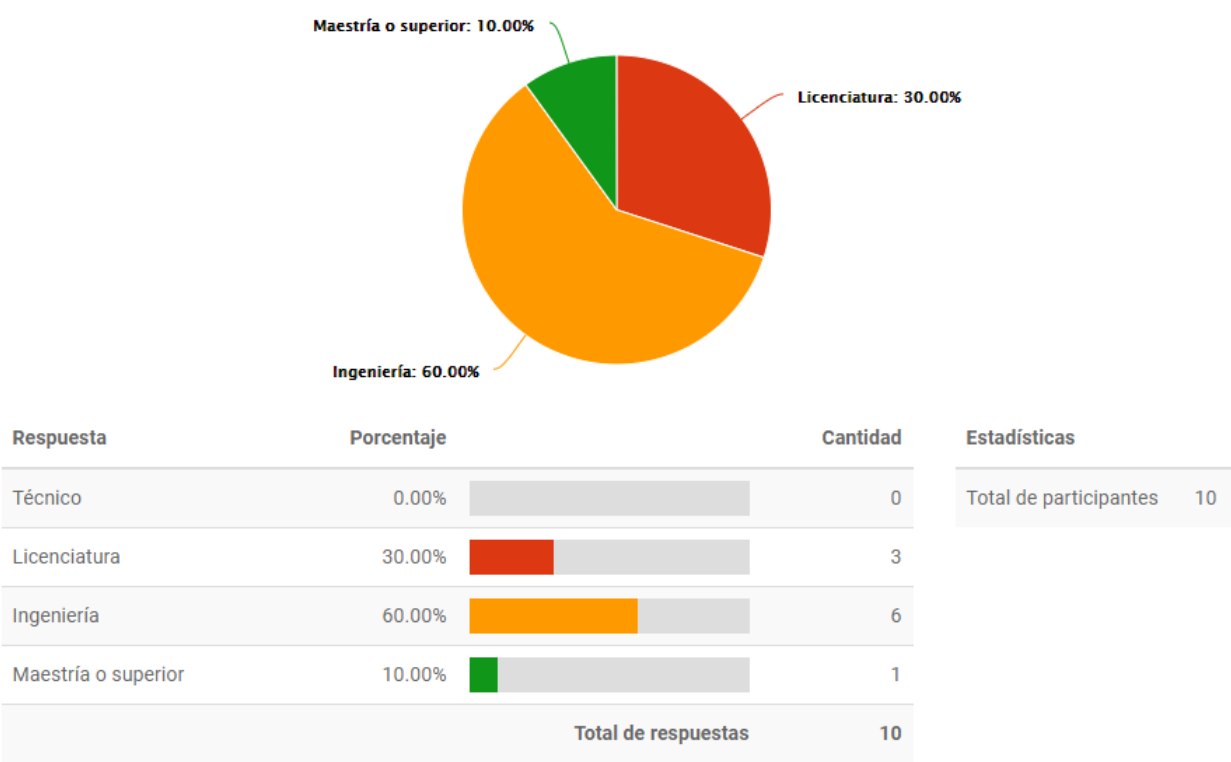
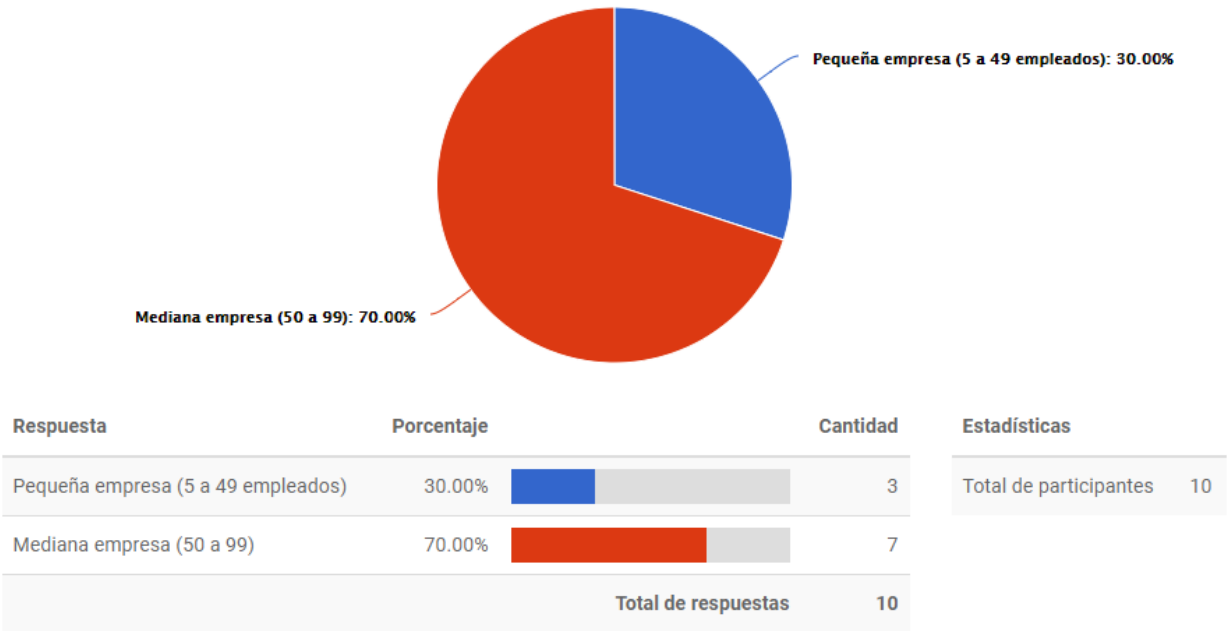


Informe de resultados

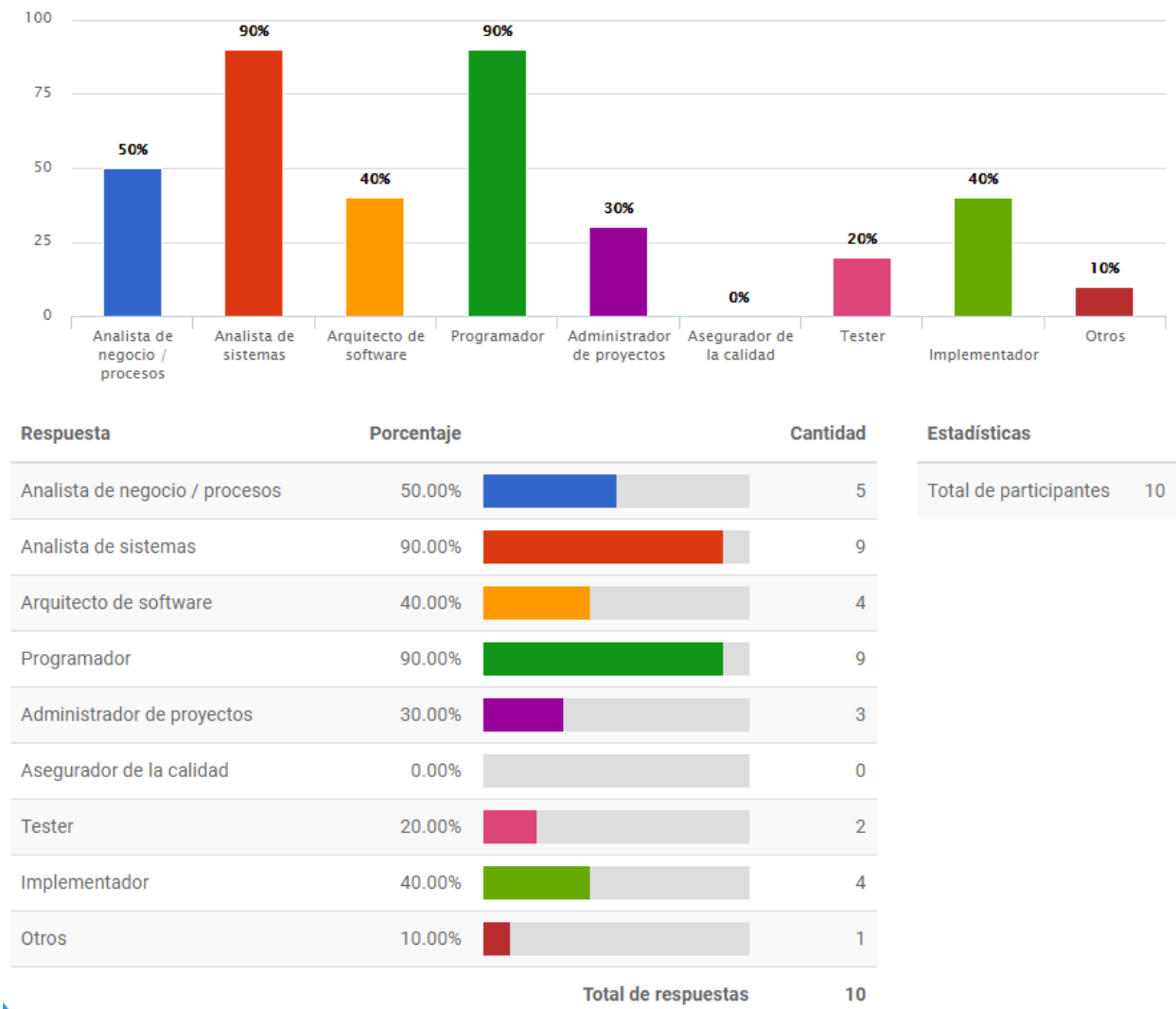
1 Nivel educativo que posee



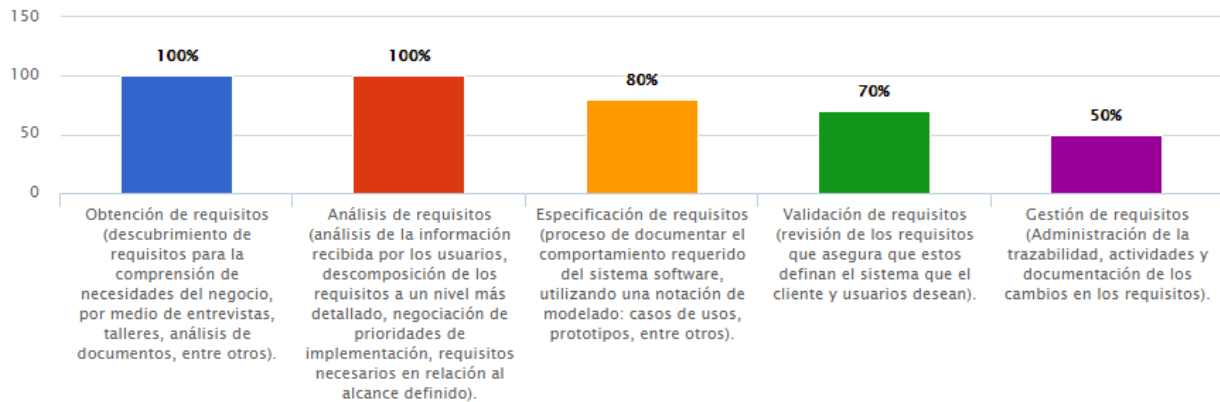
2 Tamaño de la empresa donde labora actualmente



3 Roles que ha desempeñado en su carrera profesional (Incluya el que desempeña en la actualidad)

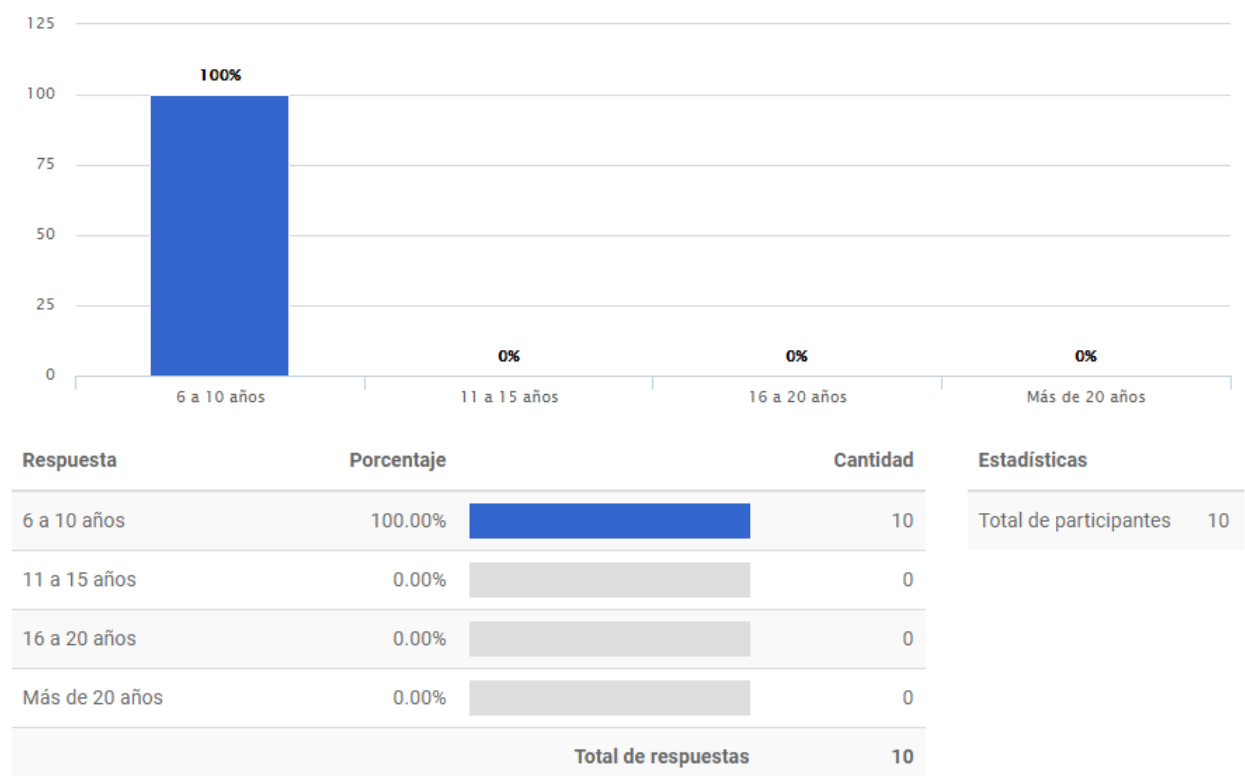


4 ¿En qué actividad de requisitos de software ha participado? (Selección múltiple)



