó la encuesta en identificar los factores que generan GroupThink en los equipos de trabajo, basándose en su taxonomía de la ausencia de este rasgo:

- Se definieron claramente los objetivos del proyecto.
- El líder del proyecto cuenta con el perfil profesional adecuado
- La alta dirección brinda el apoyo al equipo del proyecto
- Los integrantes del equipo fueron seleccionados apropiadamente
- Fueron asignados suficientes recursos materiales y económicos para el proyecto.
- Se establecieron buenos canales de comunicación
- Hay crítica en la gestión, la visión no es endogámica.



Se logró identificar tras las respuestas que la mayoría de estos puntos se encuentran presentes en la gestión de los proyectos. Existe presencia de GroupThink, —Ello podría deberse a que en su mayoría los encuestados trabajan con una dinámica de equipos ágiles. Los equipos auto organizados ágiles más comunes son unidades cohesivas que generalmente están físicamente aisladas de la corriente principal de la organización más grande. Juntos aprenden procesos ágiles, aprenden a trabajar en su grupo cerrado y trabajan para lograr sus objetivos de sprint al mismo tiempo.

Janis (1971) describió como la tendencia de algunos grupos a tratar de minimizar el conflicto y llegar a un consenso sin probar, analizar y evaluar suficientemente sus ideas. La investigación de Janis sugirió que el desarrollo de las normas de un grupo tiende a poner límites al pensamiento independiente y creativo de los miembros del grupo. Como resultado, el análisis de un grupo puede estar sesgado y conducir a malas decisiones.

Es fácil que la colaboración se convierta en un pensamiento grupal en equipos muy unidos, porque el deseo de consenso y toma de decisiones rápidos puede ser fuerte. Sin embargo, tanto los miembros del equipo como los gerentes pueden reconocer los síntomas y hacer ajustes para guiar a los equipos de regreso al alto desempeño.

Los equipos ágiles pueden ser muy eficaces en el desarrollo y la entrega de software, pero el GroupThink es un peligro que puede obstaculizar su capacidad para ser creativos en su trabajo. Los gerentes y equipos deben ser conscientes del GroupThink y deben administrarlo individualmente, analizando los factores intrínsecos y extrínsecos que afectan directamente al equipo.

H4. La falta de participación del cliente en el proyecto y el uso de herramientas obsoletas que no se adaptan a los cambios disruptivos de la transformación digital que se está atravesando actualmente, puede acarrear fallas en el triángulo de hierro.

En base al análisis de los encuestados se puede determinar, que a pesar de los cambios disruptivos que estamos viviendo, aun en este momento existen ciertas empresas que les costara adaptarse transformación digital y la innovación del modelo de negocio que se deberían empezar a plantear.

Kolsky(2018) en su encuesta anual enfatizando fuertemente la importancia que se debe dar a la experiencia del cliente; estos son los aspectos más destacados:



Las investigaciones muestran que el 55% de los consumidores están dispuestos a pagar más por una buena experiencia garantizada. Kolsky señala la palabra "garantizado", señalando que los clientes ya no están satisfechos con solo que se les prometa una buena experiencia.

El 84% de los consumidores se siente frustrado cuando el agente no tiene información.

La frustración del cliente conduce a lo siguiente: el 13% le dice a 15 o más personas si no están contentas. Por el contrario, el 72% de los consumidores compartirán una experiencia positiva con 6 o más personas. El 67% de los consumidores consideran las malas experiencias como motivo de abandono.

Kolsky investigó y encuestó a líderes empresariales para encontrar otras tendencias de importancia crítica con respecto a la experiencia del cliente, que incluyen:

El 50% de la experiencia del cliente se produce de forma natural, cada 5 años.

Solo 1 de cada 26 clientes descontentos se queja. El resto se agita. Una lección aquí es que las empresas no deben ver la ausencia de comentarios como un signo de satisfacción. El verdadero enemigo es la indiferencia.

El 91% de los clientes descontentos que no se quejan simplemente se van.

Es entre 6 y 7 veces más caro para las empresas atraer nuevos clientes que mantener a los clientes existentes. El 65% de las empresas pueden realizar ventas adicionales o cruzadas con éxito a los clientes existentes. Solo el 12% de las empresas pueden realizar ventas adicionales o cruzadas con éxito a nuevos clientes.

El consejo de Kolsky a las empresas es que se concentren en su trabajo. El trabajo de la empresa se trata menos de saber lo que quiere el cliente y más de construir la infraestructura para permitir que los clientes elijan, personalicen y optimicen libremente su experiencia en cada interacción en función de sus necesidades. Las empresas deben utilizar las tecnologías y herramientas adecuadas para construir una infraestructura de autoaprendizaje, diseñada para una mejora continúa basada en los comentarios medidos de los clientes. Al final, gana la conveniencia. La infraestructura debe ser adaptable, flexible, escalable e inteligente.

Los resultados de la investigación y la encuesta de Kolsky con respecto a las redes sociales, los dispositivos móviles, las comunidades, el mapeo de viajes, el compromiso y el servicio al cliente de gestión del conocimiento también deberían ser una revelación para los ejecutivos de negocios.



Por estas razones es básico que desde el relevamiento de los requerimientos tomemos en consideración y nos preparemos para estos cambios e incluso de alguna manera podamos asesorar al cliente que sería lo más adecuado para el usuario final y de esa manera conseguir proyectos exitosos.

6.5 Conclusiones

Como se planteó en el objetivo de este trabajo, abordar la calidad del software desde la perspectiva del análisis de los requerimientos, analizando las metodologías, herramientas y buenas prácticas de software, brindando lineamientos para poder ejercer un control de la gestión del levantamiento de requerimientos mediante procesos sistémicos para evitar factores críticos de fracaso, afectan directamente a las variables del triángulo de hierro: alcance, tiempo y costo.

Al obtener las respuestas de los encuestados se pudo concluir en cuanto al tratamiento uniforme de los requisitos, en las empresas de desarrollo los analistas usan diferentes técnicas para identificar, separar y clasificar los requisitos. Los criterios usados para la aplicación de las técnicas son variados y dependen en gran medida de la experiencia del analista. Muchas veces, los requisitos no funcionales o los atributos de calidad adquieren importancia sólo en el momento de modelar la arquitectura, limitando el tratamiento uniforme de los requisitos.

Para un proyecto de software es muy importante contar con buenas prácticas de análisis de requerimientos y que las mismas sean flexibles de acuerdo con las condiciones de un proyecto o aplicación específico. En el estudio se detectó que, dada la diversidad de escenarios para originar los requisitos, no todas las empresas encuestadas usan las más adecuadas o definitivamente no la usan. En base a informes presentados en este trabajo se puede evidenciar que el uso de las buenas prácticas genera una mejor comprensión de los requerimientos, que derivaban en uno de los factores de éxito de proyecto.

