

8. INICIAR

Este capítulo incluye los procesos relacionados con la iniciación de un proyecto: *Create Project Vision*, *Identify Scrum Master and Stakeholder(s)*, *Form Scrum Team*, *Develop Epic(s)*, *Create Prioritized Product Backlog*, y *Conduct Release Planning*.

Initiate, tal como se define en la *Una guía para el conocimiento de Scrum (Guía SBOK™)*, es aplicable a los siguientes:

- *Portfolio, programs y/o projects* de cualquier sector
- *Products*, servicios o cualquier otro resultado que se entregarán a los *Stakeholders*
- *Projects* de cualquier tamaño y complejidad

El término "*product*" en la *Guía SBOK™* puede referirse a un producto, servicio, o cualquier otra prestación. Scrum se puede aplicar de manera efectiva a cualquier proyecto en cualquier industria-desde pequeños proyectos o equipos con tan sólo seis miembros del equipo, hasta proyectos grandes y complejos que cuentan con cientos de miembros por equipo.

A fin de facilitar la mejor aplicación del marco de Scrum, en este capítulo se identifican las entradas, herramientas y salidas de cada proceso, ya sea como "obligatoria" u "opcional". Las entradas, herramientas y salidas indicadas por asteriscos (*) son obligatorias, mientras que las que no tienen asteriscos son opcionales.

Se recomienda que el *Scrum Team* y aquellas personas que recién inician el aprendizaje sobre el marco y los procesos Scrum se centren principalmente en las aportaciones obligatorias, las herramientas y los productos; mientras que los *Product Owners*, *Scrum Masters*, y otros practicantes de Scrum con ya más experiencia sobre éste se esfuercen por alcanzar un conocimiento más profundo de la información de todo el capítulo. También es importante darse cuenta de que, aunque todos los procesos se definen de forma única en la *Guía SBOK™*, no se lleva a cabo de forma secuencial o por separado necesariamente. A veces, puede ser más apropiado combinar algunos procesos, dependiendo de los requisitos específicos de cada proyecto.

Este capítulo está escrito desde la perspectiva de un *Scrum Team* que está trabajando en un *Sprint* para producir *Entregables (Deliverables)* como parte de un proyecto más amplio. Sin embargo, la información que se describe es igualmente aplicable a proyectos completos, programas y *portfolios*. Información adicional relacionada con el uso de Scrum para proyectos (projects), programas (programs) y portafolios (*portfolios*) está disponible desde el capítulo 2 hasta el 7, los que cubren los principios y aspectos de Scrum.

La figura 8-1 proporciona una visión general de los procesos de *Initiate phase*, que son los siguientes:

8.1 Create Project Vision—En este proceso, el *Project Business Case* es revisado para crear un *Project Vision Statement* que servirá de inspiración y proporcionará un enfoque de todo el proyecto. El *Product Owner* se identifica en este proceso.

8.2 Identificar al Scrum Master y al/a los stakeholder(s)—En este proceso, el *Scrum Master* y los *stakeholders* se identifican utilizando criterios de selección específicos.

8.3 Form Scrum Team—En este proceso, se seleccionan a los miembros del *Scrum Team*. Normalmente, el *Product Owner* es el responsable principal de la selección de los miembros del equipo, pero a menudo lo hace en *collaboration* con el *Scrum Team*.

8.4 Develop Epic(s)—En este proceso, el *Project Vision Statement* sirve como la base para el desarrollo de *epics*. *User Group Meetings* pueden tomar lugar para discutir el/los *Epic(s)* apropiado(s).

8.5 Create Prioritized Product Backlog—En este proceso, los *Epic(s)* son refinados, elaborados, y luego priorizados para crear el *Prioritized Product Backlog* del proyecto. Lo que se conoce como *Done Criteria* también se establece en este punto.

8.6 Conduct Release Planning—En este proceso, el *Scrum Core Team* revisa los *User Stories* en el *Prioritized Product Backlog* para desarrollar un *Release Planning Schedule*, que es esencialmente un programa de implementación por fases que se puede compartir con los *stakeholders* del proyecto. Los *Length of Sprints* también se determinan en este proceso.

Figura 8-1: Información general sobre *Initiate*

Nota: Los asteriscos (*) denotan una entrada, herramientas o salida "obligatoria" para el proceso correspondiente.

La figura 8-2 muestra las entradas obligatorias, herramientas y salidas para los procesos en *Initiate Phase*.