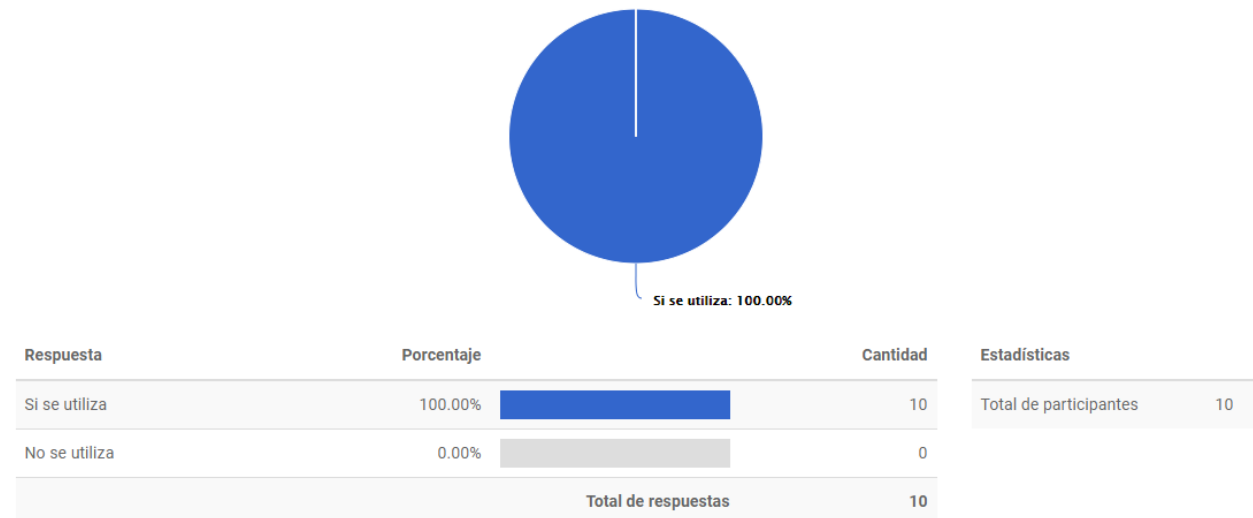
Respuesta	Porcentaje	Cantidad	Estadísticas
Obtención de requisitos (descubrimiento de requisitos para la comprensión de necesidades del negocio, por medio de entrevistas, talleres, análisis de documentos, entre otros).	100.00%	10	<div>Total de participantes 10</div>
Análisis de requisitos (análisis de la información recibida por los usuarios, descomposición de los requisitos a un nivel más detallado, negociación de prioridades de implementación, requisitos necesarios en relación al alcance definido).	100.00%	10	
Especificación de requisitos (proceso de documentar el comportamiento requerido del sistema software, utilizando una notación de modelado: casos de usos, prototipos, entre otros).	80.00%	8	
Validación de requisitos (revisión de los requisitos que asegura que estos definan el sistema que el cliente y usuarios desean).	70.00%	7	
Gestión de requisitos (Administración de la trazabilidad, actividades y documentación de los cambios en los requisitos).	50.00%	5	
Total de respuestas			10

5 ¿Años de experiencia desempeñando puestos o roles relacionados a requisitos de software?

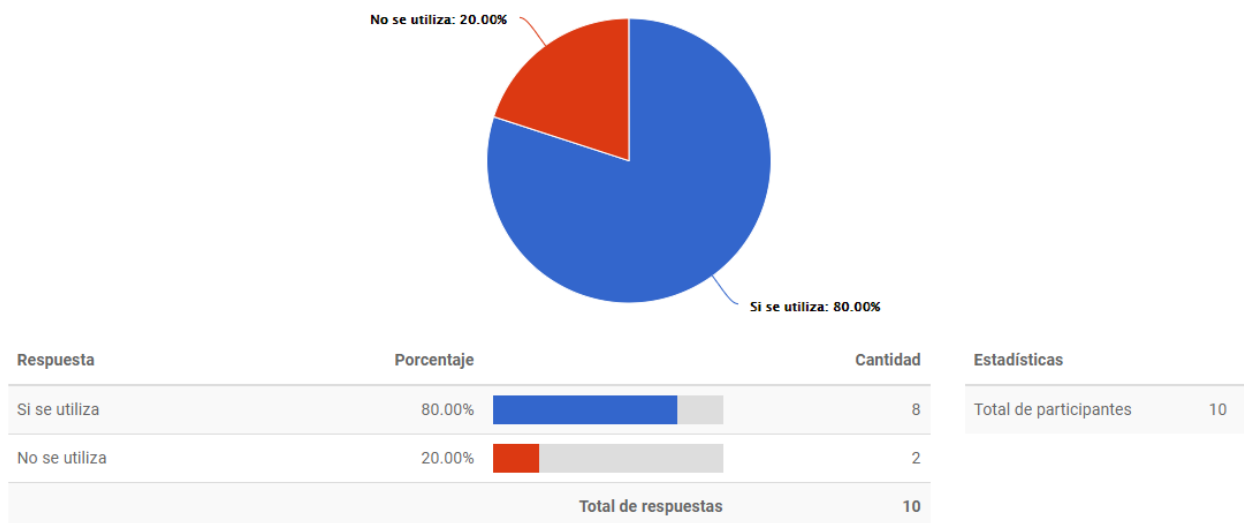


6 Según su experiencia, valide la utilización de las buenas prácticas de OBTENCIÓN de requisitos siguientes:

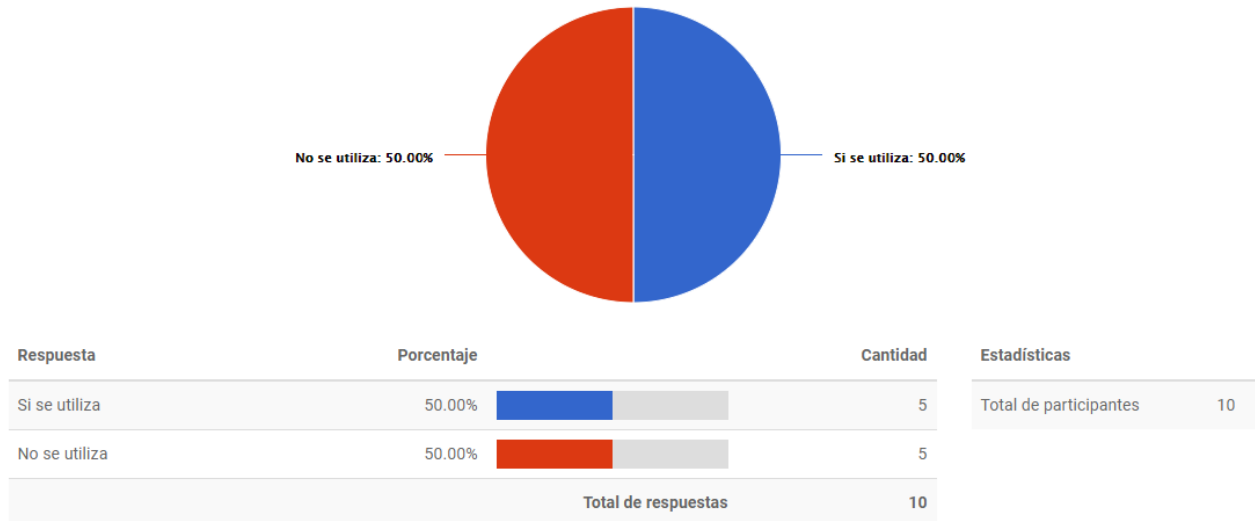
1. Utilizar técnicas para obtención de requisitos (Entrevistas, cuestionarios, observación, historia de usuarios, casos de uso, escenarios, prototipos, talleres de facilitación, análisis de documentos, análisis de tormentas de ideas y examinar informes de problemas).



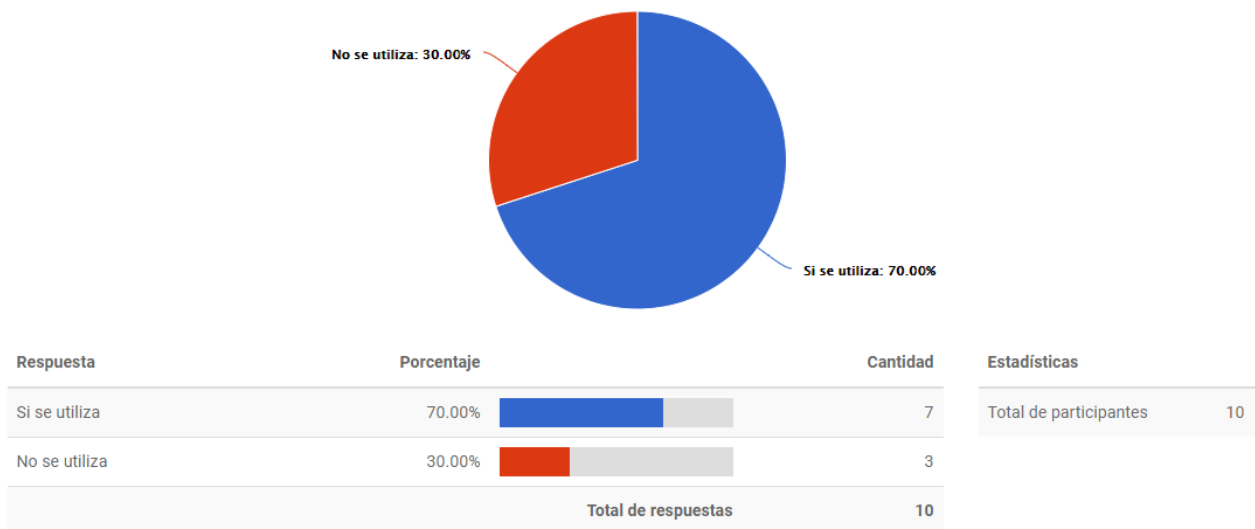
2. Identificar, verificar e involucrar a las partes interesadas (stakeholders) y administrar los diferentes puntos de vista.



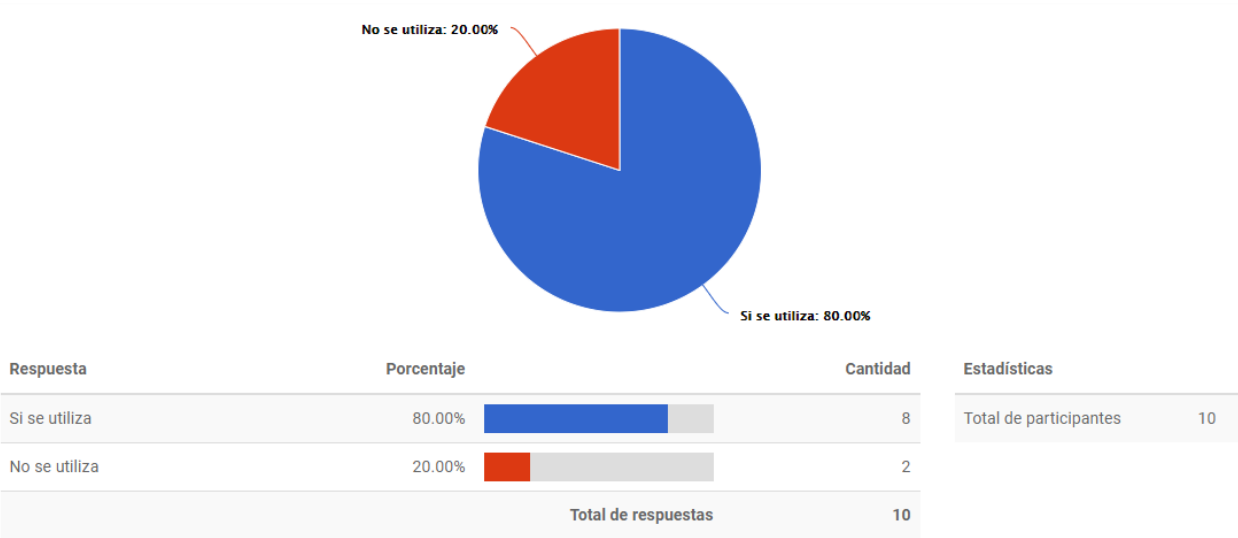
3. Definir y documentar claramente el alcance y visión del proyecto.