En cuanto a los mapas mentales la mayor parte de los encuestados son conscientes de su gran aporte para el relevamiento, sin embargo, no usan técnicas específicas, si no que las mismas están supeditadas al *expertise* del analista funcional que este asignado a determinado proyecto.



Para las organizaciones es importante tener una forma flexible de integrar nuevos marcos metodológicos. El aprendizaje y adopción de nuevos paradigmas se puede convertir en un riesgo, pero en realidad la experiencia de los analistas en el uso de las técnicas se convierte en el factor clave de éxito. Sin embargo, uno de los grandes problemas que ahora enfrentan los proyectos de Software en la presencia del Groupthink el cual se puede evidenciar en las encuestas realizadas dado que gran parte de los encuestados pertenecen a equipos ágiles los cuales por lo general son competentes en el desarrollo y la entrega, pero GroupThink puede obstaculizar su capacidad para ser creativos en su trabajo. Los gerentes y equipos deben ser conscientes del pensamiento grupal y deben administrarlo individualmente, dentro del equipo y externamente, si es necesario.

Cada proyecto tiene el objetivo de ofrecer soluciones a los problemas de los clientes, generar valor tanto para el cliente como el usuario final y eso se logra mejorando la participación y comunicación de los interesados, y gestionando cada uno de los requerimientos de la manera más óptima ya sea usando herramientas innovadoras o adaptándonos de la mejor manera al proyecto que nos presente.

Presman (2010) indicaba que no existe un único enfoque para mejorar el mal software. Sin embargo mediante la combinación de métodos completos para todas las fases del desarrollo del software, mejores herramientas para automatizar estos métodos, arquetipos más potentes para la implementación del software, establecer procesos sistémicos en las organizaciones, mejores técnicas para garantía de calidad del software y una filosofía predominante para coordinación, control y gestión, podemos conseguir un mejor análisis de los requerimientos y con ello mejorar la calidad del software.

6.6 Futuras líneas de Investigación

Como trabajos futuros en el relevamiento correcto de los requerimientos, para una mejor gestión de proyectos se plantea lo siguiente:

1. Dado que el enfoque común que se identificó en la encuesta de este trabajo es la metodología Ágil para la mayor parte de los proyectos, se considera que se debería hacer una



investigación tanto de técnicas, herramientas o procesos sistémicos que logren ayudar en esta problemática:

- El proceso no es visible. Los administradores necesitan entregas regulares para medir el avance. Si los sistemas se desarrollan rápidamente, resulta poco efectivo en términos de costos producir documentos que reflejen cada versión del sistema.
 - La estructura del sistema tiende a degradarse conforme se tienen nuevos incrementos. A menos que se gaste tiempo y dinero en la refactorización para mejorar el software, el cambio regular tiende a corromper su estructura. La incorporación de más cambios de software se vuelve cada vez más difícil y costosa.
2. Dado que actualmente gran parte de los analistas funcionales encuestados tienen falencias para encarar los puntos citados, resultaría interesante enfocarlo únicamente en la metodología ágil indicando vulnerabilidad intrínseca de los equipos de desarrollo de software ágiles, y quizás analizando algún grupo determinado de trabajo.
 3. En cuanto a las Mapas Mentales se logró identificar que del universo de encuestados todos son conscientes que la ayuda visual es una clave para realizar un mejor relevamiento de los requerimientos, por ello se considera posible ampliar este tema y presentar diferentes opciones y herramientas, que se adaptativas a los cambios acelerados de este mundo VICA.



Bibliografía

- Abdoulaye, A. (2003). Conceptualisation et Dissemination des “Bonnes Pratiques”», en
Éducation: Essai d’une Approche Internationale à Partir d’Enseignements Tirés d’un
Projet. Obtenido de
http://portal.unesco.org/education/en/file_download.php/f1685fde2633dd9b3b20fd828d6bfa92abdoulaye.pdf
- Akella, P., & Rao, K. (2011). “Effective Independent Quality Assessment using IV&V”,
International Journal of Computer Science & Information Technology, 3(3), 188-198.
- Amaya, V. (2013). *SlideShare: ¿Qué hacer para dominar el arte del levantamiento de requerimientos?* Obtenido de Available at:
<http://es.slideshare.net/RevistaSG/webinar-levantamiento-reqsv1>
- Ambler, S. (2009). *Agile Modeling: Effective Practices for Extreme Programming and the Unified Process*. New York: John Wiley & Sons Inc.
- Ameca Carreón, J. (7 de Octubre de 2014). *Visión sistémica según Peter Senge*. Obtenido de
<https://www.gestiopolis.com/vision-sistemica-segun-peter-senge/>
- Armijo, M. (2004). Buenas Practicas de Gestión Pública en America Latina. *Ponencia Presentada en el IX Congreso Internacional del CLAD sobre la Reforma del Estado y de la Administración Pública*. Madrid.
- Aston, B. (2021). *10 Best Requirements Management Tools & Software Of 2021*. Obtenido de <https://thedigitalprojectmanager.com>:
<https://thedigitalprojectmanager.com/requirements-management-tools/>
- BABOK. (2015). *A Guide to the Business Analysis Body of Knowledge BABOK® Guide*. (I. I. Analysis, Ed.) Toronto, Canada.
- Balaguera Reina, D. A. (2017). *Identificación y análisis de los factores de éxito y fracaso en la gerencia de proyectos*. Bogota: Escuela colombiana de ingeniería Julio Garavito.
- Bounds, G. (1998). *The last word on project management* (30(11) ed.). IIE Solutions.
- Brown, M. (2014). *GROUPTHINK IN SOFTWARE ENGINEERING*.
- Buzan, T., & Buzan , B. (1995). *Libro de los Mapas Mentales*. España: Urano.