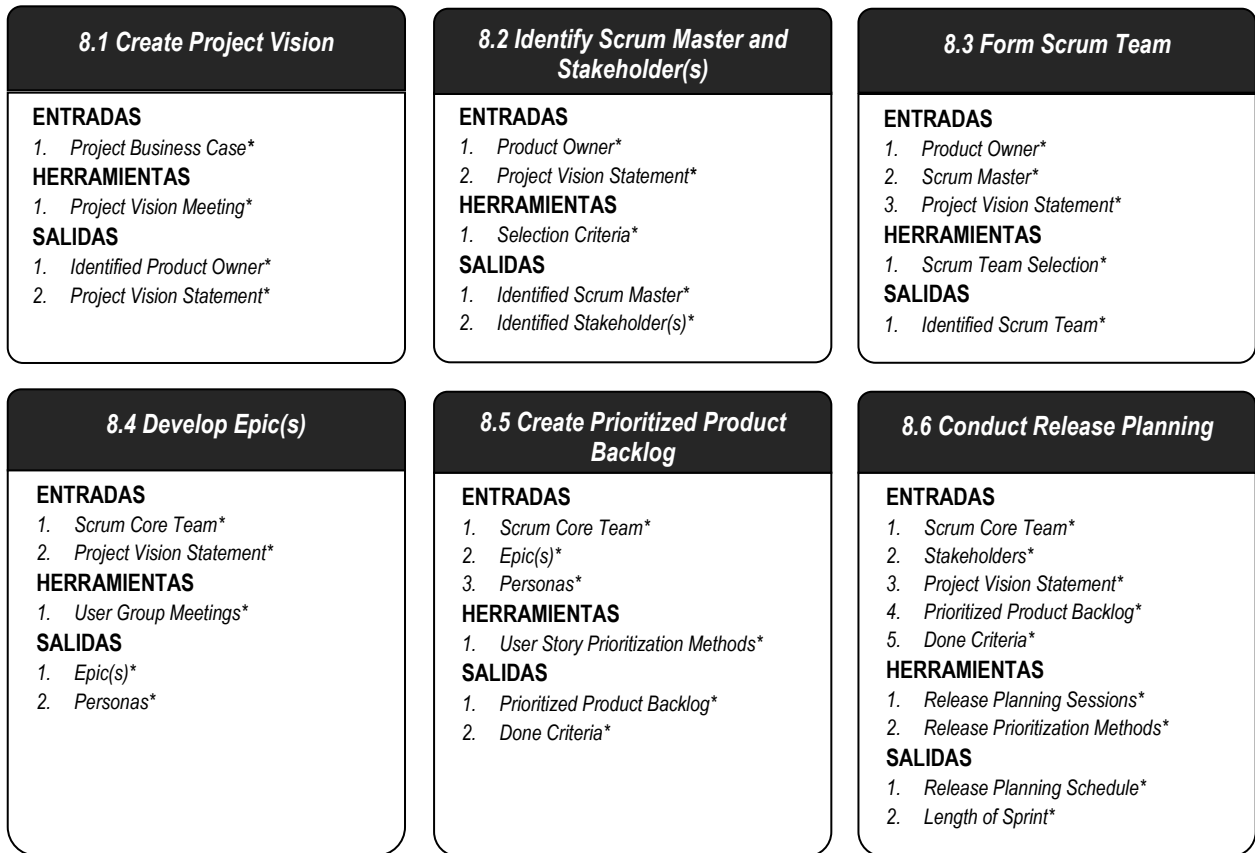


Figura 8-2: Información general de *Initiate* (Esenciales)

Nota: Los asteriscos (*) denotan una entrada, herramienta o salida "obligatoria" para el proceso correspondiente.

8.1 Create Project Vision

La Figura 8-3 muestra todas las entradas, las herramientas y las salidas para el proceso de *Create Project Vision*.

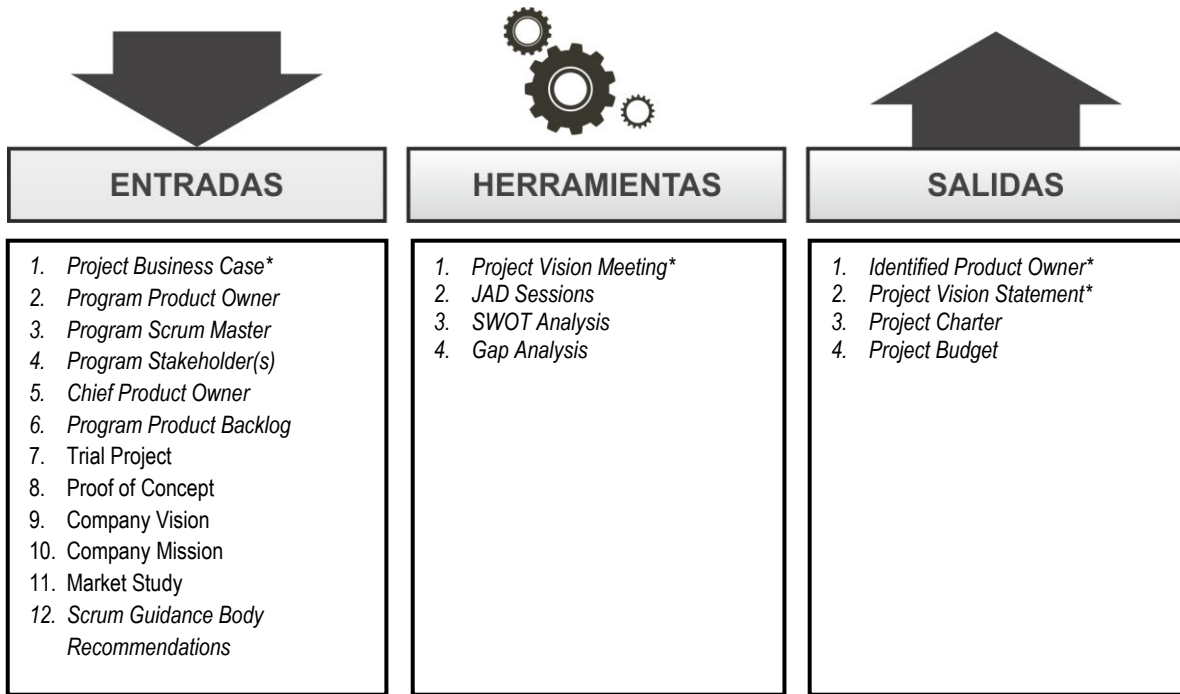
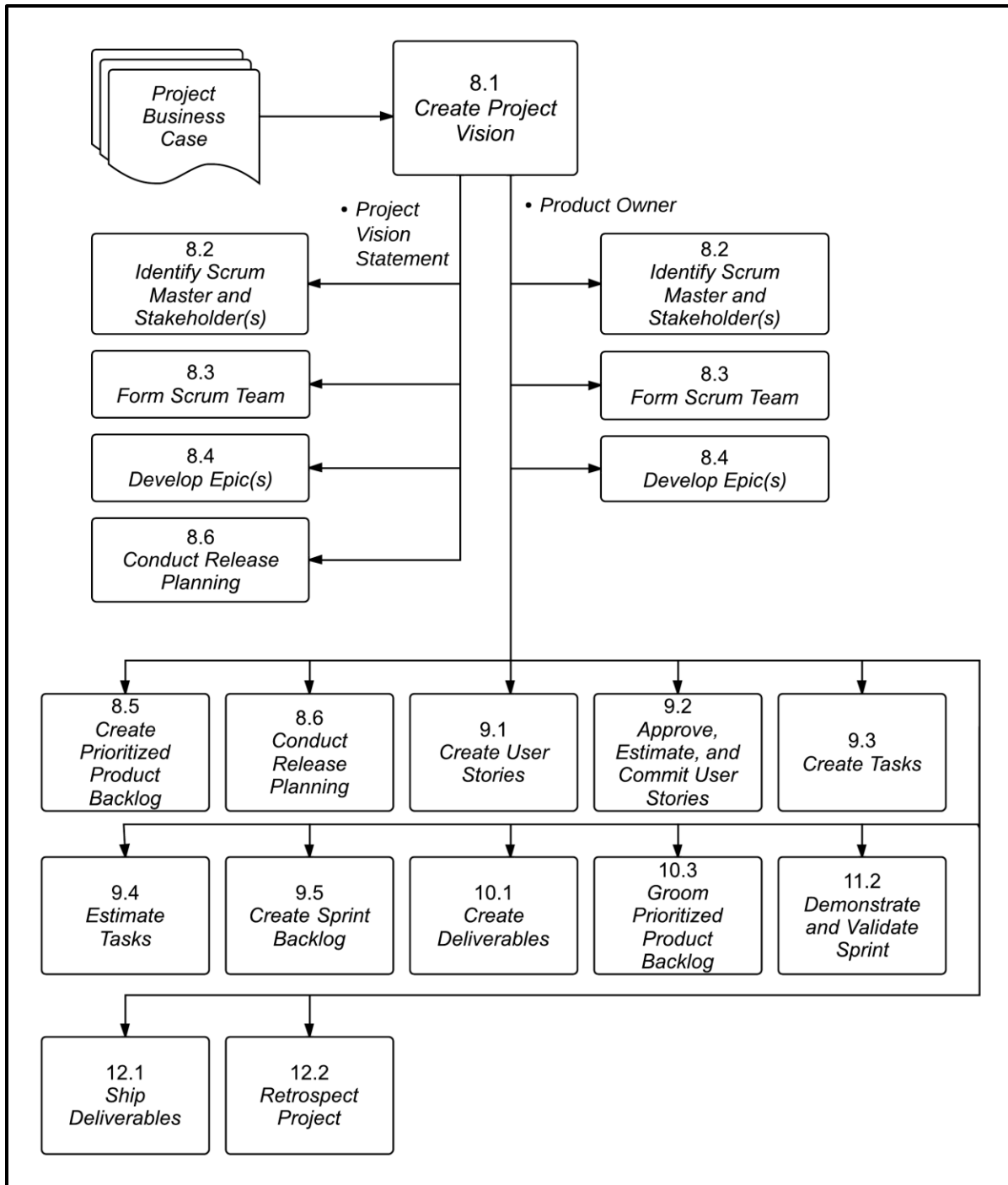