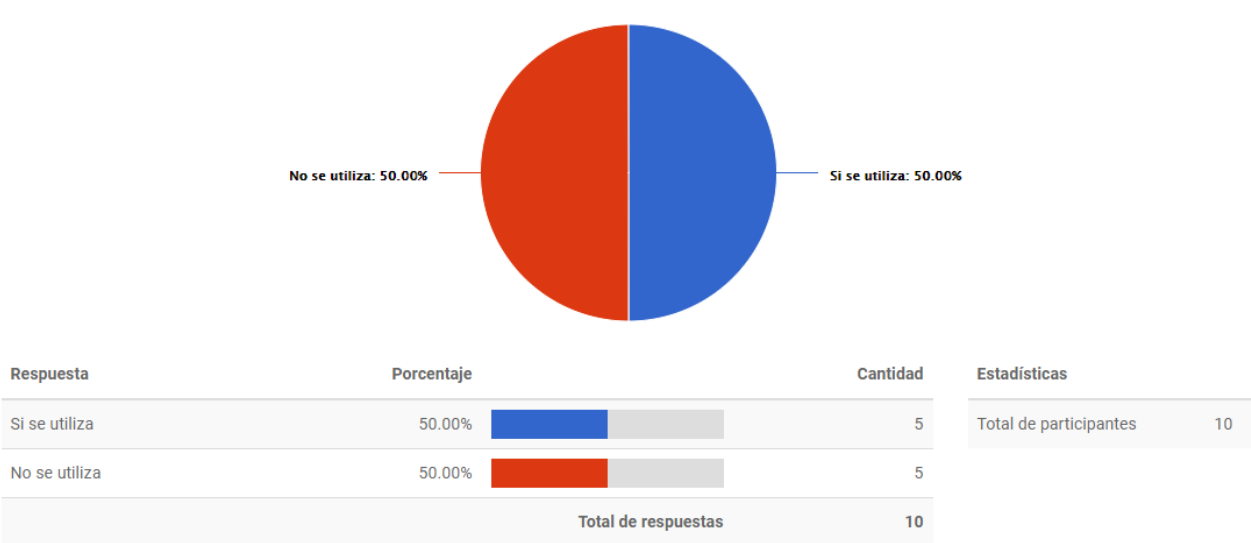
4. Identificar y evaluar todas las fuentes potenciales de requisitos (políticas de negocio, procedimientos, estándares, lecciones aprendidas, archivos de proyectos anteriores, requisitos del entorno del negocio, componentes de productos heredados y estatutos reguladores).



5. Establecer y mantener acuerdos con los stakeholders (negociación y resolución de conflictos).



6. Comunicar de manera efectiva y utilizar modelos de comunicación.



7 Explicación del experto (razones de la no utilización o argumentaciones adicionales que justifique la utilización de las buenas prácticas anteriores):

ENTREVISTADO	RESPUESTA
1	<p>Pregunta 2. En el momento de realizar la obtención de los requisitos es necesario que diferentes personas aporten sus puntos de vista ya que la experiencia en el negocio es diferente en cada uno de ellos.</p> <p>Pregunta 3. En ocasiones el usuario final o dueño del proceso no encuentra la forma de solicitar expresando claramente lo que necesita. Pero al presentar un documento este sirve de partida contractual.</p> <p>Pregunta 5. El analista debe definir los límites del alcance para el desarrollo, por medio del establecimiento de acuerdos con las partes involucradas.</p>
2	<p>Pregunta 2. No aplica, porque el requerimiento muchas veces se ve con un solo encargado y no se toman en cuenta todas las partes involucradas.</p>
3	<p>Pregunta 2. Esta buena práctica es una de las más importantes, ya que si no se involucra e identifica a todas las partes interesadas del proyecto, esto puede afectar a la hora de pasar al ambiente de producción el desarrollo, debido a que se tendría que trabajar y cumplir los requisitos del nuevo stakeholders lo que se traduce en costos, ya que causa atraso en las campañas programadas porque afecta directamente la planificación estratégica de la organización y por ende la del proyecto.</p> <p>Punto 6. Uno de los principales problemas es que la comunicación no fluye correctamente a todos los involucrados, quedándose hasta ciertos niveles organizaciones de los que forman parte del proyecto. Para poder aplicar esta buena práctica debe definirse un procedimiento de cómo se llevara a cabo la comunicación es decir un plan de comunicación, a su vez se debe delegar a una persona como responsable del cumplimiento de estas funciones.</p>
4	<p>Pregunta 1. En mi caso personal la utilización no se da al 100%, las partes involucradas muchas veces dejan solo en la parte de obtención al área de sistema y es sistemas el que se encarga de ver cuáles son los requisitos que se necesitan, cuando ya se tiene un modelo este se presenta a ciertas partes involucradas no todas ya que el factor tiempo afecta de gran manera el inicio de la obtención de los requisitos.</p> <p>El documentar la obtención no es viable ya que requiere de una gran cantidad de tiempo y por lo general no se tiene a una persona propiamente designada para esta función.</p>
5	<p>Pregunta 3. Al no poseer una metodología bien definida para llevar acabo sesiones y control del proyecto con los usuarios, estos no se interesan en la documentación sino en obtener resultados.</p> <p>Pregunta 4. Generalmente los analistas de sistemas o la persona a cargo solo se centra en la necesidad inmediata que hay que cubrir, es decir presentar un resultado a medias ya que</p>

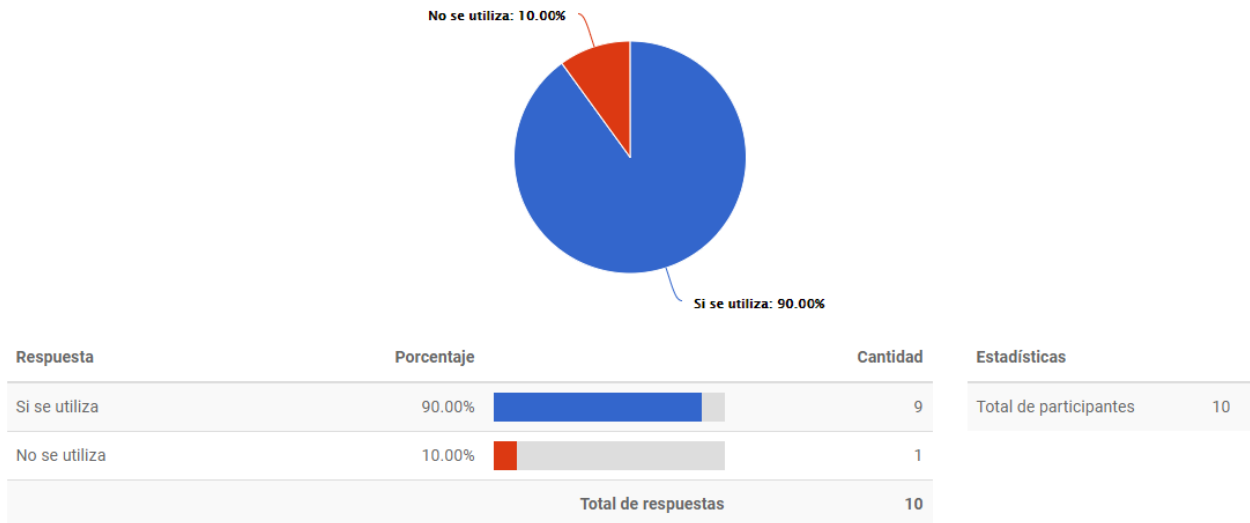
UN FRAMEWORK DE BUENAS PRÁCTICAS PARA LOS REQUISITOS DE SOFTWARE

	<p>muchas veces no se analizan si ya existen precedentes de proyectos anteriores, estatutos reguladores o estándares con los que habría que alinearse o tomar en cuenta.</p> <p>Pregunta 6. Los stakeholder no comprenden la importancia de su participación en el proceso de ciclo de vida del software, interesándose en tener solo la herramienta final.</p>
6	<p>Debido a la diversidad de pensamiento e interpretación, tanto del que toma los requisitos como del usuario, es necesario establecer lo más pronto posible sesiones de trabajo donde aparte de hacer el levantamiento del requisito, se puedan identificar "puntos secundarios críticos" que impactan sobre el requisito, los cuales pueden darse por entendido por el usuario, debido a que de forma natural se utilizan en su operación, pero no necesariamente son visibles en primera línea en el proceso. Ejemplos: Leyes fiscales o particulares de los clientes de los usuarios, procesos internos particulares aplicados al proceso estándar, etc. También es necesario corroborar y validar 2 o tres veces la toma del requisito ya que a veces el usuario tiene una idea del requerimiento pero necesita consolidar y aterrizar la idea final.</p>
7	<p>Pregunta 2. Si no se identifica a las partes interesadas, todos tendrían una visión diferente del proyecto.</p> <p>Pregunta 3. Esta buena práctica tiene relación directa con el punto 2, El definir el alcance y visión del proyecto permite administrar una única visión del proyecto, en donde quedan plasmadas las ideas y objetivos a cumplir.</p> <p>Pregunta 4. Apoya como un punto de partida y respaldo, ya que si en un futuro se dan cambios organizacionales no afecta con gran impacto al proyecto, evitando iniciar de nuevo.</p> <p>Pregunta 5. Uno de factores del porque esta práctica no se aplique es debido a que los involucrados muchas veces no toman con seriedad el proyecto. Otro factor es que exista una figura autoritaria (gerente, jefatura, entre otros) que se encarga de delegar (imponer) que se va a desarrollar sin importar la opinión de las otras partes, es decir no importa quienes salgan afectadas.</p> <p>Pregunta 6. Hay que utilizar modelos efectivos y que muestren el enfoque de seguimiento al proyecto. Considero que es importante como buena práctica pero estableciendo los identificadores o parámetros a comunicar.</p>
8	<p>Pregunta 3,4 y 6. Las respuestas marcadas como no aplicables, no es porque no se puedan aplicar, sino porque son las principales carencias que a mi experiencia tenemos en el país en cuanto al proceso desarrollo de software.</p>
9	<p>Pregunta 3. Definir y documentar claramente el alcance y visión del proyecto, no se hace; porque surgen solicitudes de cambio y se obvia la documentación. Esto trae como consecuencia que al finalizar el software, este termina siendo diferente al requerimiento inicial, esto porque no se identifica desde un inicio los requisitos y el alcance real.</p>

	<p>Pregunta 4. Esta buena práctica ayuda a identificar hasta el tema de infraestructura que se puede reutilizar, cuando hay servicios web que ya hacen algo, como componentes de recursos heredados que permiten agilizar el desarrollo del software.</p> <p>Pregunta 5. Establecer y mantener acuerdos con los stakeholders (negociación y resolución de conflictos). Tiene que ver con el punto 3, ya que desde la visión y el alcance del proyecto podemos ir evaluando los requerimientos que cambian con el tiempo por las mismas necesidades organizacionales, en consecuencia siempre se debe estar abierto a los cambios.</p> <p>Pregunta 6. Debe de ir escrito en el documento de acuerdos, el modelo de comunicación. Se da en el país que piden aumentar los requisitos y solo de palabra, lo ideal es que quede documentado aunque sea por medio de un correo, y que esa información pueda estar accesible a las partes involucradas, por ejemplo: pueden realizarse minutas de reuniones, en mi caso personal donde laboro se hace una minuta para comprobar que se han solicitado cambios.</p>
10	<p>Pregunta 3. En mi experiencia hubieron muchos cambios de alcance que afectaron ya el curso construido, es buena práctica siempre y cuando no se cambie a corto plazo.</p> <p>Pregunta 5. El no establecer y mantener acuerdos con los stakeholder, en mi caso personal no se aplicó muchas veces y afectó, por otro lado al sí aplicar esta buena práctica pero no comunicar los acuerdos también afectó.</p> <p>Pregunta 6: La comunicación muchas veces en las reuniones no es eficaz en el ciclo de vida del proyecto, debido a que no existen indicadores a medir. Ejemplo de ello son las reuniones diarias en SCRUM que muchas veces se llevan a cabo solo por salir del compromiso, otro ejemplo seria las reuniones de retrospectivas del sprint, que sucede lo mismo al no comunicar eficientemente a todas las partes interesadas las cosas a mejorar (indicadores) tanto en el equipo de desarrollo como las partes del negocio.</p>

8 Según su experiencia, valide la utilización de las buenas prácticas de ANÁLISIS de requisitos siguientes:

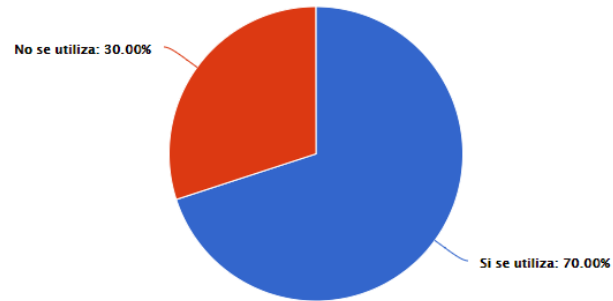
1. Priorizar los requisitos de software.