- Caballero, A. (9 de septiembre de 2015). *Pensamiento grupal* . Obtenido de Psicología con alfonso: <http://psicologiaconalfonso.com/pensamiento-grupal/>
- Callaway, M. J. (1984). GroupThink: Effects of Cohesive-ness y procedimientos de resolución de problemas en la toma de decisiones en grupo. *comportamiento social*. 157-164.
- Canchari. (2018). La validacion de requisitos de software como base del exito de proyectos de sistemas informaticos desarrollados e Implementados en la Comisión Nacional DEVIDA. Huancayo , Perú: Universidad Nacional del Centro del Peru.
- Cartwright, D. Z. (1968). *Dinámica de grupo: investigación y teoría*, Haprer y Row.
- Carvajal, S., M, M., & Vargas, G. (2008). *La Trazabilidad en el proceso de requerimientos de software*. Obtenido de <http://www.iiis.org/CDs2008/CD2008CSC/CISCI2008/PapersPdf/C601UZ.pdf>
- Cerpa, N., & Verner, J. (2009). "Why Did Your Project Fail?". *Communications of the ACM*, 52(12), 130-134.
- Chamoun, Y. (2002). *Administración profesional de proyectos: la guía*. Mc Graw Hill.
- Davis, A., & Zowghi, D. (2006). *Good requirements practices are neither necessary nor sufficient* (Vol. 11). doi:10.1007/s00766-004-0206-4
- Delgado , E., & Erly. (2008). "Metodologías de desarrollo de software. ¿Cuál es el camino?". *Revista de Arquitectura e Ingeniería*, 10-34.
- DeMarco, T. y. (1987). *Peopleware: Proyectos y equipos productivos*, Casa Dorest Publishing Co. Nueva York.
- DeMarco, T., & Lister, T. (1987). *Peopleware: Productive Projects and Teams*. New York.
- Driskell, J. E., & Salas, E. (s.f.). Group decision making under stress. *Journal of Applied*. 76, 473-478.
- Fairley, R. E., & Bourque, P. (2014). *SWEBOK: guide to the software engineering body of knowledge*. IEEE Computer Society.
- Gacitua, R. (2003). *Métodos de desarrollo de software: El desafío pendiente de la estandarización. Software Development Methodologies: A Duel Pending for Standardization*. Theoria.



- Gallivan, M., & Srite, M. (s.f.). *Information Technology and Culture: Identifying Fragmentary and Holistic Perspectives of Culture* (Vol. 15). Information and Organization.
- García, R. (2018). *Estudio comparativo entre las metodologías ágiles y las metodologías tradicionales para la gestión de proyectos de software*. Universidad de Oviedo.
- Garner. (7 de Marzo de 2016). *For Better Requirements, Gartner Says You Need The Right Tools*. Obtenido de <https://www.blueprintsys.com>:
<https://www.blueprintsys.com/blog/for-better-requirements-gartner-says-you-need-the-right-tools>
- Ghosh, B., & Daugherty, P. (2015). *Change BUSINESS FUTURES 2021*. Obtenido de <https://www.accenture.com>:
https://www.accenture.com/t00010101T000000__w__/co-es/_acnmedia/Accenture/Conversion-Assets/DotCom/Documents/Local/es-la/PDF2/Accenture-The-Future-Of-Applications-Spanish.pdf
- Goldstein, H. (2005). *“Who Killed the Virtual Case File?”* IEEE Spectrum.
- Goleman, D. (1998). *Inteligencia Emocional*. Barcelona, España: Kairos.
- Gómez Fuentes, M. (2011). *NOTAS DEL CURSO: ANALISIS DE REQUERIMIENTOS*. México: UNIVERSIDAD AUTONOMA METROPOLITANA.
- González Dematine, J. (2021). *Equipos interdisciplinarios: la clave para entender a los clientes*. Human Capital Manager en Nubiral. Obtenido de <https://www.infotechnology.com/innovacion/equipos-interdisciplinarios-la-clave-para-entender-a-los-clientes/> Por Juan Andrés González Dematine, Human Capital Manager en Nubiral. 2
- Group, S. (2015). *Chaos Report*. Standish group.
- Group, T. S. (1994). *Chaos Report 1994*. Obtenido de https://www.standishgroup.com/sample_research_files/chaos_report_1994.pdf
- Group, T. S. (2014). *Chaos Report 2014*. Obtenido de https://www.standishgroup.com/sample_research_files/CHAOSReport2014.pdf
- Hiranabe, K. (2006). Agile Modeling with Ming Map and UML. *Better Software Magazine*.
- Hofmann, H. F., & Lehner, F. (2001). *Requeriments Engineering as a success factor in software proyects*. IEEE Software.



- Hughes, M. W. (1986). *Why projects fail: The effects of ignoring the obvious* (Vol. 18). Industrial Engineering.
- Hughes, R. G. (1993). *Tácticas de poder, influencia e influencia. En Liderazgo: mejora de las lecciones de la experiencia*. Richard D. Irwin, Inc.
- Humphrey, W. (1997). *Introduction to the Personal Software Process*. Massachusetts: Addison Wesley Logman.
- I, S., & P, S. (1997). *Requirements Engineering: A Good Practice Guide*. John Wiley & Sons.
- Ian, A., & Ljerka, B. D. (2009). *Discovering requirements: how to specify products and services*. NJ: Wiley: England Hoboken.
- IEEE, S. C. (1990). *IEEE Standard Glossary of Software Engineering Terminology*. New York: IEEE.
- Institute, P. M. (2014). *Project Management Institute Anual 2014 report* (BRA-120-2015 (6-15) ed.).
- INTECO. (2014). *Buenas prácticas en el área de informática. Instituto Nacional de Tecnologías de la Comunicación (INTECO)*. España. Obtenido de www.inteco.com
- ISO/IEC/IEEE. (2011). *Systems and software engineering -- Life cycle processes -- Requirements engineering*. 29148. Obtenido de <http://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?punumber=6146377>
- Janis, I. (1972). *Víctimas del pensamiento grupal*. Boston: Houghton Mifflin.
- Janis, I. (1982). *Counteracting the adverse effects of concurrence-seeking in policy-planning groups: Theory and Research perspectives*. New York: Group Decision Making.
- JANIS, I., & Mann, L. (1977). *Decision Making: A Psychological Analysis of*. New York: The Free Press.
- Jannis , I. (1942). *Victims of Groupthink: A Psychological Study of Foreign-Policy*. Boston. doi:ISBN 0395140447
- Jorgensen, M., & Grimstad, S. (2012). *Software Development Estimation Biases: The Role of Interdependence*. *EE Transactions On Software Engineering*, 38(3), 677-693.
- Jr, F. B. (1995). *The Mythical Man-Month: Essays on Software Engineering* (20th anniversary ed.). Addison-Wesley.
- Kayser, T. (1994). *Poder de equipo*. BurrRidge: Irwin.