Figura 8-3: *Create Project Vision* – Entradas, Herramientas, y Salidas

Nota: Los asteriscos (*) denotan una entrada, herramienta o salida "obligatoria" para el proceso correspondiente.

Figura 8-4: El diagrama de flujos de datos del *Create Project Vision*

8.1.1 Entradas

8.1.1.1 *Project Business Case**

Un caso de negocio (*business case*) puede ser un documento bien estructurado o simplemente una declaración verbal que expresa la razón para iniciar un proyecto. Puede ser formal y completo, o informal y breve. Independientemente del formato, a menudo incluye información sustancial sobre los antecedentes del proyecto, los objetivos del negocio y los resultados deseados, un *SWOT* y *Gap Analysis*, una lista de los *risks* identificados, y las estimaciones de tiempo, el esfuerzo y costo.

El proyecto se inicia con la presentación del *Project Business Case*. Un caso de negocio se le presenta a los *stakeholders* y patrocinadores (*sponsors*). Los *stakeholders* así comprenden los beneficios de negocio esperados de tal proyecto y los patrocinadores confirman que van a proporcionar los recursos financieros para el proyecto.

8.1.1.2 *Program Product Owner*

8

El *Program Product Owner* es la persona responsable de maximizar el valor del negocio para un programa. Él/ella es responsable de la articulación de requisitos de los *customers* y de mantener *business justification* para el *program* y puede aportar de forma importante sobre qué proyectos deben ser implementados, y de que manera, en un *program*. El *Program Product Owner* también administra el *Program Product Backlog*.

El *Program Product Owner* interface con el *Portfolio Program Owner* para asegurar la alineación del programa con las metas y objetivos del *portfolio*. Él/ella también tiene que ver con el nombramiento de los *Product Owners* para los proyectos individuales, y debe asegurar que la visión, los objetivos, los resultados, y las liberaciones de los proyectos individuales en el programa estén alineados con los del programa.

8.1.1.3 *Program Scrum Master*

El *Program Scrum Master* es un facilitador que asegura que todos los equipos del proyecto en el programa dispongan de un entorno propicio para completar con éxito sus proyectos. El *Program Scrum Master* guía, facilita y les enseña las prácticas de Scrum a todos los que participan en el programa; también sirve como guía para los *Scrum Masters* de los distintos proyectos; elimina los *impediments* de los diferentes equipos del proyecto; coordina con el *Scrum Guidance Body* para definir los objetivos relacionados con la normativa de calidad, el gobierno, la seguridad y otros parámetros claves de la organización; y asegura que los procesos de Scrum se estén siguiendo de manera efectiva a lo largo del programa.

El *Program Scrum Master* debe interactuar con el *Portfolio Scrum Master* para asegurar la alineación del *program* con las metas y objetivos del *portfolio*. También tiene que ver con el nombramiento del *Scrum Masters* para proyectos individuales y debe asegurar que la visión, los objetivos, los resultados, y las liberaciones de los proyectos individuales en el programa estén alineados con los del programa.

8.1.1.4 **Program Stakeholder(s)**

Program Stakeholder(s) es un término colectivo que incluye a los *customers*, los usuarios y patrocinadores para un programa. Influyen en todos los proyectos del programa durante todo el desarrollo del proyecto. El/Los *Program Stakeholder(s)* también puede(n) ayudar a definir la visión del proyecto y a proporcionar orientación en relación con el valor del negocio.