2. Analizar los riesgos y viabilidad de los requisitos.

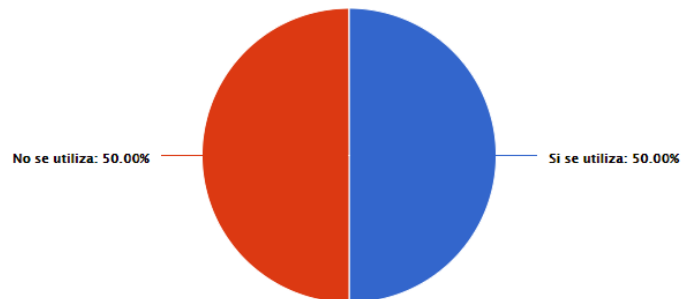


3. Construir de modelos técnicos (diagramas de casos de uso, modelos de flujo de datos, modelos de estado, modelos basados en objetivos, interacciones de usuarios, modelos de objetos, modelos de datos, entidad relación, transición de estado, entre otros), simulaciones y prototipos.



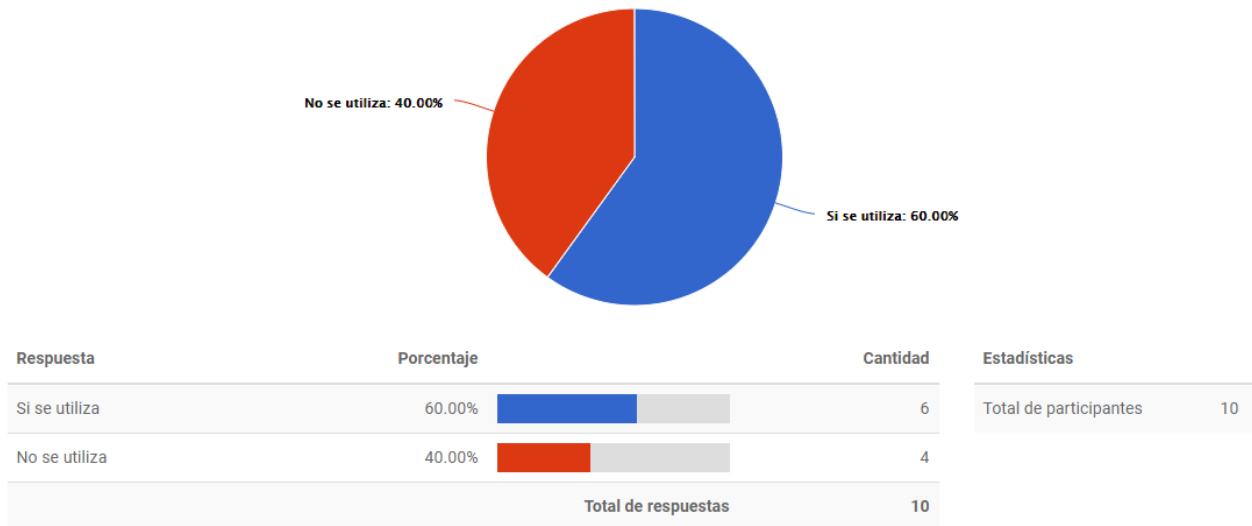
Respuesta	Porcentaje	Cantidad	Estadísticas
Si se utiliza	70.00%	7	Total de participantes 10
No se utiliza	30.00%	3	
Total de respuestas			10

4. Desarrollar modelado conceptual (contexto).



Respuesta	Porcentaje	Cantidad	Estadísticas
Si se utiliza	50.00%	5	Total de participantes 10
No se utiliza	50.00%	5	
Total de respuestas			10

5. Descomposición lógica (clasificación, relación y jerarquía de requisitos).



9 Explicación del experto (razones de la no utilización o argumentaciones adicionales que justifique la utilización de las buenas prácticas anteriores):

ENTREVISTADO	RESPUESTA
1	<p>Pregunta 3. El modelado técnico muchas veces se realiza a medias o no se realiza, porque se considera una pérdida de tiempo y se prefiere pasar directamente al desarrollo; lo cual causa reprocesos en el desarrollo al no tenerlos, ya que son un gran aporte en el entendimiento del problema a solucionar.</p>
2	<p>Pregunta 1. En mi caso personal los requisitos se priorizan en base a los riesgos y el impacto negativo que pueden tener. Por ejemplo con entes externos tales como: proveedores, entes reguladores, gobierno, competencia directa en el rubro, entre otros.</p> <p>Pregunta 3. El no tener modelos técnicos (diagramas) impacta directamente en el entendimiento de los procesos de negocio. Estos se utilizan para estimar tiempos en el plan de trabajo del equipo de desarrollo, lo que permite tener una mejor visión de las actividades a desarrollar. Esta buena práctica se puede utilizar solo para proyectos grandes, que sobrepasen un tiempo mayor a 2 meses, en mi caso personal creo que no es necesario para pequeños cambios o actualizaciones menores en el software, esto debido a con frecuencias se realizan pequeños ajustes en contratiempo para la operatividad del negocio.</p> <p>Pregunta 4. Es el modelado más común y utilizado para cualquier tamaño de proyecto, ayuda mucho a entender la lógica de funcionamiento del negocio sin entrar a detalles técnicos.</p> <p>Pregunta 5. Esta parte en el país se realiza más a criterio de cada desarrollador por llevar un orden de manera personal, no hay un proceso, no se clasifican los requisitos principalmente</p>

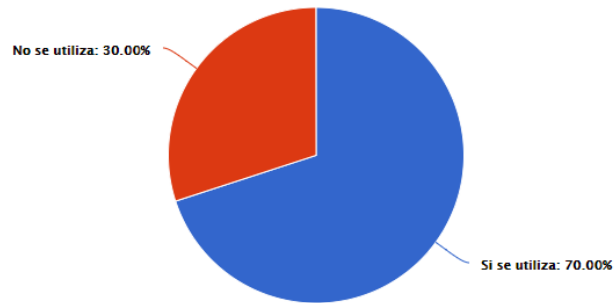
	<p>en los no funcionales. Pero debería de realizarla formalmente un analista de sistemas antes de comenzar el desarrollo del software.</p>
3	<p>Pregunta 1. El priorizar los requisitos de software es muy utilizado debido a que hay requisitos que son más grandes que otros, esta práctica ayuda a realizar una designación más equitativa ya que permite asignar los más trabajosos al personal con mayor experiencia, para realizarlos en un tiempo estipulado. El factor impacto es lo que define como se priorizan los requisitos, por ejemplo los cambios que vienen de ley tales como: la modificación de porcentajes del AFP, que van a afectar planillas y otros procesos. Como son cambios de prioridad Alta e Impacto Alto, se asignan a las personas con mayor experiencia.</p> <p>Pregunta 2. Analizar los riesgos y viabilidad de los requisitos. En mi experiencia esta buena práctica no se hace, dándose cuenta de los riesgos o factibilidad hasta que ya se está en la marcha del desarrollo, pero a mi juicio lo ideal sería que si se realizara. Uno de los factores del porque no se utiliza es debido a que no se le dedica tiempo por lo mismo que en el país todo es para ya.</p> <p>Pregunta 3. Construcción de modelos técnicos, en mi experiencia actual es el arquitecto de software quien realiza esto (modelado o diagramación) y a resultado de gran ayuda en el entendimiento, el requisito va claro y dibujado para lo que se va a hacer, uno de los beneficios es que agiliza el desarrollo para que el desarrollador solo arme, sé que en muchos lugares no se hace porque en el país todo es para ya (urgencia), todo es para ayer, y por la premura esto se lo pasan por alto. En su experiencia anterior solo se utilizaba como una documentación posterior pero nunca como un diseño para el análisis, en el cual se incluyen los patrones de diseño lo que permite reutilizar. La diferencia con mi experiencia anterior y a la actual está en que existan personas dedicadas a cada modelado como arquitectos y analistas, en donde influye mucho el tamaño de la empresa, en empresas más pequeñas normalmente el mismo que analiza, programa y hace todo, por lo que en empresas un poco más grandes ya del tipo medianas ya hay roles como Project manager, analistas de sistemas, arquitectos de software, coordinador de analistas, programadores, a la larga los beneficios se ven. En empresas medianas estas prácticas son mejor aplicables y definidas para cada etapa del desarrollo de software, incluyendo tester, encargados de calidad, y la implementación entre el project manager y el arquitecto.</p> <p>Pregunta 4. Desarrollar modelado conceptual (contexto). Si es muy necesario, muy utilizados y si se dedica el tiempo para hacerse, más que todo sirve de apoyo para que los arquitectos puedan identificar esos componentes que van a ser necesarios en el software.</p> <p>Pregunta 5. Para esta buena práctica considero que si ya cumplimos con la priorización de requisitos, considero que la descomposición es casi automática, y que la clasificación depende de la delicadeza del requisito por lo que se debe hacer.</p>
4	<p>Pregunta 1. Sirve ya que se dan casos que todo es prioridad pero debe definirse una ruta, entendiendo que todo es crítico pero con una estrategia de cual atacar primero.</p>

UN FRAMEWORK DE BUENAS PRÁCTICAS PARA LOS REQUISITOS DE SOFTWARE

	<p>Pregunta 2. En El Salvador no se utiliza como debería de hacerse, solo se lleva a cabo por cumplir un requisito el cual permita poder pasar a producción.</p> <p>Pregunta 3. Proporciona facilidad de entendimiento para poder desarrollar puntualmente la necesidad que solicita el usuario final.</p> <p>Pregunta 4. El usarlo daría una buena apertura o visión del negocio o entidades involucradas.</p> <p>Pregunta 5. si es necesario para llevar un mejor seguimiento y entendimiento de los requisitos. de manera informal lo llevamos, ayuda a ordenar las ideas y con que hay que cumplir ya sea de leyes internas o externas es decir el cumplimiento organizacional</p>
5	<p>Pregunta 3. Depende del tamaño y de las personas que forman el equipo del proyecto, porque hay ocasiones que una sola persona realiza diferentes roles, lo que requiere una mayor inversión en el tiempo y los resultados los quieren para ayer.</p> <p>Pregunta 5. Hay ocasiones en que existen personas que poseen peso en el negocio (jefes por ejemplo) que imponen una jerarquía de requisitos.</p>
6	<p>Pregunta 4 y 5. En muchos casos los líderes de proyecto presionan por que los tiempos de entrega sean tempranos y no observan el valor agregado que el desarrollo de modelos o descomposición lógica ofrecen.</p>
7	<p>Pregunta 4 y 5. Cuando se hace el análisis si se toma en cuenta cada punto de los anteriores expuesto ya que de esto depende el buen funcionamiento, en ocasiones y por el factor tiempo podría quedar pendiente la descomposición lógica así como el modelo conceptual, los diagramas son muy importantes en esta fase por eso no pueden ser obviados</p>
8	<p>Pregunta 4 y 5. El desarrollo de modelado o diagramación así como la descomposición de los requisitos la mayoría de ocasiones no se utiliza por ahorrar tiempo y costos.</p>
9	<p>Pregunta 2. No se utiliza ya que esta buena práctica no se hace a conciencia, en mi caso no soy responsable de realizarlo.</p> <p>Pregunta 4. No se utilizó bajo la experiencia del proyecto en el que participo actualmente.</p>
10	<p>La creación de diagramas y esquemas es muy importante en la fase de análisis ya que permite hacer una revisión ágil del requerimiento ("una imagen vale más que mil palabras..."). Es recomendable que en la revisión se utilicen materiales de apoyo como pizarras, post-it o similares y se anime al usuario a participar en la corrección o complementación del requerimiento, teniendo presente siempre por parte del analista la delimitación y alcance del problema. Basado en mi experiencia, cuando entra en un ambiente "jovial" se motiva la creatividad, lo cual ayuda para la revisión y hasta mejora del proceso planteado. En la medida de lo posible trato de aplicarlo, pero esto depende de la confianza y relación con los usuarios.</p>
Total de respuestas	10