- Kemerer, C., & Paulk, M. (2009). "The Impact of Design and Code Reviews on Software Quality: An Empirical Study Based on PSP Data", *IEEE Transactions on Software Engineering*. 25(4), 534-550.
- Kerzner. (2013). *Project Management: A systems approach to Planning, Scheduling and Controlling* (11th ed.). New Jersey: John Wiley & Sons.
- Khaled , E. E., & Nazim, H. M. (2005). A field Study of Requeriments Engineering Practices in Information Systems Development. England.
- Klastorin. (2007). *Administración de Proyectos*. México: Alfaomega.
- Kolsky , E. (06 de Dec de 2017). *50 Important Customer Experience Stats for Business Leaders*. Obtenido de https://www.huffpost.com/entry/50-important-customer-exp_b_8295772
- Kontio, J. e. (2004). "Using Focus Group Method in Software Engineering: Obtaining Practitioner and User Experience",. *In Proceedings of the International Symposium on Empirical Software Engineering*, (págs. 271-280). Finland.
- Kontio, J. e. (2004). Uso del método de grupos focales en ingeniería de software: obtención Experiencia de usuario y practicante". (págs. 271-280). Actas del Simposio Internacional sobre Ingeniería de Software Empírica.
- Kulbytè, T. (4 de mayo de 2021). *37 Customer Experience Statistics You Need to Know for 2021*. Obtenido de <https://www.superoffice.com/blog/customer-experience-statistics/#respond>
- Leana, C. (1985). Una prueba parcial del modelo de pensamiento Grupal de Janis: Efectos de cohesión del grupo y el comportamiento del lider en la toma de decisiones defectuosa. *J. Gestionar*, 11, (15-18).
- Lederer, A., & Prasad, J. (1992). Nine Management Guidelines for Better Cost Estimating. *Communications of the ACM*, 35(2), 51-59.
- Lehtola, L, L., Kauppinen, M., & Vahaniity, J. (2009). Linking Business and Requirements Engineering: Is Solution Planning a Missing Activity in Software Product Companies? *Requirements Engineering*. 14: 113-128.
- López, C. A., & Cuello de Avila, S. (2016). Efectividad metodológica para el levantamiento de requerimientos de aplicaciones web. *Revista Ontare*, 4(2), 71-98. doi:<https://doi.org/10.21158/23823399.v4.n2.2016.1628>



- M, C. (1997). Risk Management May Not Be for Everyone . *IEEE Software*, 14, No. 3, 21-24.
- Masooma Yousuf, M. A. (2015.). *International Journal of Advanced Research in Computer Science and Software Engineering. A Systematic Approach for Requirements Elicitation Techniques* (Vol. 4). A Review.
- McConnel, S. (1996). *Desarrollo rápido: Taming Wild Software Schedules*, Redmond. Washington: Microsoft Press.
- Miller, R., & Lessard, D. R. (2008). *Decision-Making on Mega-Projects: Cost-Benefit Analysis, Planning and Innovation. Evolving strategy: risk management and the shaping of mega-projects.* . Massachusetts: Edward Elgar Publishing Inc.
- Miranda, S. (1994). Avoidance of Groupthink: Meeting management using group support system . *Small Group Research*, 25, 105-136.
- Miranda, S. (1994). Evitación del pensamiento de grupo: gestión de reuniones. 105 - 136.
- Muganda, N. D. (2014). Visión de grupo haciendo deficiencia en el proceso de ingeniería de requisitos: Hacia un modelo de Crowdsourcing. Obtenido de http://papers.ssrn.com/sol3/papers.cfm?abstract_id=2132040
- Mulder, P. (2013). *Modelo de cultura organizacional de Edgar Schein* . Obtenido de <https://www.toolshero.com/leadership/organizational-culture-model-schein/>
- Nicholas , M. (2021). <https://blog.mindmanager.com/blog/2021/02/requirements-gathering-tools/>.
- Nisbett, R. (2003). *The Geography of Thought: How Asians and Westerners Think Differently*. New York: Free Press.
- Ochara, N. M. (2012). Groupthink Decision Making Deficiency in the Requirements Engineering Process: Towards a Crowdsourcing Model. Obtenido de SSRN: <https://ssrn.com/abstract=2132040> or <http://dx.doi.org/10.2139/ssrn.2132040>
- Olmos, A. (11 de Junio de 2015). *El Pensamiento Sistémico en la industria del software*. Obtenido de <http://dwsoftware.mx/blog/el-pensamiento-sistemico-en-la-industria-del-software/>
- Osorio, N. A., & Castro , H. G. (2011). Management of Quality in Software Development. *Revista de investigacion de Sistemas e Informatica*, 8(1), 65-69.



- P w c. (2018). *1pwc.com/future-of-cxExperience is everything: Here's how to get it right*.
Obtenido de <https://www.pwc.com>: <https://www.pwc.com/us/en/advisory-services/publications/consumer-intelligence-series/pwc-consumer-intelligence-series-customer-experience.pdf>
- Palacio, J. (2006). *Gestión de proyectos ágil: conceptos básicos*. .
- Parker, K. (2015). *Mejores prácticas de ingeniería de requisitos*. Micro Focus®.
- Perez , M. A., Griman , A. C., & Mendoza, L. E. (2004). *Prototipo de Modelo sistémico de Calidad (MOSCA) del software* (Vol. 8). Caracas, Venezuela.
- Pérez-Virgen, H. L.-M.-A. (2013). Levantamiento De Requerimientos Basados En El Conocimiento Del Proceso. *Revista Científica*, (16) , 42–51.
- Pfleeger, S. L. (s.f.). *Ingeniería del Software. Teoría y Práctica*. 2007.
- Pinto, J., & Slevin, D. (1987). *Critical factors in successful project implementation*, *IEEE Transactions on Engineering Management* (Vol. 34(1)).
- Pleeger, S. L. (2002). *Ingeniería de software. Teoría y práctica* (1ª ed.). Buenos Aires: Pearson Education.
- Pressman. (2010). *INGENIERÍA DEL SOFTWARE Un enfoque Práctico*. (Septima ed.). McGRAW-HILL.
- Pressman, R. (2006). *Ingeniería del Software: Un Enfoque Práctico*. McGraw-Hill.
- Quinn, R. F. (1990). *Convertirse en un gerente maestro: un marco de competencias*. Nueva York: John Wiley & Sons.
- Rosnay, J. D. (1975). *Le macroscope: Vers une vision globale* . París: Seuil.
- Ruhe G, E. A. (2002). Quantitative Win Win a new method for decision support in requirements negotiation. *In proceedings of the 14th International Conference on Software Engineering and Knowledge Engineering (SEKE'02)*, 159–166.
- Ruiz Arias, M. C. (2015). *Determinación del éxito del proyecto*. Valencia, España: Estudio de caso práctico.
- Sáez, F. (2019). *FacileThings*. Obtenido de <https://facilethings.com/blog/es/systems-thinking>
- Salinger, S. e. (2008). Metodología de desarrollo de un esquema de codificación utilizando Teoría para el análisis cualitativo de la programación por pares. *Una revista interdisciplinaria sobre seres humanos en entornos de TIC*, 4(1), 9-25.