El/Los *Program Stakeholder(s)* debe(n) interactuar con los *Portfolio Stakeholders* para asegurar la alineación del programa con las metas y objetivos del *portfolio*. Ellos también están involucrados con el nombramiento del/de los *Stakeholder(s)* para proyectos individuales y se aseguran que la visión, los objetivos, los resultados, y los lanzamientos de los proyectos individuales en el programa estén alineados con los del programa.

8.1.1.5 **Chief Product Owner**

En el caso de grandes proyectos con numerosos *Scrum Teams*, podría ser necesario tener un *Chief Product Owner*. Esta función se encarga de coordinar el trabajo de los múltiples *Product Owners*. El *Chief Product Owner* prepara y mantiene el *Prioritized Product Backlog* para los proyectos grandes, usándolo para coordinar el trabajo a través de los *Product Owners* de los *Scrum Teams*. Los *Product Owners*, a su vez, gestionan sus respectivas partes del *Prioritized Product Backlog*.

El *Chief Product Owner* también se comunica con el *Program Product Owner* para asegurar la alineación de los proyectos grandes con las metas y objetivos del programa.

8.1.1.6 **Program Product Backlog**

El *Program Product Owner* desarrolla el *Program Product Backlog* del producto que contiene una lista de prioridades de negocios de alto nivel y los requisitos del proyecto escritos preferiblemente en forma de *Program Backlog Items* del programa. Luego, estos son refinados por los *Product Owners* de los proyectos individuales, ya que crean y dan prioridad a pedidos pendientes del producto para sus proyectos. Estos *Prioritized Product Backlogs* tienen *User Stories* más pequeños pero detallados que pueden ser aprobados, estimados, y aplicados por miembros del *Scrum Team*.

El *Program Product Owner* mantiene de forma continua el *Program Product Backlog* para garantizar que los nuevos *business requirements* se añadan y que los requisitos existentes estén debidamente documentados y priorizados. Esto asegura que a los requisitos más valiosos en el cumplimiento de los objetivos del programa se les esté dando alta prioridad y que los restantes reciban una prioridad más baja.

El *Program Product Backlog* creado para el programa presenta una imagen más grande de todos los proyectos que forman parte del programa. Por lo tanto, puede servir de orientación en relación con los objetivos del proyecto, el alcance, los objetivos y los beneficios esperados del negocio.

8.1.1.7 Trial Project

Si es factible, un demo o prueba del proyecto en pequeña escala podría ser ejecutado como un experimento para predecir y evaluar la viabilidad, tiempo, costo, *risks*, y los posibles efectos del proyecto. Esto ayuda a evaluar el entorno de práctica y guías del diseño del proyecto con anterioridad a la iniciación del proyecto.

8.1.1.8 Proof of Concept

Proof of Concept demuestra y verifica que la idea detrás del proyecto actual es potencialmente viable en el entorno real. A menudo esto se hace en la forma de un prototipo diseñado para determinar la viabilidad técnica y financiera, ayudar a comprender los requisitos, y ayudar en la evaluación de las decisiones de diseño al principio del proceso. Sin embargo, *Proof of Concept* no necesita representar necesariamente los verdaderos *Deliverables* (Entregables) del proyecto.

8.1.1.9 Company Vision

La comprensión del *Company Vision* ayuda a que el proyecto mantenga su enfoque en los objetivos de la organización y el futuro probable de la empresa. El *Product Owner* se puede guiar por el *Company Vision* para crear el *Project Vision Statement*.

8.1.1.10 Company Mission

Company Mission ofrece un marco para la formulación de las estrategias de la empresa y orienta la toma de decisiones en general en la empresa. *Project Vision* debe enmarcarse de tal manera que su cumplimiento ayuda a la organización a llevar a cabo su misión.

8.1.1.11 Market Study

Market Study se refiere a la investigación organizada, la recopilación, la comparación y el análisis de datos relacionados con las preferencias de los *customers* sobre los productos. A menudo incluye numerosos datos sobre las tendencias del mercado, la segmentación del mercado y los procesos de comercialización. Estudio de mercado podría incluir también un estudio analítico de los competidores que proporciona una mejor comprensión de las fortalezas y debilidades de los competidores y puede ayudar a los que toman decisiones a formular productos mejor posicionados.

8.1.1.12 Recomendaciones del *Scrum Guidance Body*

El *Scrum Guidance Body* (SGB) es una función opcional. Por lo general, se compone de un grupo de documentos y/o un grupo de expertos que normalmente están involucrados en la definición de objetivos relacionados con la calidad, las regulaciones gubernamentales, la seguridad y otros parámetros claves de la organización. Estos objetivos guían la labor llevada a cabo por el *Product Owner*, *Scrum Master*, y el *Scrum Team*. El *Scrum Guidance Body* también ayuda a capturar las mejores prácticas que se deben utilizar en todos los proyectos Scrum en la organización.

El *Scrum Guidance Body* no toma decisiones relacionadas con el proyecto. Este actúa como una consultoría o una estructura de orientación para todos los niveles de jerarquía en el proyecto de organización del *portfolio*, *program* y *project*. Los *Scrum Teams* tienen la opción de pedirle asesoramiento al *Scrum Guidance Body*.

Es importante asegurarse de que la visión del proyecto esté alineada con las recomendaciones proporcionadas por el *Scrum Guidance Body* y que los procesos cumplan con las normas y directrices establecidas por el *Body* (la Administración).

8.1.2 Herramientas

8.1.2.1 *Project Vision Meeting**

Project Vision Meeting es una reunión con el/los *Program Stakeholder(s)*, *Program Product Owner*, *Program Scrum Master*, y *Chief Product Owner*. Ayuda a identificar el contexto empresarial, *business requirements* y las expectativas de los *stakeholders* con el fin de desarrollar un *Project Vision Statement* eficaz. Scrum cree en la participación y colaboración cercana con todos los representantes de las empresas para obtener su *buy-in* (convencimiento de su importancia) del proyecto y para ofrecer un valor más significativo.

8.1.2.2 *JAD Sessions*

Joint Application Design (JAD) Session es una técnica de recopilación de requisitos. Se trata de un taller facilitado altamente estructurado que acelera el proceso de *Create Project Vision*, ya que permite al/a los *stakeholder(s)* y a otros que toman decisiones llegar a un consenso sobre el alcance, los objetivos, y otras especificaciones del proyecto.