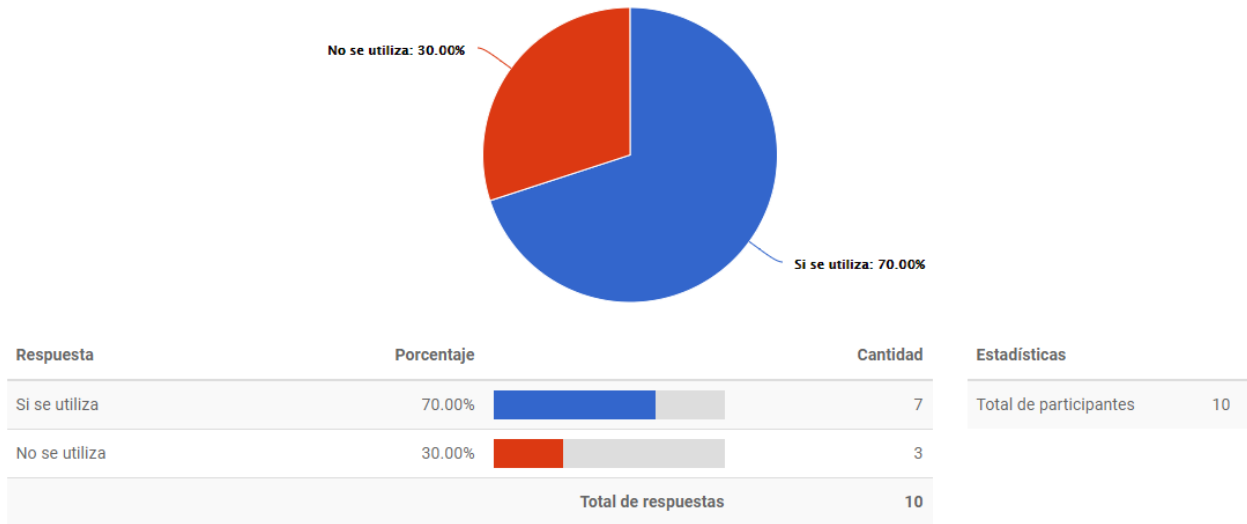
10 Según su experiencia, valide la utilización de las buenas prácticas de **ESPECIFICACIÓN** de requisitos siguientes:

1. Crear artefactos (documentar la visión y arquitectura del negocio, reglas del negocio, requisitos funcionales y no funcionales, casos de uso y objetos del negocio, atributos de calidad).

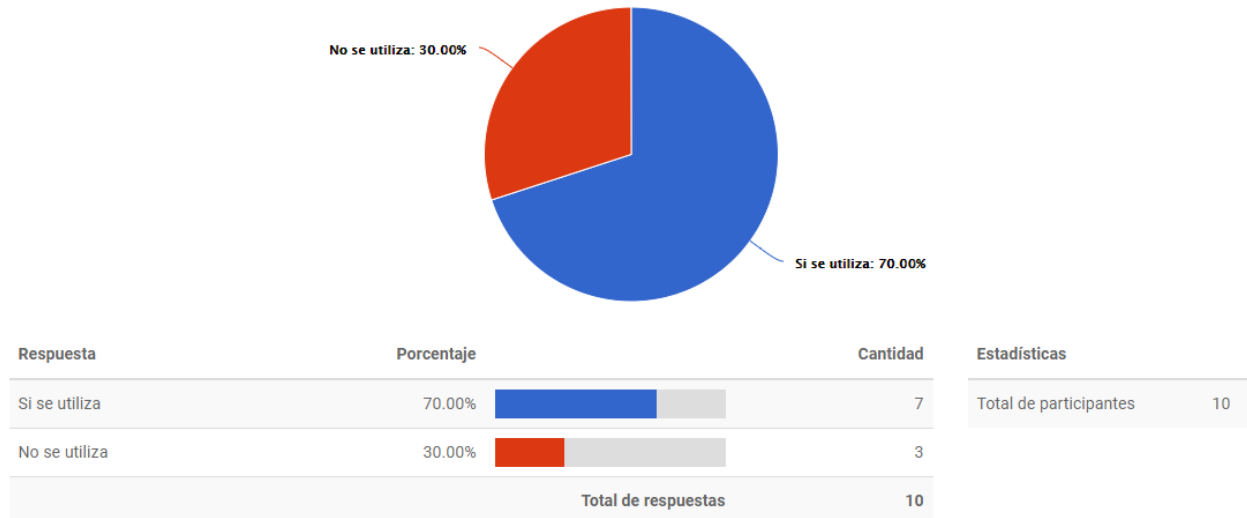


Respuesta	Porcentaje	Cantidad	Estadísticas
Si se utiliza	70.00%	7	Total de participantes 10
No se utiliza	30.00%	3	
		Total de respuestas	10

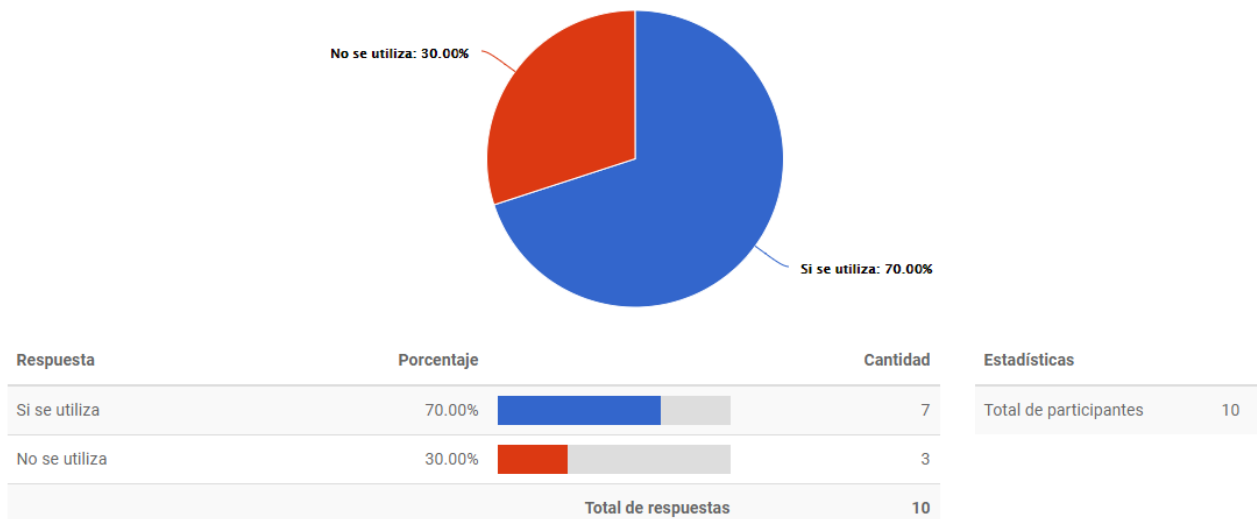
2. Documentar los requisitos de forma correcta (completos, consistentes, dentro del alcance del proyecto, testeables) en un lenguaje natural (descripción no técnica) y al mismo tiempo en lenguaje de requisitos formal utilizando reglas de especificación de requisitos (eliminar todos los pronombres de la especificación de requisitos, sustituyéndolos por los sujetos).



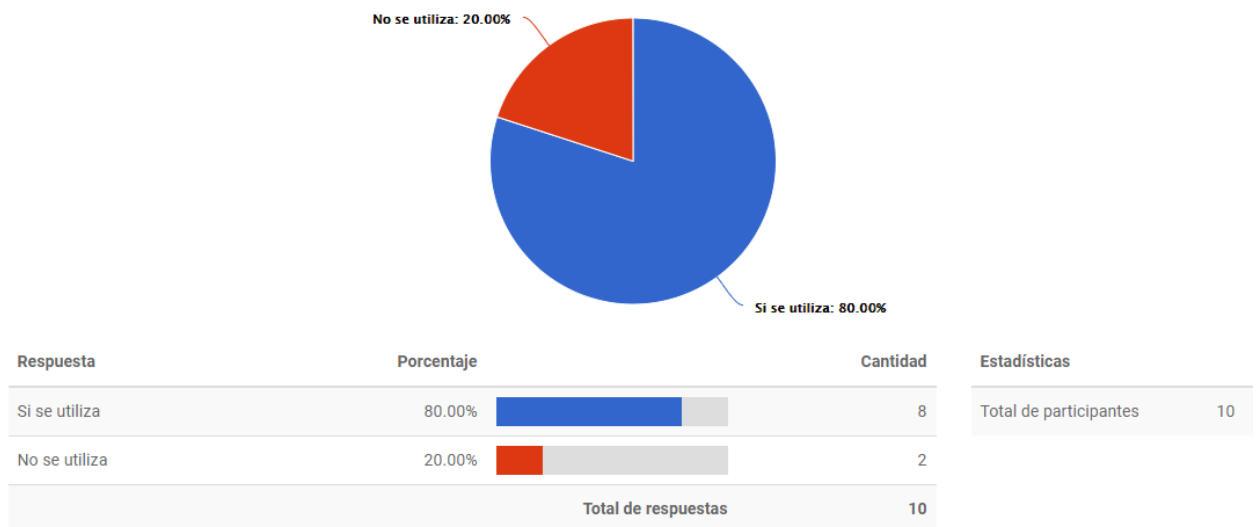
3. Definir los requisitos teniendo en cuenta la perspectiva del usuario.



4. Adoptar plantillas de documentos de requisitos (visión y alcance, casos de uso, especificación de requisitos - SRS).



5. Escribir especificación de requisitos no ambigua (cada requisito debe ser comprensible por sí solo).



11 Explicación del experto (razones de la no utilización o argumentaciones adicionales que justifique la utilización de las buenas prácticas anteriores):

ENTREVISTADO	RESPUESTA
1	<p>Pregunta 1. El documento permite el traslado de conocimiento ya sea técnico o lógico lo que mejora el tiempo en la curva de aprendizaje, en los miembros del equipo de desarrollo.</p> <p>Pregunta 2. Se debe definir entre las áreas del negocio y del equipo de desarrollo un lenguaje en común que permita un solo entendimiento en la documentación.</p> <p>Pregunta 3. Siempre que se aborde a las personas con un buen conocimiento y manejo del negocio, de lo contrario se es habitual lograr un funcionamiento que no es el esperado por los usuarios.</p> <p>Pregunta 5. El equipo del proyecto debe definir los objetivos a desarrollar y tener muy clara las funcionalidades.</p>
2	En mi experiencia estas buenas prácticas se utilizan todas.
3	<p>Pregunta 1. Crear artefactos, donde trabajo actualmente si se hace, y los pasa el arquitecto. En mi experiencia es utilizada principalmente en las medianas empresas donde hay roles mejor definidos.</p> <p>Pregunta 2. Se utiliza pero no siempre se definen correctamente los requisitos, pero si se hace, lo que permite tener una base para negociar con el cliente respecto a los requisitos solicitado, es muy importante definir como se realizara la documentación.</p> <p>Pregunta 3. Definir los requisitos teniendo en cuenta la perspectiva del usuario. A veces pienso que no, porque en el único caso que se utiliza es cuando se hace una migración porque el usuario ya tiene una forma de hacer las cosas, debido a que el usuario ya tiene experiencia previa en el software, cuando es software nuevo es una nueva experiencia para el usuario y no siempre tiene bien definido lo que necesita o como debería de funcionar. Pero al final si hay q tomar en cuenta al usuario porque él ha realizado la solicitud y él va a usar el software.</p> <p>Pregunta 4. Adoptar plantillas de documentos de requisitos si es utilizable totalmente ya que esto permite que se reutilizan estos formatos definidos como estandarizados para los proyectos.</p> <p>Pregunta 5. Escribir especificación de requisitos no ambigua, en este caso en la empresa donde laboro se utiliza una segunda opinión de otra persona para identificar que el requisito este completo o que quede claro.</p>
4	<p>Pregunta 1. Esta tarea la debería de llevar un analista de sistemas y el desarrollador para crear el documento funcional y el de especificaciones técnicas por medio de la retroalimentación entre ambas personas.</p>