- Sanz Esteban, A. (2012). *Marco metodológico y tecnológico para la mejora de las actividades de verificación y validación de productos software*. Universidad Carlos III de Madrid. Obtenido de <https://e-archivo.uc3m.es/handle/10016/15464>
- Sawner, P., SOMMERVILLE, I., & Viller, S. (1997). *Requirements process improvement through the phased introduction of good practice*. *Software Process. Improvement and Practice*.
- Schein, E. (1984). Llegando a una nueva conciencia de la cultura organizacional . Revisión de la gestión de Sloan.
- Schein, E. (2010). *Liderazgo y cultura organizacional* . . Jossey-Bass.
- Schiano, W., & Weiss, J. (2006). “Y2K all over again: How groupthink permeates IS and compromises security. *Business Horizons*, 49, 115-125.
- Senge, P. (1990). *La Quinta Disciplina*. Editorial Granica.
- Sperry, R. W. (1981). *Some effects of disconnecting the cerebral hemispheres*. *Nobel Lecture. Les Prix Noble*. Stockholm: Almquist Wiksel.
- Staw, B., & Ross, J. (1987). *Knowing when to pull the plug*. *Harvard Business Review* (Vol. 65(2)).
- Std.830-1998, I. (2008). *Especificacion de Requisitos segun el estandar*.
- Sueter, K. (2018). <https://thedigitalprojectmanager.com/requirements-gathering-guide/>.
- Taylor, S. J., & Bogdan, R. (1984). *Introducción a los métodos cualitativos*. Barcelona: Ediciones Paidós.
- Terlato , A. (Abril de 2019). *MODELOS DE GOBIERNO DE TIPO HORIZONTAL, UNA PERSPECTIVA FRENTEA LOS SISTEMAS JERÁRQUICOS CLÁSICOS*. (J. M. Streb, Ed.) Buenos Aires, Argentina: Valeria Dowding. Obtenido de *MODELOS DE GOBIERNO DE TIPO HORIZONTAL, UNA PERSPECTIVA FRENTEA LOS SISTEMAS JERÁRQUICOS CLÁSICOS*: <https://ucema.edu.ar/publicaciones/download/documentos/692.pdf>
- Terlato, A. (2014). *Motivacion y Productividad en Empresas Argentinas. (Tesis Doctorado)*. UCEMA, Buenos Aires, Argentina.
- Terstine , M. (2015). El progreso de la Investigacion de Ingenieria de Requisitos . *Revista Antioqueña de las Ciencias Computacionales y la Ingenieria de Software*, 5(1), 18-24.



- Turner, M. A. (1992). Amenaza, cohesión y eficacia de grupo: prueba de una perspectiva de mantenimiento de la identidad social en el pensamiento de grupo. 781 - 796.
- Van Genuchten, M. (1991). "Why is Software Late? An Empirical Study of Reasons for Delay in Software Development". *IEEE Transactions on Software Engineering*, 17(6), 582-590.
- Wade. (2021). *The Importance of Customer Involvement in Software Projects*. Obtenido de Solidetech: <https://blog.soliditech.com/blog/the-importance-of-customer-involvement-in-software-projects>
- Wiegers, K. (11 de September de 2012). <https://www.jamasoftware.com/>. Obtenido de <https://www.jamasoftware.com/blog/the-business-value-of-better-requirements-management/>
- Wiegers, K. (2013). *Software Requirements*. En Redmond (Ed.). Washington: Third Edition ed.



Encuesta

La presente encuesta es parte de un estudio sobre la calidad de los proyectos de software desde la perspectiva de la ingeniería de requerimientos.

Desde ya las respuestas serán tratadas con absoluta confidencialidad y los resultados presentes de forma agrupada.

Por favor responde a todas las preguntas. Las respuestas son totalmente confidenciales, por lo que le pedimos lo haga con la mayor sinceridad posible, muchas gracias por su tiempo.

Encuesta

*Obligatorio

Dirección de correo electrónico *

Tu dirección de correo electrónico

Informacion General de la Empresa

1. Nombre de la Empresa

Tu respuesta

2. Industria o Sector

Tu respuesta

3. Tipo de Actividad

- ☐ Tecnológica
- ☐ Bienes y Servicios
- ☐ Gubernamental
- ☐ Industrial



4. Cantidad de Empleados

- ☐ 0 - 10
- ☐ 11 - 50
- ☐ 51 - 250
- ☐ Mas de 250 Empleados

5. Antigüedad de la Empresa

- ☐ Menor a 2 años
- ☐ 2 a 5 años
- ☐ 5 a 10 años
- ☐ Mayor de 10 años

6. Considerando los límites máximos expuestos en el cuadro de continuación.
¿En que categoría entra su empresa? Ventas totales anuales expresadas en
Pesos Argentinos 2020

Categoría	Construcción	Servicios	Comercio	Industria y minería	Agropecuaria
Micro	19.450.000	9.900.000	36.320.000	33.920.000	17.260.000
Pequeña	115.370.000	59.710.000	247.200.000	243.290.000	71.960.000
Mediana tramo 1	643.710.000	494.200.000	1.821.760.000	1.651.750.000	426.720.000
Mediana tramo 2	965.460.000	705.790.000	2.602.540.000	2.540.380.000	676.810.000

- ☐ Micro
- ☐ Pequeña
- ☐ Mediana T1
- ☐ Mediana T2



7. Cargo

Tu respuesta

8. Nivel de educación . Indique cuál es su nivel de educación

- ☐ Sin estudios universitarios
- ☐ Estudios universitarios
- ☐ Master
- ☐ Doctor

9. Cantidad de Personas a Cargo

Tu respuesta

10. Duración media de los proyectos en los que participa?

- ☐ Menos de 4 meses.
- ☐ Entre 4 meses y un año.
- ☐ Entre uno y dos años.
- ☐ Más de dos años.

11. Planificación de los proyectos. Valore la realización de la planificación de los proyectos en los que participa mediante una escala de 1(opción más desfavorable) a 5 (opción más favorable) *

1 2 3 4 5

Se planifica informalmente o de forma mínima

☐ ☐ ☐ ☐ ☐

Se planifican todas las áreas relacionadas con la gestión del proyecto (tiempo, costes, comunicaciones, calidad, riesgos, ...)



12. Seguimiento y control de los proyectos. Valore el seguimiento y control de los proyectos en los que participa mediante una escala de 1 (opción más desfavorable) a 5 (opción más favorable). *

1 2 3 4 5

Se monitoriza informalmente
o de forma mínima.

☐ ☐ ☐ ☐ ☐

Se monitoriza periódicamente
y sistemáticamente todos las
áreas relacionadas con la
gestión del proyecto (tiempo,
costes, comunicaciones,
calidad, riesgos, ...)