Esta técnica consiste en métodos para aumentar la participación del usuario, lo que acelera el desarrollo y la mejora de las especificaciones. Los *Relevant Program Stakeholder(s)*, *Program Product Owner*, *Program Scrum Master* y *Chief Product Owner* podrían reunirse para delinear y analizar los resultados de negocio deseados y visualizar su visión para el proyecto Scrum.

8.1.2.3 SWOT Analysis

SWOT es un enfoque estructurado para la planificación que ayuda a evaluar los/las *Strengths*, *Weaknesses*, *Opportunities* y *Threats* (*puntos fuertes y débiles, oportunidades y amenazas*) relacionados con un proyecto. Este tipo de análisis ayuda a identificar tanto los factores internos como externos que podrían afectar el proyecto. Las fortalezas y debilidades son factores internos, mientras que *opportunities* y *amenazas* son factores externos. La identificación de estos factores ayuda a los *stakeholders* y a aquellos que toman decisiones a finalizar los procesos, las herramientas y las técnicas que se utilizarán para lograr los objetivos del proyecto. La realización de un SWOT permite la identificación precoz de las prioridades, los cambios potenciales, y los *risks*.

8.1.2.4 Gap Analysis

Gap Analysis es una técnica que se utiliza para comparar el estado actual con el estado deseado. En una organización, se trata de determinar y documentar la diferencia entre las capacidades actuales del negocio y el conjunto final deseado de capacidades. Normalmente, se inicia un proyecto para que una organización alcance una situación deseada, por lo que llevar a cabo un *Gap analysis* ayudaría a quienes toman decisiones a determinar la necesidad de un proyecto.

Los principales pasos a seguir en *Gap Analysis* se presentan en la Figura 8-5.

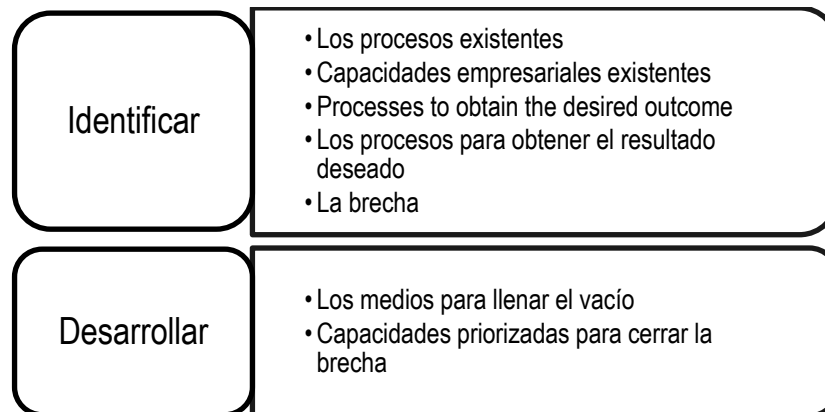


Figura 8-5: El proceso de *Gap Analysis*

8.1.3 Salidas

8.1.3.1 Identified Product Owner*

Uno de los resultados de este proceso es identificar al *Product Owner*. El *Product Owner* es la persona responsable de lograr el valor máximo empresarial para el proyecto. Él o ella también es responsable de la articulación de requisitos por parte de los *customers* y de mantener *business justification* para el proyecto. El *Product Owner* representa la voz del *customer*.

Cada *Scrum Team* tendrá un *product owner* designado. Un pequeño proyecto puede tener sólo un *Product Owner*, mientras que los proyectos más grandes pueden tener varios. Estos *Product Owners* son responsables de la gestión de sus secciones del *Prioritized Product Backlog*. Ellos escriben los *User Stories* y gestionan el mantenimiento del *Prioritized Product Backlog*.

El papel del *Product Owner* se describe con más detalle en la sección 3.4.

8.1.3.2 ***Project Vision Statement****

El resultado clave del proceso *Create Project Vision* es un *Project Vision Statement* bien estructurado. Una buena visión del proyecto explica la necesidad del negocio, y que es lo que el proyecto tiene como objetivo satisfacer, en lugar de cómo se va a satisfacer la necesidad.

El *Project Vision statement* no debería ser muy específico y debe ser flexible. Es posible que el conocimiento actual sobre el proyecto esté basado en suposiciones que luego vayan a cambiar conforme avanza el proyecto, por lo que es importante que la visión del proyecto sea lo suficientemente flexible como para adaptarse a estos cambios. La visión del proyecto debe centrarse en el problema y no la solución.

Ejemplo:

VMFoods, una cadena de supermercados, quiere ampliar con un portal de comercio electrónico en línea y se ha comunicado con su firma para crear el producto.

Project Vision: Desarrollar una herramienta fácil de usar y estéticamente agradable canal de ventas en línea para VMFoods.

8.1.3.3 ***Project Charter***

Un *Project Charter* es una declaración oficial de los objetivos y resultados deseados del proyecto. En varias organizaciones, el *Project Charter* es el documento que oficialmente y formalmente autoriza el proyecto, dándole al equipo la autoridad por escrita para comenzar el proyecto.

8.1.3.4 ***Project Budget***

El *Project Budget* es un documento financiero que incluye el costo de las personas, materiales y otros gastos relacionados en un proyecto. El *Project Budget* típicamente es firmado por el/los patrocinador(es) para asegurar que haya suficientes fondos disponibles. Una vez firmado, el *Product Owner* y el *Scrum Master* estarán regularmente envueltos con la gestión del *Project Budget* para garantizar que las personas y los recursos necesarios para las actividades del proyecto estén disponibles.