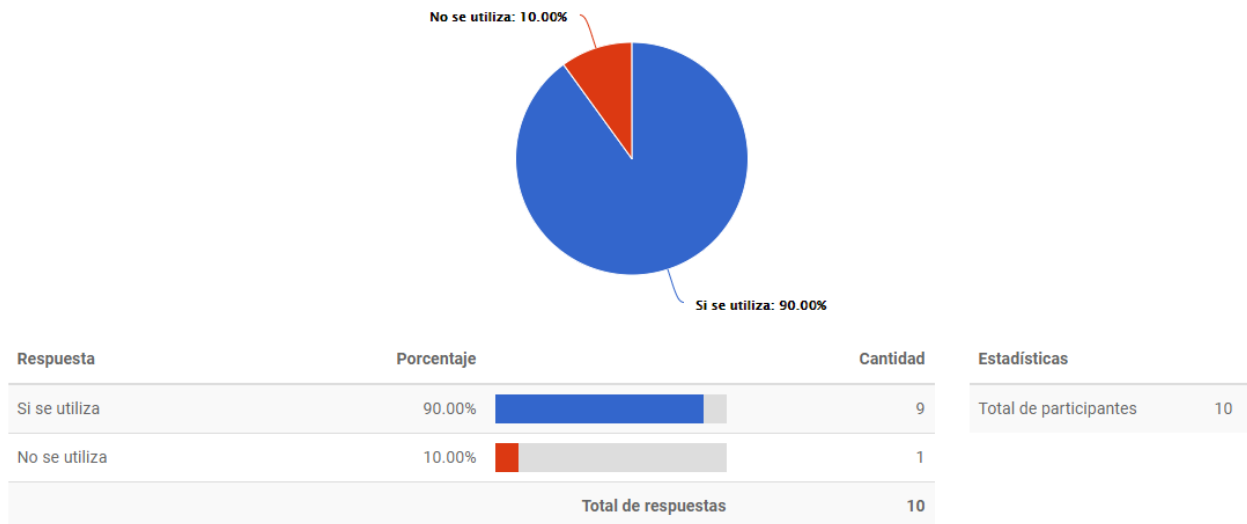
UN FRAMEWORK DE BUENAS PRÁCTICAS PARA LOS REQUISITOS DE SOFTWARE

	<p>Pregunta 2. Comúnmente lo realiza un analista de sistema antes de trasladarlo al equipo de desarrollo. Y este se encarga de retroalimentarlo con el líder de proceso de negocio (usuario) y en el caso de requerirlo con los desarrolladores.</p> <p>Pregunta 3. A veces el usuario se queda corto en lo que solicita, y es el programador al que le toca proponer funcionalidades para mejorar el producto de software. Esta buena práctica depende de que tanto conoce y maneja el negocio el usuario.</p>
5	<p>Pregunta 1. En la mayoría de proyectos la documentación existe pero se mantiene desactualizada (no tienen la información completa) o muchas veces simplemente no se realiza por el factor tiempo.</p> <p>En lo personal considero que esta documentación ayuda mucho cuando se realiza, por ejemplo: en los casos cuando se da rotación del personal, futuros cambios en el software, implementación de buenas prácticas de reutilización. Siendo de gran beneficio ya que volvería las actividades antes mencionadas más eficientes.</p> <p>Pregunta 4. El tener plantillas de documentos de requisitos proporciona los siguientes beneficios: Agiliza en la búsqueda de información, cambios en los requisitos, fácil entendimiento.</p> <p>Pregunta 5. Para cumplir con esta buena práctica debe mantenerse a la persona que levanta los requisitos de inicio a fin del proyecto, ya que en mi experiencia el cambiarla crea ambigüedades es decir contradicciones con la nueva persona, también hay que asegurarse del buen nivel técnico y conocimiento del negocio para levantar buenos requisitos.</p>
6	<p>Pregunta 3. En la medida de lo posible debe plantearse los requisitos teniendo en cuenta la perspectiva de la empresa como primera instancia. Esto debido a que en ocasiones el usuario solamente conoce la porción del negocio que él maneja, es necesario realizar las investigaciones pertinentes para determinar si el requisito solo está abarcando la porción específica del usuario o si hay un problema mayor de fondo.</p>
7	<p>Pregunta 1. En algunas empresas documentar no es trabajo de valor quieren ver a la gente programando, gente que ejecute y este lineamiento que viene heredado de un mal liderazgo de las jefaturas.</p> <p>Todos los puntos anteriores son indispensables para la definición de un producto, la ausencia de alguno de estos elementos supone no tener una definición del producto por lo que generar una espiral infinita de inconformidades con respecto al requisito de negocio.</p>
8	<p>Las especificaciones antes mencionadas en mi rubro no cobran mayor realce ya que no hay tiempo para la documentación, únicamente se tiene en cuenta lo que el usuario puede necesitar pero sin llegar a colocar eso en plantillas, documentos. Este punto no se realiza por que en mi rubro (Servicios y Logística) el tiempo de entrega es muy reducido dejando esto en segundo plano.</p>

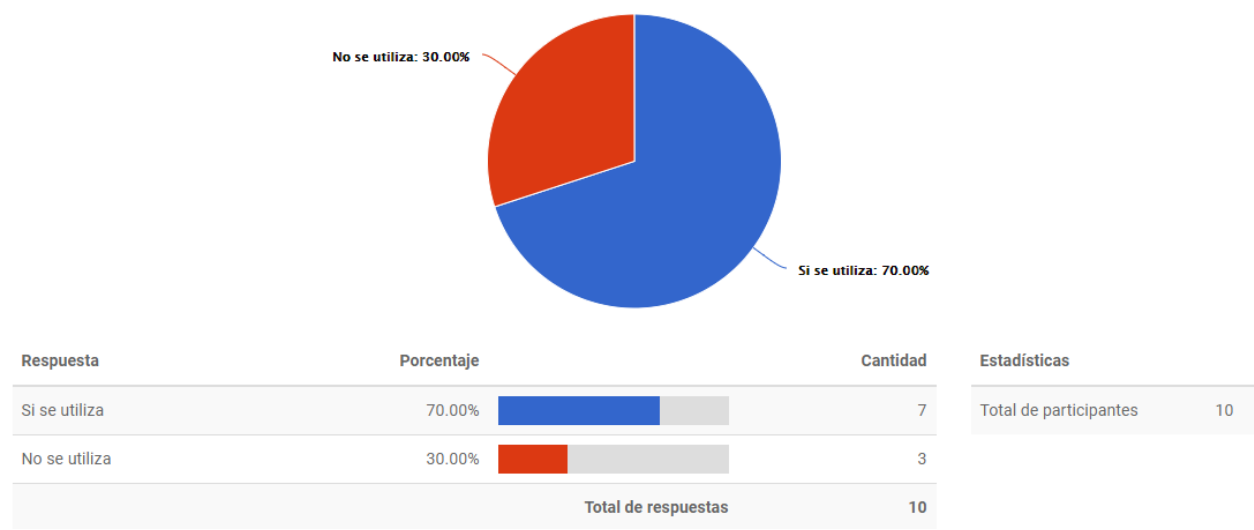
9	<p>Pregunta 1: En mi caso los documentos existen pero se mantienen desactualizados o con falta de información.</p> <p>Pregunta 3: Si se utiliza por medio de las historias de usuario, estas se validan con el usuario y son ellos los que dan el aval para desarrollarlos.</p> <p>Pregunta 5: si se utiliza pero en mi experiencia cambia la perspectiva o resultados obtenidos, lo cual depende del conocimiento técnico y negocio que posea la persona a cargo. Para tener mejores resultados y subsanar la debilidad anterior debe de realizarse como una necesidad el intercambio o revisión de los requisitos de software y de la documentación en general con otras personas para que estos queden claros y se entienda lo mismo.</p>
10	Culturalmente las empresas no comprenden que el desarrollo de software debe tener procedimientos claros y bien definidos, no solo en su construcción sino en la definición de los requisitos; y que requiere mucho tiempo e involucramiento del usuario solicitante.
Total de respuesta	10

12 Según su experiencia, valide la utilización de las buenas prácticas de VALIDACIÓN de requisitos siguientes:

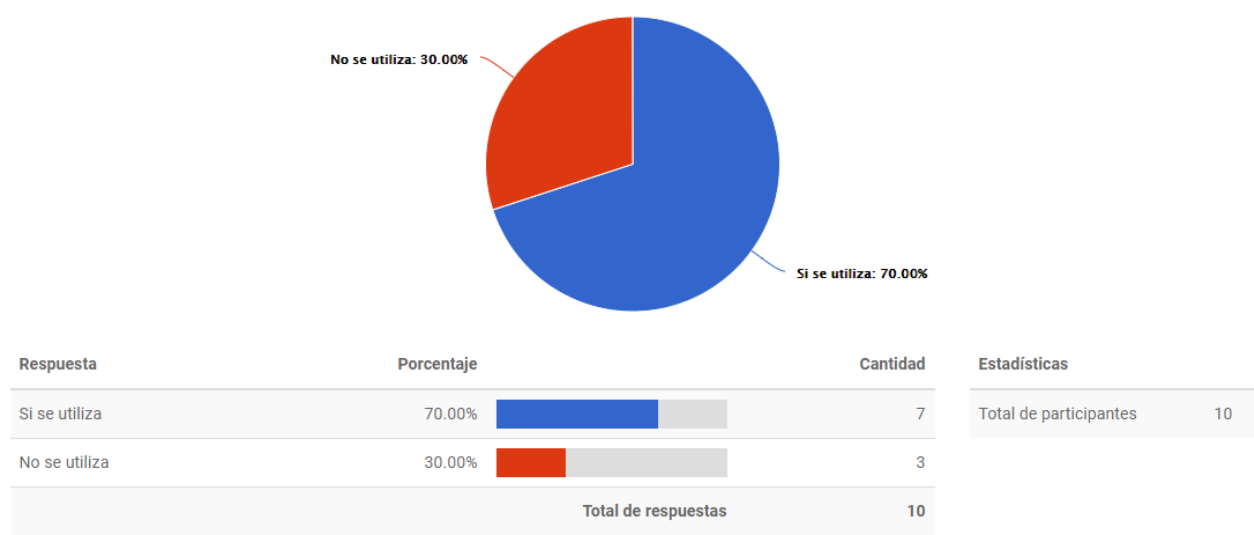
1. Establecer para cada requisito criterios de validación y aceptación de las partes interesadas.



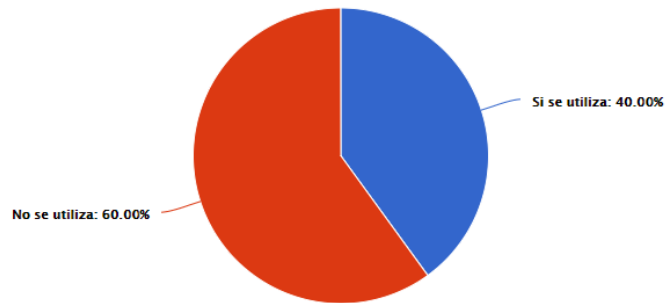
2. Validar la documentación de los requisitos funcionales y técnicos en su contenido, documentación y acuerdos.



3. Analizar y validar los requisitos (trazabilidad, supuestos válidos, esenciales y consistentes con el diseño).

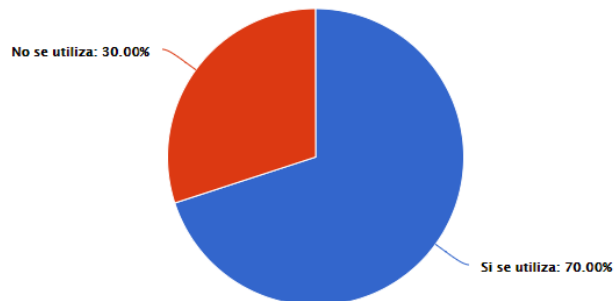


4. Revisar los requisitos y utilizar inspecciones de requisitos formales tanto con usuarios como con proveedores.



Respuesta	Porcentaje	Cantidad	Estadísticas
Si se utiliza	40.00%	4	Total de participantes 10
No se utiliza	60.00%	6	
Total de respuestas			10

5. Utilización de prototipos para validar la interpretación de los requisitos de software (las características clave de las nuevas aplicaciones) y para obtener nuevos requisitos.