13. Avance (o retraso) del proyecto. Señale si tuvo algún retraso en los proyectos que ha trabajado. (expresado en porcentaje) *

1 2 3 4 5 6 7 8 9 10

Pocos proyectos con
retraso (1 - 10%)

☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐

Muchos proyectos con
retraso (10 - 100%)

14. Retraso del Proyecto. Considera que los proyectos se retrasaron por: *

- ☐ No se estableció correctamente el alcance
- ☐ Planificación de tiempos incorrecta
- ☐ El presupuesto muy ajustado
- ☐ Calidad del producto no cumple con los estándares
- ☐ Todas las anteriores
- ☐ Otros

15. Satisfacción del cliente del proyecto. Considera que la satisfacción del cliente, en los proyectos que participo se logro cumplir. *

- ☐ Nunca
- ☐ A veces
- ☐ Siempre



16. Proyecto Exitoso. De los aspectos listados a continuación, seleccione en que medida fueron considerados en los proyectos que trabajo. *

1= Muy mala 5= Excelente

	1	2	3	4	5	N/D
Se definieron claramente los objetivos del proyecto.	☐	☐	☐	☐	☐	☐
El líder del proyecto cuenta con el perfil profesional adecuado	☐	☐	☐	☐	☐	☐
La alta dirección brinda el apoyo al equipo del proyecto	☐	☐	☐	☐	☐	☐
Los integrantes del equipo fueron seleccionados apropiadamente	☐	☐	☐	☐	☐	☐
Fuéron asignados suficientes recursos materiales y económicos para el proyecto.	☐	☐	☐	☐	☐	☐
Se establecieron buenos canales de comunicación.	☐	☐	☐	☐	☐	☐



17. Señale cuales de estas técnicas listadas , has usado para el relevamiento de requerimientos y pondera el éxito que has tenido con las mismas. *

1= Muy mala 5= Excelente

	1	2	3	4	5	N/D
Lluvia de ideas:	☐	☐	☐	☐	☐	☐
Descomposición funcional:	☐	☐	☐	☐	☐	☐
Reglas de negocio:	☐	☐	☐	☐	☐	☐
Prototipos:	☐	☐	☐	☐	☐	☐
Entrevista:	☐	☐	☐	☐	☐	☐
Análisis de interfaz:	☐	☐	☐	☐	☐	☐
Análisis de documentos:	☐	☐	☐	☐	☐	☐
Grupo de enfoque (focus group):	☐	☐	☐	☐	☐	☐
Taller de requerimientos:	☐	☐	☐	☐	☐	☐
Modelado de procesos:	☐	☐	☐	☐	☐	☐
Descubrimiento de escenarios:	☐	☐	☐	☐	☐	☐
No se encuentra en la lista la técnica que uso	☐	☐	☐	☐	☐	☐
No uso ninguna técnica	☐	☐	☐	☐	☐	☐



18. Entre la metodología tradicional y Ágil, desde su experiencia con cual ha tenido mayor éxito en los proyectos? *

- ☐ Metodología Ágil
- ☐ Metodología Tradicional
- ☐ Ambas
- ☐ Ninguna de las mencionadas
- ☐ Otro: _____

19. Considera que las ilustraciones (mapas mentales, UML, etc...), puede servir como un mecanismo para capturar los requisitos de una manera más estructurada, rápida, clara y persuasiva ayudando tanto a los usuarios, como al equipo de trabajo *

- ☐ Nunca
- ☐ A veces
- ☐ Siempre

20. En los proyectos que ha trabajado o trabaja actualmente, usa una guía o manual de buenas practicas para el relevamiento de los requerimientos. *

- ☐ Si
- ☐ No

21. Cual es el factor de exito que ha tenido usando una guía o manual de buenas prácticas para el relevamiento de requerimientos (Solo si contestaste "SI" a la pregunta 20) *

- | | 1 | 2 | 3 | 4 | 5 | |
|------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-------------|
| Poco Exito | ☐ | ☐ | ☐ | ☐ | ☐ | Muy Exitoso |



22. Trabajo de Equipo. De las características listadas a continuación, cual(es) considera que estuvieron presentes durante el proyecto, con su equipo de trabajo. *

1= Poco Presente 5= Muy Presente

	1	2	3	4	5	N/D
El grupo ha experimentado fracasos recientes	☐	☐	☐	☐	☐	☐
Su equipo de trabajo sobreestima su velocidad y capacidad	☐	☐	☐	☐	☐	☐
Composición homogénea de grupo	☐	☐	☐	☐	☐	☐
No existe un método estándar para evaluar ideas y decisiones.	☐	☐	☐	☐	☐	☐
El equipo está constantemente de acuerdo con poca discusión durante las reuniones.	☐	☐	☐	☐	☐	☐
Esteriotipos negativos de los que están fuera del grupo.	☐	☐	☐	☐	☐	☐
Líderes que promueven sus propias soluciones	☐	☐	☐	☐	☐	☐
En las reuniones ¿El silencio se percibe como aceptación?	☐	☐	☐	☐	☐	☐

23. ¿Todos los miembros del equipo están satisfechos con el resultado de los proyectos? *

	1	2	3	4	5	6	7	8	9	10	
Poco de Acuerdo	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	Totalmente de acuerdo

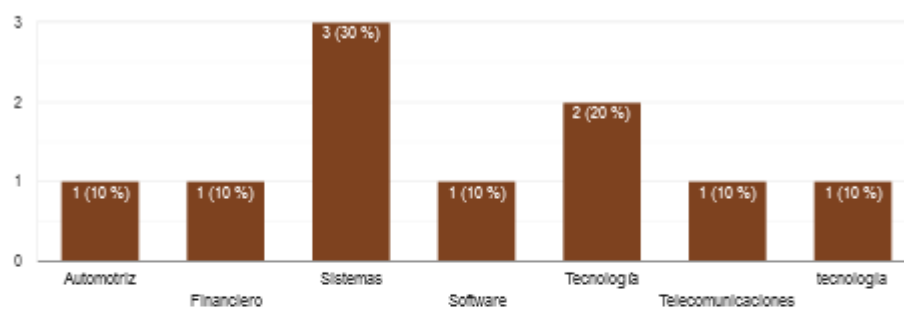


Resultados

Una muestra de las respuestas obtenidas por parte de los encuestados.