8.2 Identify Scrum Master and Stakeholder(s)

La Figura 8-6 muestra todas las entradas, las herramientas y las salidas para el proceso *Identify Scrum Master and Stakeholder(s)*

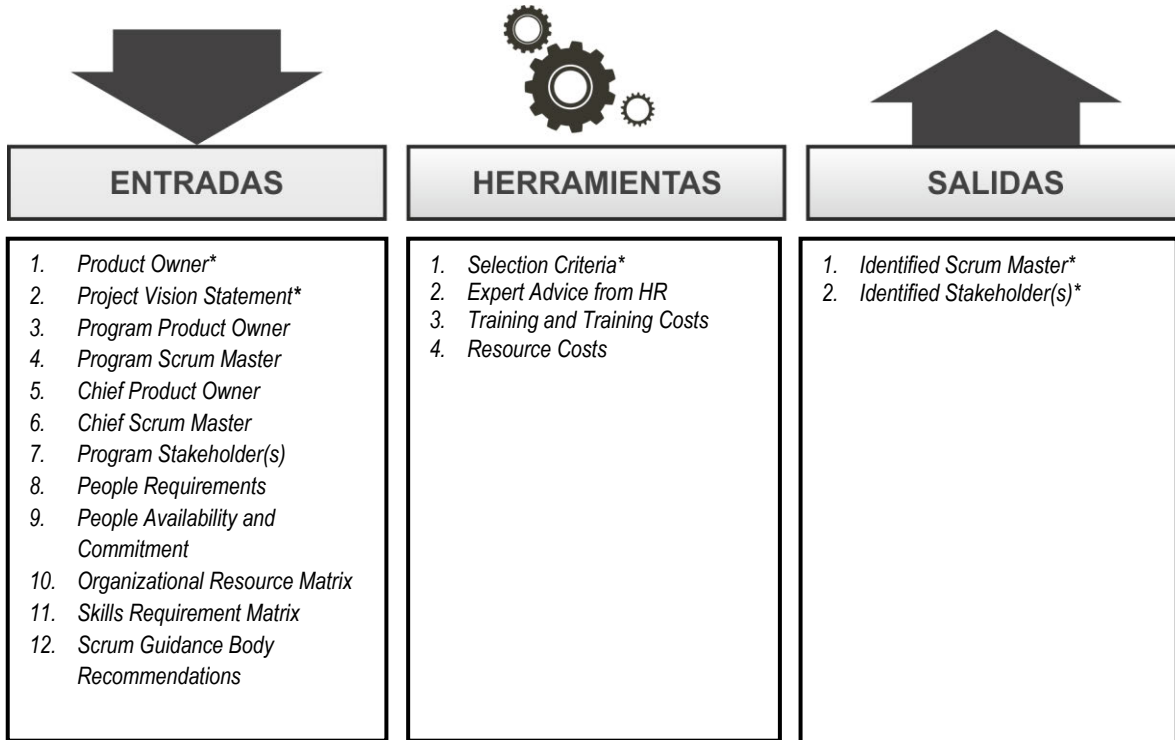


Figura 8-6: Identify Scrum Master and Stakeholder(s) —Entradas, Herramientas, y Salidas

Nota: Los asteriscos () denotan una entrada, herramientas o salida "obligatoria" para el proceso correspondiente.*

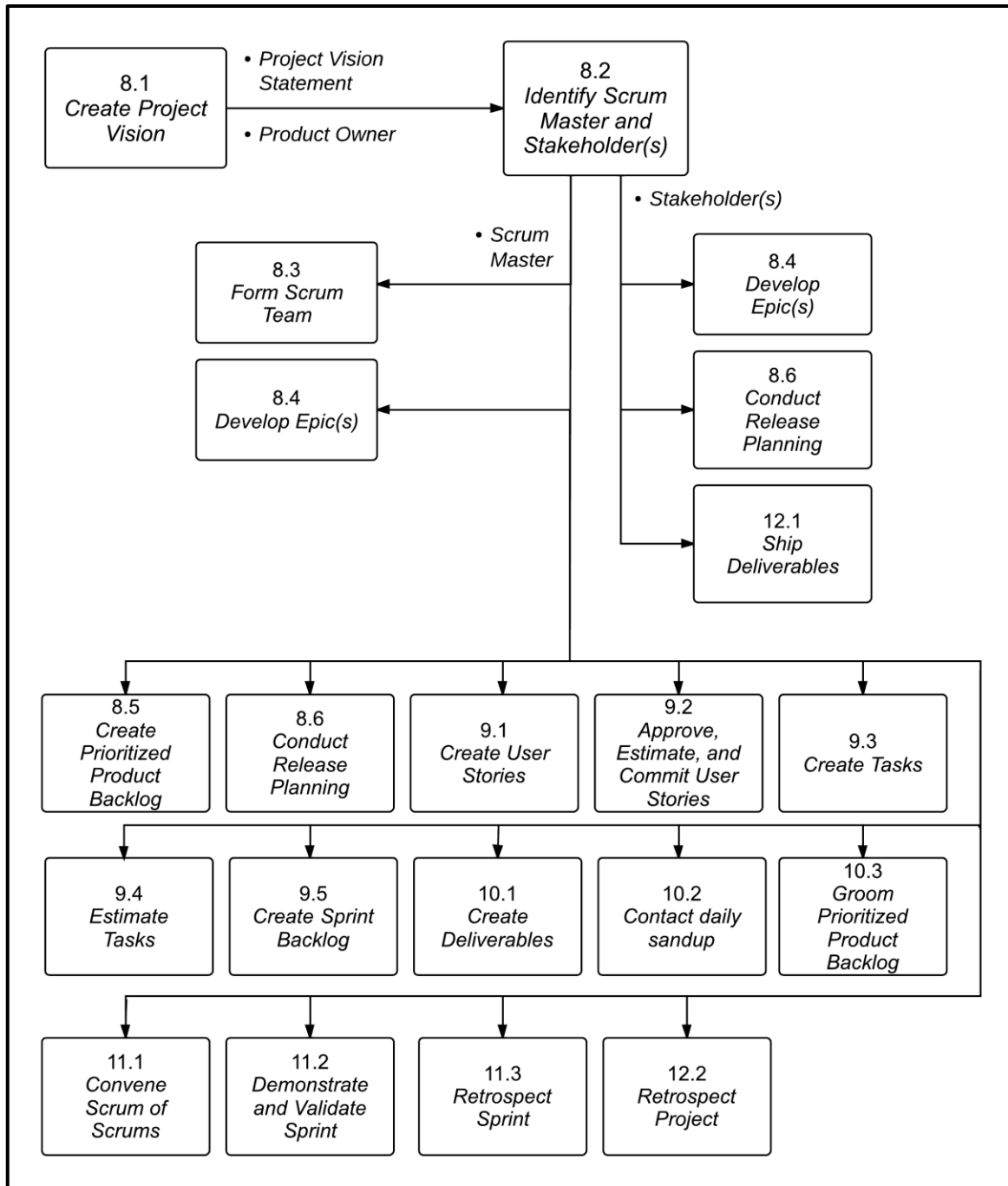


Figura 8-7: *Identify Scrum Master and Stakeholder(s)*—Diagrama de flujo de datos

8.2.1 Entradas

8.2.1.1 *Product Owner**

Descrito en la sección 8.1.3.1.