Respuesta	Porcentaje	Cantidad	Estadísticas
Si se utiliza	70.00%	7	Total de participantes 10
No se utiliza	30.00%	3	
Total de respuestas			10

13 Explicación del experto (razones de la no utilización o argumentaciones adicionales que justifique la utilización de las buenas prácticas anteriores):

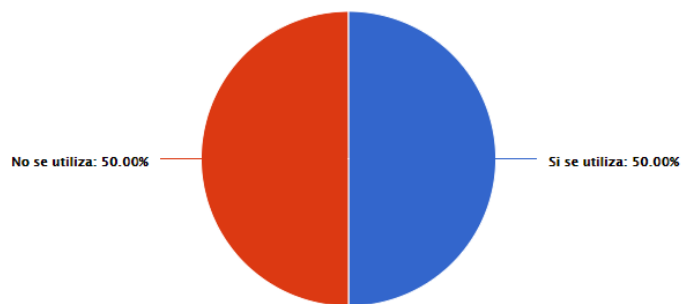
ENTREVISTADO	RESPUESTA
1	<p>Pregunta 1. En la utilización de esta buena práctica se define un tiempo de revisión, en el que se permite realizar comentarios u observaciones para mejorar los requisitos.</p> <p>Pregunta 3. Los diseños y patrones deben ser uniformes con el diseño para que no sean mal interpretados por los usuarios.</p> <p>Pregunta 5. Muy utilizada ya que permite definir una idea del funcionamiento que tendrá el software por parte de todos los involucrados.</p>
2	<p>Pregunta 1. Si se utiliza, pero hay porcentajes de en el ciclo de vida del desarrollo donde no existen criterios de aceptación, esto sucede cuando la persona encargada del negocio no toma la seriedad del caso y/o que hay que sacar el desarrollo a contra tiempo.</p> <p>Pregunta 2. Esta buena práctica no es muy utilizada por falta de recursos. Otro de los factores es cuando una sola persona del equipo técnico conoce y se adueña del requisito.</p> <p>Pregunta 4. Culturalmente en El Salvador se tiene la costumbre del desarrollo a prueba y error.</p> <p>Pregunta 5. Esta buena práctica es de gran ayuda porque confirma que se está haciendo lo que en verdad proporcionara valor al usuario.</p>
3	<p>Pregunta 4. Por mi experiencia si se revisa con el usuario, se encuentran errores a tiempo y sirve mucho antes de pasar al desarrollo, igualmente al realizar inspecciones formales se logra identificar los errores rápidamente en la documentación.</p> <p>Pregunta 5. Si se utiliza y es de gran apoyo resultando en la espera del feedback por parte de los involucrados para aplicar los cambios eficazmente.</p>
4	<p>Normalmente se deja de lado la documentación o se inicia con ella pero a medida avanza el proceso de desarrollo, se deja de actualizar. Esto no solo es a nivel de El Salvador sino a nivel latinoamericano.</p>
5	<p>Pregunta 4. Por lo general la revisión de los requisitos solo se realiza con un encargado de la empresa o representante del negocio lo que se produce quedan fuera algunas partes involucradas, pero ya son directrices de los líderes de la organización.</p> <p>Pregunta 5. Muchas veces no se utiliza por economizar tiempo y costos, dando como consecuencia que los nuevos requisitos aparezcan hasta en la presentación de avances del desarrollo o entregas de módulos.</p>
6	<p>Pregunta 1. Este punto lo realizan los de calidad de software (QA), es utilizado cuando no se puede llevar acabo alguna funcionalidad por razones técnicas, por lo que permite buscar soluciones alternativas.</p>

	<p>Pregunta 2. Función propiamente de los QA, ya que son los que validan el requisito con el desarrollo realizado.</p> <p>Pregunta 3. La mayoría de lugares heredan las metodologías establecidas con anterioridad, por lo que se respeta esa forma de trabajo. Esto limita la innovación o la flexibilidad técnica para el desarrollo, es decir que no se pueden implementar herramientas nuevas o actualizar metodologías. Otro factor es que las empresas se encuentran atadas a proveedores que proporcionan los Core del negocio, por lo que se está sujeto a alineamientos que ellos proporcionen.</p> <p>Pregunta 4. No se realiza y el no utilizarlo conlleva a reprocesos (tiempo y mayor utilización de recursos).</p> <p>Pregunta 5. Esta práctica es muy utilizada por los analista de sistemas, sirve para retroalimentar los requisitos permitiendo obtener nuevas características de las necesidades del usuario. Y detallarlas de cómo será lograda en la funcionalidad.</p>
7	<p>Pregunta 5. No siempre es necesario establecer un prototipo, esto está en función de la complejidad de la solución.</p>
8	<p>Pregunta 1. En mi experiencia la parte de la validación no es una parte que se desarrolle en la empresa en las empresas salvadoreñas, por lo que al no tener una documentación no podemos validar los requisitos de manera óptima, por lo que únicamente se deja el desarrollo con el criterio del usuario final o partes interesadas, estas últimas en ocasiones no aportan mayor beneficio al software porque ellos dejan esa función exclusiva a IT.</p>
9	<p>Pregunta 2. Es importante tener toda la documentación en regla y respaldada como evidencia para futuros conflictos en el cumplimiento de los requisitos. No solamente el usuario involucrado debe validar la documentación, también su jefe superior.</p> <p>Pregunta 5. En la medida de lo posible deben utilizarse programas de simulación para que el usuario pueda corroborar si se está trabajando en la línea correcta según sus expectativas.</p>
10	<p>Pregunta 1 y 2. Se hace un documento de aceptación de los requisitos y sobre esos se trabaja, hasta que estos están establecidos y validados se procede al desarrollo mientras no se inicia.</p> <p>Pregunta 3. Se hace con el documento de controles de cambio para manejar la trazabilidad, supuestos válidos, para mantener una buena trazabilidad. Si el requisito no aplica se rechaza y se documenta el rechazo, debería darse siempre en todos los proyectos pero en la práctica ya se han dado casos que no se da, por el factor tiempo y no contar con una buena planificación.</p> <p>Pregunta 4. No se hace, en la práctica el requisito se hace hasta que esta QA haciendo pruebas y esto no se utiliza en la práctica y si el requisito no está bien planteado requiere volverlo a rehacer.</p>

	Pregunta 5. En el documento de requisitos queda el acuerdo de la fecha del primer demo, para que el usuario vea funcionando lo que ha pedido porque muchas veces aplica funcionalidad, diseño, como un preliminar de lo que el cliente ha solicitado y a partir de allí formular entregables para que el cliente vea avances del software que ha solicitado.
Total de respuestas	10

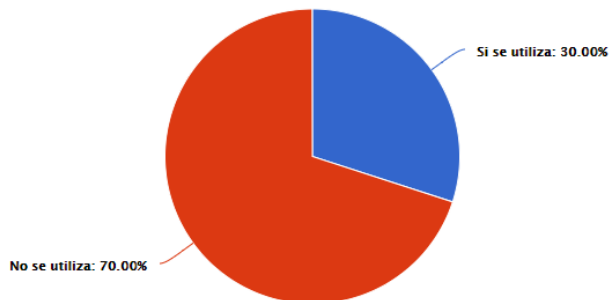
14 Según su experiencia, valide la no utilización de las buenas prácticas de GESTIÓN de requisitos siguientes:

1. Gestionar de forma efectiva y eficiente el control de cambios de los requisitos (introducción, cambios y eliminación), utilizando un plan y herramientas de configuración y control de cambios.



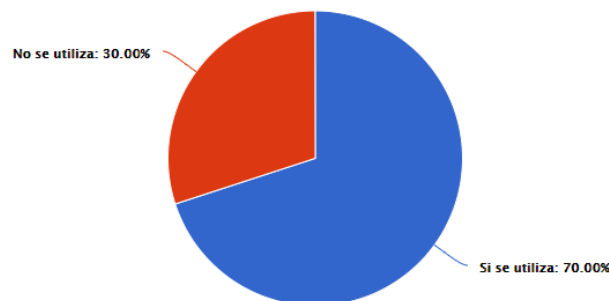
Respuesta	Porcentaje	Cantidad	Estadísticas
Si se utiliza	50.00%	5	Total de participantes 10
No se utiliza	50.00%	5	
Total de respuestas		10	

2. Llevar el seguimiento de la vida de un requisito, utilizando la técnica de matriz de trazabilidad de requisitos.



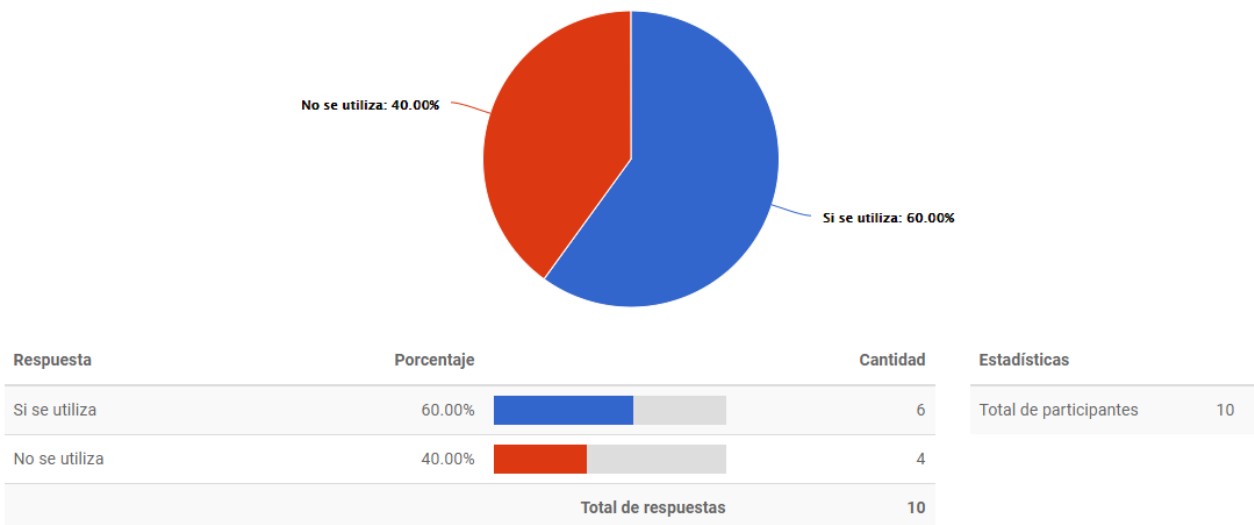
Respuesta	Porcentaje	Cantidad	Estadísticas
Si se utiliza	30.00%	3	Total de participantes 10
No se utiliza	70.00%	7	
Total de respuestas			10

3. Establecer la línea base de los requisitos para asegurar que cualquier modificación en los requisitos que cambien la línea base se trate como cambios de alcance.

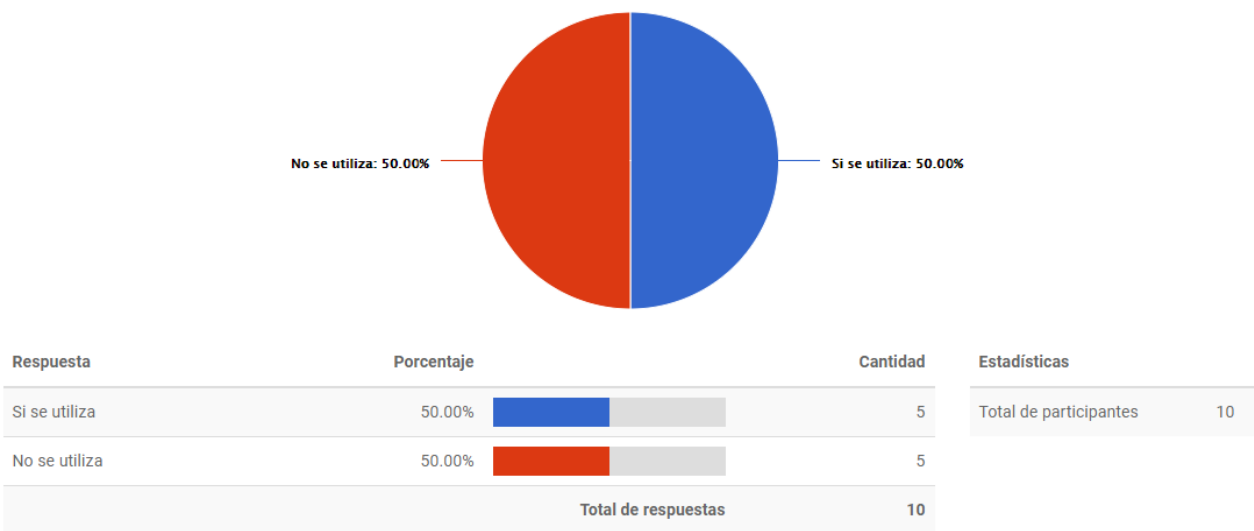


Respuesta	Porcentaje	Cantidad	Estadísticas
Si se utiliza	70.00%	7	Total de participantes 10
No se utiliza	30.00%	3	
Total de respuestas			10

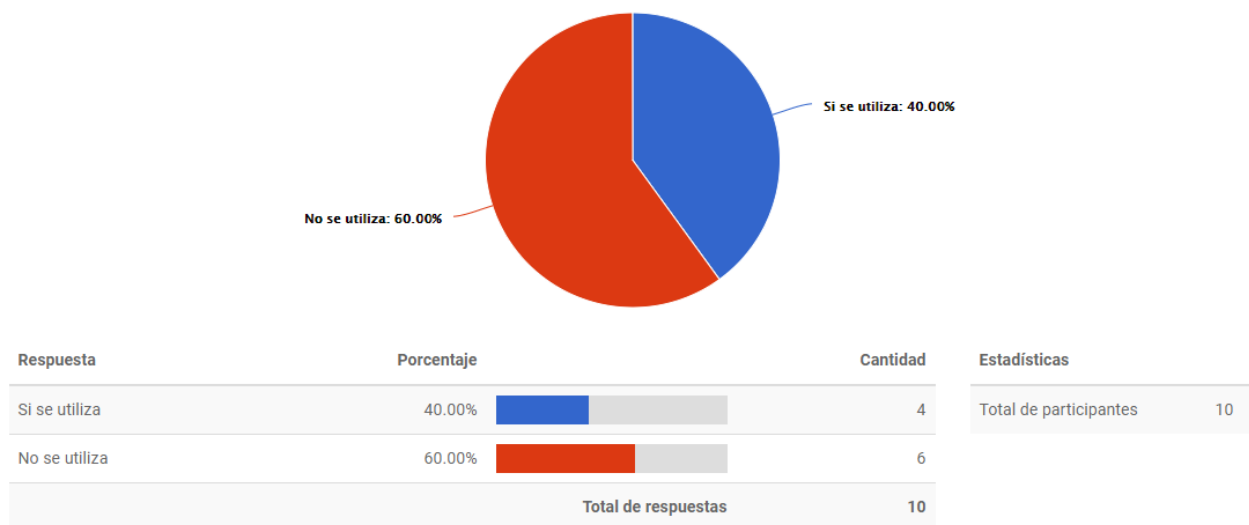
4. Supervisar y dar seguimiento al estado de los requisitos de software (especificado, verificado, analizado, entre otros) mediante un proceso definido.