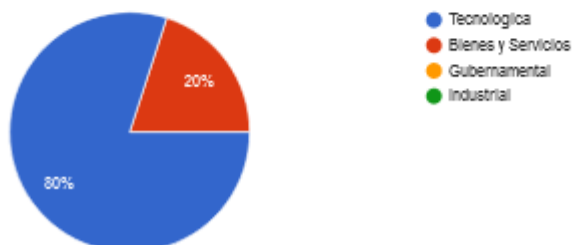
2. Industria o Sector

10 respuestas



3. Tipo de Actividad

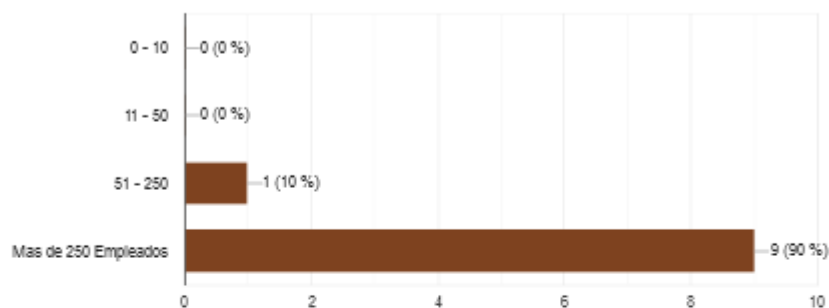
10 respuestas





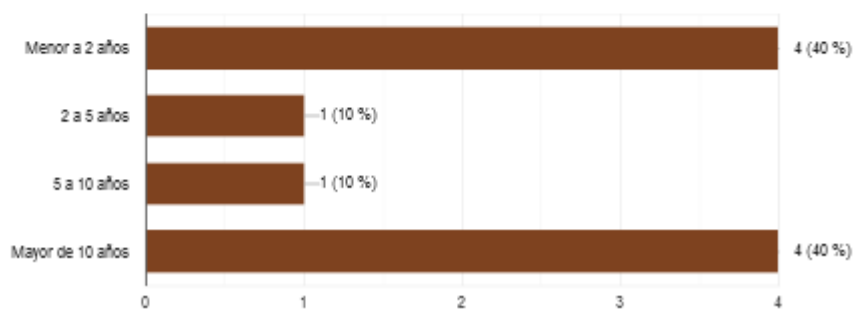
4. Cantidad de Empleados

10 respuestas



5. Antigüedad de la Empresa

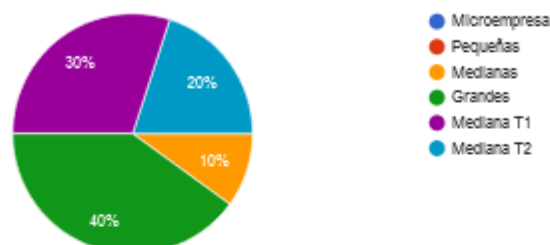
10 respuestas





6. Considerando los límites máximos expuestos en el cuadro de continuación. ¿En qué categoría entra su empresa? Ventas totales anuales expresadas en Pesos Argentinos 2020

10 respuestas



Información a quién corresponda la encuesta

7. Cargo

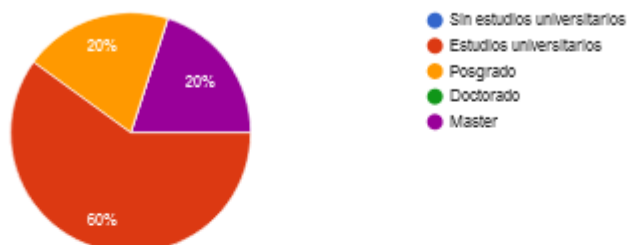
10 respuestas





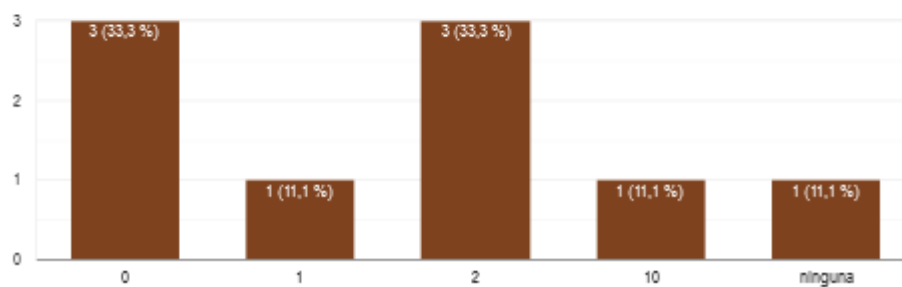
8. Nivel de educación . Indique cuál es su nivel de educación

10 respuestas



9. Cantidad de Personas a Cargo

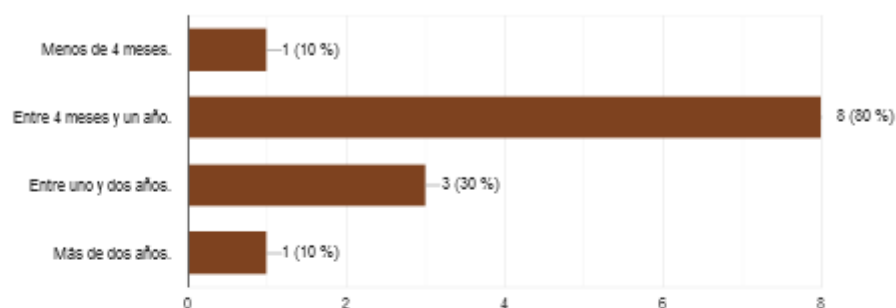
9 respuestas





10. Duración promedio de los proyectos en los que participa?

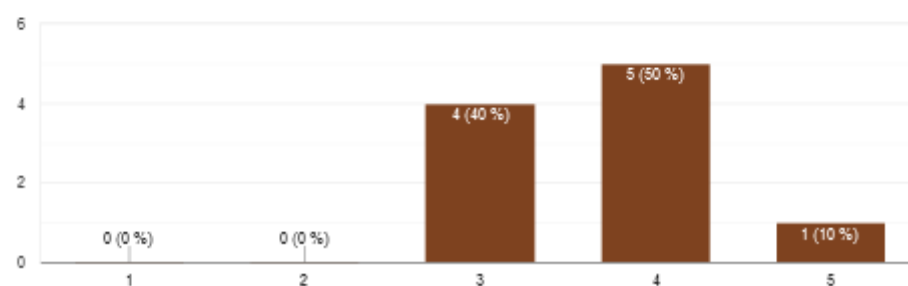
10 respuestas



11. Planificación de los proyectos. Valore la realización de la planificación de los proyectos en los que participa mediante una escala de 1 (opción más desfavorable) a 5 (opción más favorable)



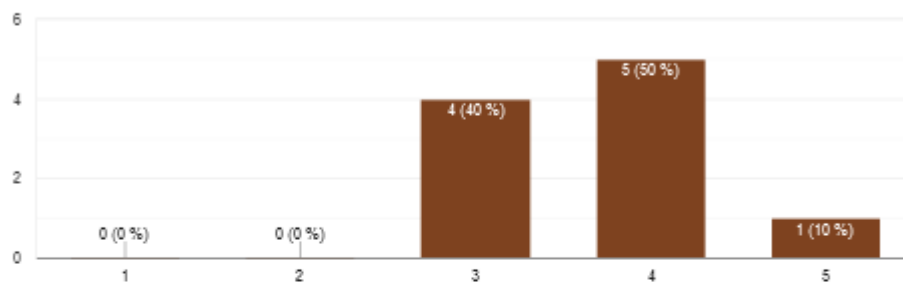
10 respuestas





12. Seguimiento y control de los proyectos. Valore el seguimiento y control de los proyectos en los que participa mediante una escala de 1 (opción más desfavorable) a 5 (opción más favorable).