8.2.1.2 *Project Vision Statement**

Descrito en la sección 8.1.3.2.

8.2.1.3 *Program Product Owner*

Descrito en la sección 8.1.1.2.

8.2.1.4 *Program Scrum Master*

Descrito en la sección 8.1.1.3.

8.2.1.5 *Chief Product Owner*

Descrito en la sección 8.1.1.5.

8.2.1.6 *Chief Scrum Master*

Los grandes proyectos requieren que múltiples *Scrum Masters* trabajen en paralelo. Es posible que la información obtenida por un equipo se le deba comunicar adecuadamente a los otros equipos – es el *Chief Scrum Master* el responsable de esta actividad.

La coordinación entre distintos *Scrum Teams* que colaboran en un proyecto se realiza normalmente a través del *Scrum of Scrums Meeting* (SoS) (véase la sección 3.7.2.1). Esto es análogo al *Daily Standup Meeting* y es facilitado por el *Chief Scrum Master*. El *Chief Scrum Master* típicamente es el responsable de atender los *impediments* que más afectan a un *Scrum Team*.

8.2.1.7 **Program Stakeholder(s)**

Descrito en la sección 8.1.1.4.

8.2.1.8 **People Requirements**

People Requirements es uno de los pasos iniciales en la selección del *Scrum Master* y *Stakeholder(s)*. Es importante documentar las funciones y responsabilidades de todos los que se verían involucrados en la realización de las tareas del proyecto. Esto incluye a todas las personas involucradas en el proyecto, en cualquier calidad, independientemente de que su papel sea básico o no.

Por lo general, el *Product Owner* o el *Scrum Master* colaboran con el *Human Resource Department* de la empresa para determinar y concluir los *People Requirements*.

8.2.1.9 **Personas Disponibilidad y Compromiso**

Antes de seleccionar al *Scrum Master* y al/ a los *stakeholder(s)*, se debe confirmar su disponibilidad. Sólo los miembros del equipo que estarán disponibles y que puedan comprometerse plenamente con el proyecto deben ser seleccionados. En *People Availability and Commitment* es representado comúnmente en forma de calendario que muestra cuando los recursos humanos estarán disponibles para trabajar durante toda la duración del proyecto.

Para ser eficaz, los *Scrum Teams* idealmente deberían tener de seis a diez miembros; y la sustitución de las personas o el cambio de los miembros del equipo no son aconsejables en un *Scrum Core Team*. Por lo tanto, es importante contar con personas en un *Scrum Core Team* que estén disponibles y plenamente comprometidas con el proyecto.

8.2.1.10 **Organizational Resource Matrix**

Organizational Resource Matrix es una representación jerárquica de una combinación de una estructura de organización funcional y una estructura organizativa proyectizada. Las organizaciones matriciales reúnen a miembros de varios equipos del proyecto de diferentes departamentos funcionales, tales como alguien de la tecnología, las finanzas, marketing, ventas, manufactura, y otros departamentos – para así crear equipos multifuncionales.

Los miembros del equipo en una organización matricial cumplen dos objetivos - funcionales y de proyecto. Los miembros del equipo son dirigidos por el/los *Product Owner(s)* con respecto a las actividades de proyectos, mientras que los directores funcionales realizan actividades de gestión relacionadas con sus departamentos, como las evaluaciones de desempeño y la aprobación de ausencia.

8.2.1.11 Skills Requirement Matrix

Skills Requirement Matrix también conocido como un marco de competencias, se utiliza para evaluar las carencias de habilidades y los requisitos de formación para los miembros del equipo. Una matriz de habilidades asigna las habilidades, las capacidades y el nivel de interés de los miembros del equipo en el uso de esas habilidades y capacidades en un proyecto. Al utilizar esta matriz, la organización puede evaluar los vacíos de habilidades en los miembros del equipo e identificar a los empleados que van a necesitar más formación en un área o competencia particular.

8.2.1.12 Scrum Guidance Body Recommendations

Descrito en la sección 8.1.1.12.

8.2.2 Herramientas

8

8.2.2.1 Criterios de selección*

La selección adecuada de *Scrum Master(s)* y la identificación del/de *stakeholder(s)* pertinente(s) es crucial para el éxito de cualquier proyecto. En algunos proyectos, puede ser que haya habido pre-condiciones que estipulen determinados miembros del equipo y sus roles.

Cuando hay flexibilidad en la elección del/de los *Scrum Master(s)*, se deben considerar los siguientes *Selection Criteria*:

1. *Habilidades para la resolución de problemas*—Es uno de los principales criterios a considerar al seleccionar al/a los *Scrum Master(s)*. El/los *Scrum Master(s)* debe(n) tener las habilidades y experiencia necesarias para ayudar a eliminar cualquier *impediment* que encare el *Scrum Team*.
2. *Disponibilidad*—El *Scrum Master* debe estar disponible para programar, supervisar y facilitar varias reuniones, incluyendo *Release Planning Meeting*, *Daily Standup Meeting*, y otras reuniones relacionadas al *Sprint*.
3. *Compromiso*—El *Scrum Master* se debe comprometer a que el *Scrum Team* esté dotado de un ambiente de trabajo propicio para asegurar la entrega exitosa de los proyectos Scrum.
4. *Servant Leadership Style*—Para más detalles, consulte la sección 3.10.4.1

En la identificación del/de los *stakeholder(s)*, es importante recordar que el/los *stakeholders* incluye a todos los *customers*, los usuarios y patrocinadores, quienes a menudo interactúan con el *Product Owner*, *Scrum Master* y *Scrum Team* para proveer entradas y facilitar la creación de los productos del proyecto. Los *stakeholders* influyen el proyecto a lo largo de su ciclo de vida.