5. Medir el número y la gravedad de los defectos en los requisitos definidos.



6. Formar a los analistas de requisitos para asegurar que tienen el conocimiento, entre otros aspectos, de cómo escribir y gestionar buenos requisitos.



15 Explicación del experto (razones de la no utilización o argumentaciones adicionales que justifique la utilización de las buenas prácticas anteriores):

ENTREVISTADO	RESPUESTA
1	Es importante que se cree un departamento o área especializada y preparada para la toma, análisis y gestión de los requerimientos. La calidad de los análisis dependerá de tres factores principales: tiempo invertido de los usuarios involucrados, conocimiento del negocio y capacidad de aprendizaje del mismo analista de requisitos y tiempo invertido en el ciclo de análisis del requisito.
2	Esta parte no es utilizada en la empresa, como anteriormente se dijo el tiempo y falta de personal hace que este punto no sea factible. Para definir si un requisito no aplica queda a criterio del desarrollador este hace las observaciones necesarios y es el quien decide que aplica y no para el desarrollo, además de ser el encargado de gestionar y supervisar.
3	<p>Pregunta 3. En cuanto a los cambios de alcance, también es común que intervengan directores o incluso presidentes de las empresas y soliciten o autoricen cambios que modifican el alcance creyendo que son de impacto mínimo cuando en realidad representan un gran esfuerzo adicional para los equipos de desarrollo. Ésto es producto de no tener una metodología formal y una forma sistemática y certera de medir los impactos.</p> <p>Pregunta 6. Normalmente no se capacita adecuadamente al personal de tecnologías de información. No deberían de pensar los analistas de sistemas que en internet se encuentran</p>

	<p>todas las respuestas. Comúnmente se maneja una frase popular "Google es tu amigo", haciendo referencia precisamente a este hecho.</p>
4	<p>Pregunta 1. No se aplican debido a la falta de utilización de herramientas por el alto costo de inversión. Esto se debe a que la administración no visualiza el beneficio directo de estas. Estas permiten verificar y llevar un mejor control con el desarrollo, son de gran ayuda ya que permiten identificar las últimas versiones, regresar a versiones anteriores del software, realizar migraciones con menores riesgos y nuevas implementaciones. Todo esto se traduce en eficiencia en la administración, lo que puede realizarse en horas por hacerlo manual o no llevar el control se traduce a semanas.</p> <p>Pregunta 2. Sirve para llevar el control de los proyectos, sacar porcentajes de avance y toma de decisiones para implementar cambios (nuevos requisitos por ejemplo del tipo legal, normativo en otras palabras urgentes) también utilizados para distribuir los recursos del equipo de desarrollo.</p> <p>Pregunta 3. En mi experiencia se tratan como mejora al producto de software, por ende se trasladan a la cola del proyecto. Solo que sea critico se introducen, es decir cambios por un ente regulador.</p> <p>Pregunta 4. Utilizado para buscar un compromiso con la especificación del usuario final, cualquier cambio debe ser autorizado tanto por el equipo de desarrollo como los líderes definidos del negocio con el fin de evitar supuestos no válidos y conflictos entre las partes. Un procedimiento formal quizá no se tenga documentado no existe, pero este puede darse por un correo electrónico, una reunión, entre otros.</p> <p>Pregunta 5. Esta actividad la realizan los analista de sistemas, por medio de un versionamiento del requisito de software, pero hay ausentismo de la gravedad ya que hay desarrollos que han llegado hasta producción y es ahí donde ya se da la relevancia necesaria.</p> <p>Pregunta 6. Esto se realiza por medio de un plan de capacitaciones, hay cursos para enseñar a levantar los requisitos funcionales y no funcionales, estos se enfocan en como hablar con los usuarios y como se deben llenar las secciones de los documentos de requisitos de software definidos en la empresa. Es decir para cada sección deben llevar un enfoque específico.</p>
5	<p>Pregunta 1. Dependiendo de la dimensión del proyecto hay que evaluar el nivel de controles y documentaciones, por ejemplo si hay una situación regla y es un riesgo a la empresa, aun cuando no agrega valor a la empresa si hay que documentar al mayor grado posible. Otro ejemplo es el tema de lavado de dinero, aun cuando la empresa sea pequeña es un tema critico que requiere de controles, documentación y herramientas que permitan un mayor respaldo.</p> <p>Pregunta 2. En una empresa mediana como en la que laboro, la estructura de la organización no permite que cada quien asuma un rol, hacer esto no permite un desarrollo ágil, porque se solicita un desarrollo rápido que se logre cumplir en tiempo. Los stakeholders no asumen su responsabilidad y refieren con otra persona, ni siquiera ellos asumen sus roles y dejan la</p>

	<p>responsabilidad a TI el de volverse analistas de negocio, con el riesgo que no se asimile bien el proceso.</p> <p>Pregunta 3. Si se hace, un poco apegado a scrum, las historias de usuario son la línea base tomando en cuenta dependencia, criticidad y valor agregado.</p> <p>Pregunta 4. Si es utilizada, cada necesidad del negocio se vuelve para nosotros un caso de negocio o requerimiento de alto nivel, y a través de eso le damos seguimiento al estatus, mas no al estatus de cada requisito a nivel de detalle.</p> <p>Pregunta 5. No se utiliza, por lo general el usuario no sabe lo que quiera, no hay un requisito que se aplique al 100%, desde que iniciamos sabes que nos está diciendo un 80% o 60% de lo que necesita, no nos agrega valor al proceso porque estamos más enfocados a entregar rápido que de calidad.</p> <p>Pregunta 6. Las empresas medianas y pequeñas no le dan relevancia a capacitar en este tipo de roles, la administración tradicional asume que el desarrollador es capaz de desarrollar todo eso empíricamente. El problema en esta parte, es que los financieros tienen demasiado peso en estos temas, la administración tradicional desmerece las necesidades que informática tiene, no reconocen la necesidad de tener un scrum master por dar un ejemplo. Dicho de otra forma una analogía es pensar que "el mismo arquitecto puede pegar ladrillos". La tendencia de las empresas es delegar responsabilidades, tal vez una empresa que se dedique a desarrollar software (fábrica de software) valore un poco más, pero las empresas que no son de desarrollo no lo hacen, no tienen la visión de que informática es un área estratégica para su negocio y buscan delegar las responsabilidades ante proyectos tan caros como suelen ser los de informática, en algunos casos los mismos jefes o gerentes de informática no son personas con conocimientos en administración de ciclo de vida de aplicaciones.</p>
6	<p>Pregunta 1. Por mi experiencia les puedo decir que existen muchos software que permiten llevar un buen control, por ejemplo con la metodología SCRUM, pero en la práctica no se hace por desconocimiento.</p> <p>Pregunta 2. No se hace, no se utiliza por desconocimiento, la mejor forma seria usar las herramientas que ayudan a hacer esto.</p> <p>Pregunta 3. Si se establece y se utiliza porque va relacionado a la negociación, costo y tiempo, el problema surge cuando ya hay cambios, porque producen problemas de trazabilidad y seguimiento.</p> <p>Pregunta 4. No se utiliza, hasta que se termina el software o avance se valida, supervisan y no se define estados ni se controlan. La mayoría de empresas concuerda que contratar a alguien especializado para eso es costoso, por ejemplo llámese Project manager u otro, en las empresas una persona hace todo y por lo que no puede llevar el control de todo por cuestiones de cumplimiento del tiempo establecido.</p> <p>Pregunta 5. No se hace, hemos tenido experiencias, donde se ha establecido un tiempo de dos meses y surgen cambios y no se modifica el tiempo de entrega y eso implica más trabajo,</p>

	<p>quedarse más horas que tiene que ver con las solicitudes de cambio, quienes pagan y culpan siempre de los errores son los programadores, pero esto puede originarse del desorden o no estandarización de los procesos de negocio de las organizaciones.</p> <p>Pregunta 6. Los profesionales que se dedican a este tipo de tareas no han tenido una formación formal, van a prendiendo en la marcha de forma empírica es decir de los errores del pasado y por supuesto que no es la mejor forma. Pero el problema es que las empresas no quieren invertir en contratar a alguien especializado o formar el recurso.</p>
7	<p>Pregunta 1. No se hace a conciencia el proceso y por ende hay ausencia de documentación.</p> <p>Pregunta 2: No siempre se utiliza la matriz de trazabilidad para ciertos requisitos.</p> <p>Pregunta 3: Aunque si existe los requisitos base, cuando nacen otros, se les prioriza sobre los primeros, según la necesidad del negocio (entidades reguladoras, cambios en la forma de hacer negocios, etc.)</p> <p>Pregunta 5: Se puede decir que aunque se anuncien los problemas nunca se llevó el control para no repetirlos de nuevo.</p> <p>Pregunta 6: Cuando he trabajado con especialistas en análisis de negocio y usuarios que conocen los procesos he visto buenos resultados ya que estos ayudan a captar los requisitos de manera más eficiente.</p>
8	<p>Pregunta 3. Siempre y cuando se apegue a la metodología de desarrollo utilizada.</p> <p>Pregunta 4. Esto permite pasar a la siguiente fase teniendo la autorización y visto bueno de las áreas involucradas</p> <p>Pregunta 5. El número de defectos indicara que tan clara esta la idea o funcionamiento de los requisitos solicitados.</p> <p>Pregunta 6 - La formación tiene que ser en ambos sentidos, negocio y desarrollo.</p>
9	<p>Pregunta 2. No se utiliza el llevar en una matriz de trazabilidad los requisitos, sino hasta la entrega de avances, se ve si se logra el requerimiento.</p>
10	<p>Pregunta 1. Se pasan los desarrollos a producción prometiendo que la documentación o el control se realizaran posteriormente. A veces por la urgencia del negocio y la costumbre de la mala práctica de desarrollar en ambientes de producción.</p> <p>Pregunta 2. No se realiza porque no se le da la importancia necesaria.</p> <p>Pregunta 3. Muchas veces se meten cambios durante el tiempo establecido de los requisitos a desarrollarse, estos son requisitos ocultos o necesidades urgentes del negocio por lo que no se respeta la línea base definida.</p> <p>Pregunta 4. En mi experiencia de debe hacer pero realizándolo a conciencia, sino no funciona.</p> <p>Pregunta 5. Serviría para no volver a cometer los mismos errores. Esto se realiza de manera breve en las reuniones de seguimientos semanales en una metodología ágil, ejemplo SCRUM pero no se le da importancia a ese seguimiento.</p>

UN FRAMEWORK DE BUENAS PRÁCTICAS PARA LOS REQUISITOS DE SOFTWARE

	<p>Pregunta 6. Acá deben capacitar a la persona para conocer el negocio. Ayudaría para que la persona despeje al equipo de desarrollo cualquier duda en el funcionamiento del negocio y técnico. Escribiría de forma correcta y adecuada cada requisito para que lo entiendan los del negocio y equipos del desarrollo. Evitaría retrasos en el proyecto ya que sería un buen apoyo para validar la funcionalidad y esto al no hacerlo causa atrasos de tiempos, lo que se traduce en costos desarrollando algo no funcional para el negocio.</p>
Total de respuestas	10