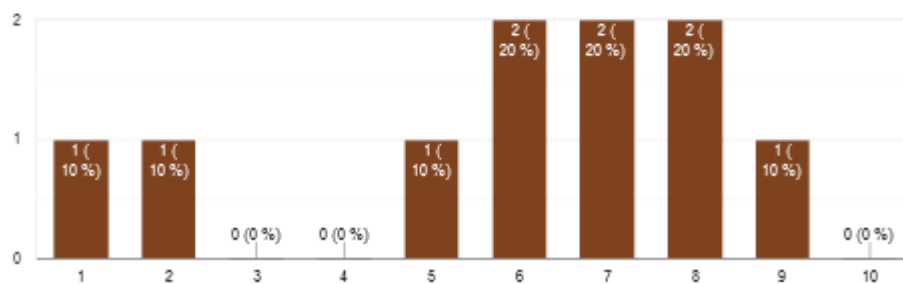
10 respuestas



13. Avance (o retraso) del proyecto. Señale si tuvo algún retraso en los proyectos que ha trabajado. (expresado en porcentaje)



10 respuestas





14. Retraso del Proyecto. Considera que los proyectos se retrasaron por:

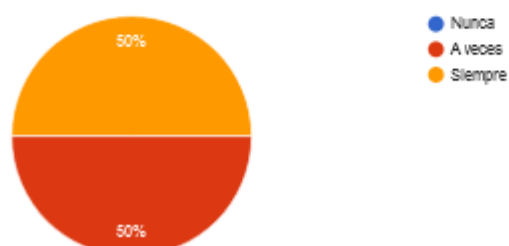
10 respuestas



15. Satisfacción del cliente del proyecto. Considera que la satisfacción del cliente , en los proyectos que participo se logro cumplir.

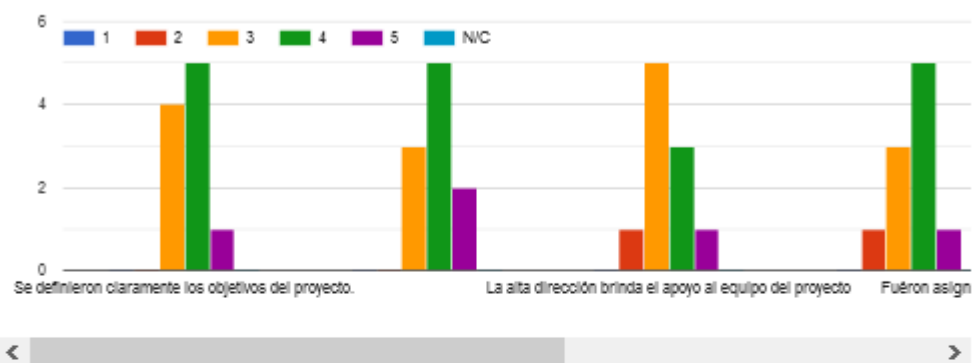


10 respuestas

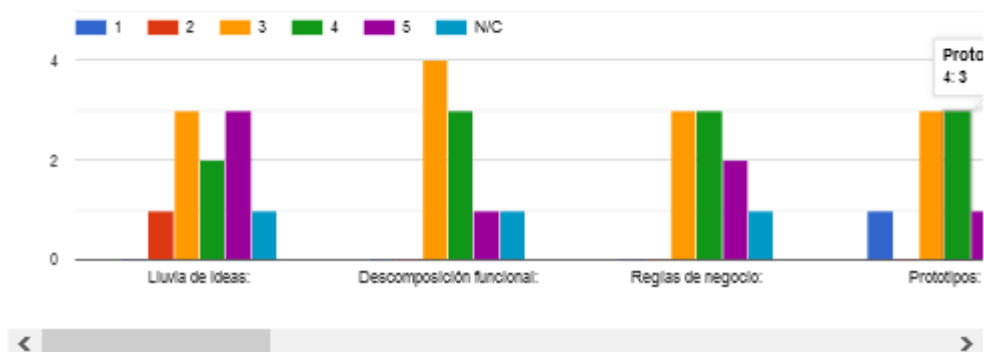




16. Proyecto Exitoso. De los aspectos listados a continuación, seleccione en qué medida fueron considerados, en los proyectos que usted trabajó.



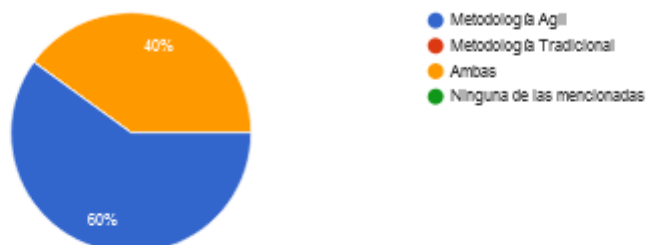
17. Señale cuales de estas técnicas listadas, ha usado para el relevamiento de requerimientos y pondera el éxito obtenido con las mismas.





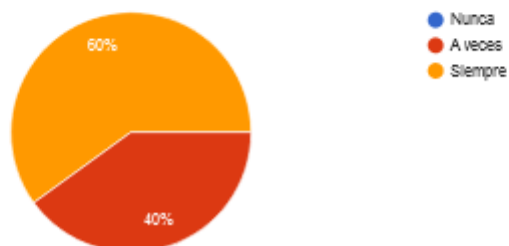
18. Entre la metodología tradicional y Ágil, desde su experiencia con cual ha tenido mayor éxito en los proyectos?

10 respuestas



19. Considera que las ilustraciones (mapas mentales, UML, etc...), puede servir como un mecanismo para capturar los requisitos de una manera más estructurada, rápida, clara y persuasiva, ayudando tanto a los usuarios como al equipo de trabajo

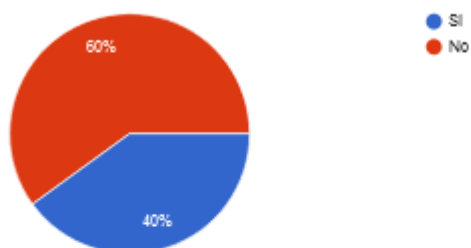
10 respuestas





20. En los proyectos que ha trabajado o trabaja actualmente, usa una guía o manual de buenas prácticas para el relevamiento de los requerimientos.

10 respuestas



21.Cuál es el factor de éxito que ha tenido usando una guía o manual de buenas prácticas para el relevamiento de requerimientos (Solo si contestaste "Si" a la pregunta 20)



10 respuestas