8.2.2.2 *Expert Advice from HR*

Expert Advice from HR puede ser útil en la identificación del *Scrum Master* y del/ de los *Stakeholder(s)*. El departamento de Recursos Humanos posee un conocimiento específico sobre los empleados de una organización y las diversas técnicas que pueden ayudar a identificar a un *Scrum Master* como a un *Stakeholder*.

8.2.2.3 Formación y costos de capacitación

Scrum es un marco radicalmente diferente a los métodos tradicionales de gestión de proyectos. Los miembros del equipo no siempre pueden poseer los conocimientos o habilidades necesarias para trabajar en el entorno de Scrum. El *Product Owner* debería evaluar las necesidades de capacitación de los miembros potenciales del equipo y facilitar la formación para eliminar tal carencia. El *Product Owner* es normalmente responsable de la evaluación y la selección de los miembros del equipo, pero a menudo lo hace en consulta con el *Scrum Master* que pueden tener un conocimiento adicional de los recursos de los que trabajan con ellos en otros proyectos.

Una formación adecuada se le debe proporcionar a los miembros del *Scrum Team*, tanto antes del inicio de los trabajos y también mientras están trabajando en sus proyectos. Los miembros del *Scrum Team* también deben estar dispuestos a aprender de los demás, y de quienes tienen más experiencia en el equipo.

8.2.2.4 Costo de los recursos

Una de las principales consideraciones en la selección de las personas tiene que ver con las ventajas y desventajas relacionadas con el nivel de experiencia en comparación al salario. Hay otros factores relacionados a las personas que influyen en el costo que también se deben considerar. Idealmente, el/los *Scrum Master(s)*, los miembros del equipo, y el/los *stakeholder(s)* deben ser *colocated*, para que puedan comunicarse con frecuencia y facilidad. Si *colocation* no es posible y hay equipos distribuidos, se debe tener recursos adicionales para facilitar la comunicación, el entendimiento de las diferencias culturales, sincronizar el trabajo y fomentar el intercambio de conocimientos.

8.2.3 Salidas

8.2.3.1 *Identified Scrum Master**

Un *Scrum Master* es un facilitador y *servant leader* que se asegura de que el *Scrum Team* esté dotado de un ambiente propicio para completar el proyecto con éxito. El *Scrum Master* guía, facilita y les enseña prácticas de *Scrum* a todos los involucrados en el proyecto; borra *impediments* que encara el equipo; y asegura que se estén siguiendo los procesos de *Scrum*. Es la responsabilidad del *Product Owner* identificar al *Scrum Master* para un proyecto *Scrum*.

El papel de *Scrum Master* se describe con más detalle en la sección 3.4

8.2.3.2 *Identified Stakeholder(s)**

El/Los *Stakeholder(s)*, que es un término colectivo que incluye a los *customers*, los usuarios y los patrocinadores, con frecuencia interactúa(n) con el *Scrum Core Team* e influye(n) en el proyecto durante todo el proceso de desarrollo de productos. Es para los *stakeholders* que el proyecto produce los beneficios deseados de colaboración.

La función del/de los *stakeholder(s)* se describe en la sección 3.3.2.

8.3 Form Scrum Team

Figura 8-8 muestra todas las entradas, las herramientas y las salidas para el proceso de *Form Scrum Team*.

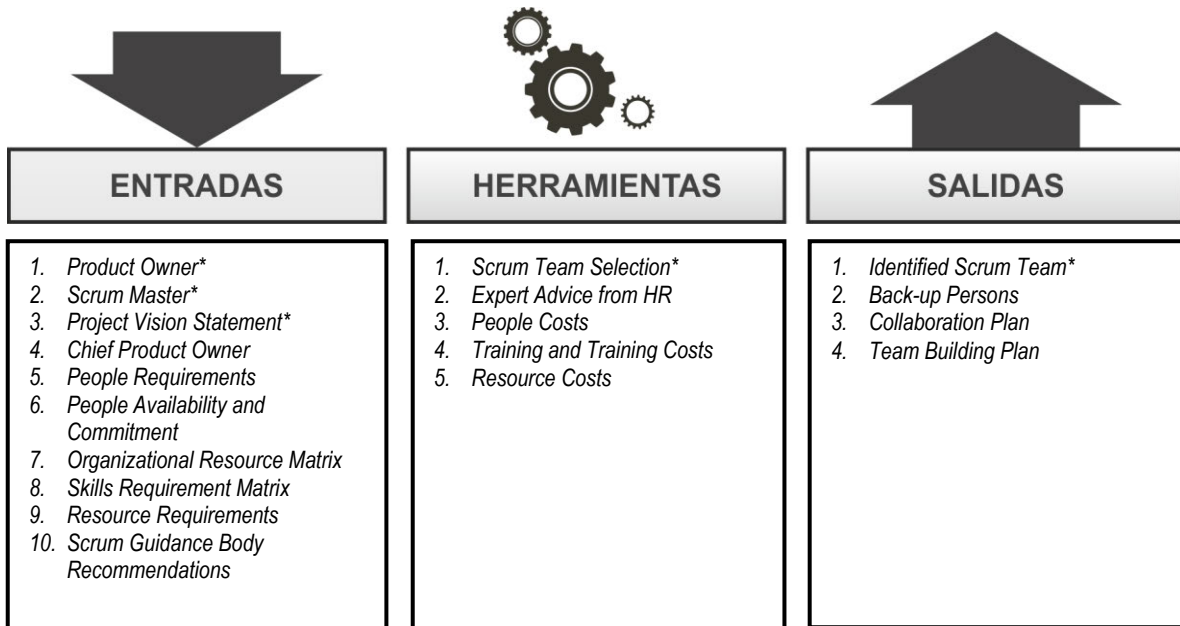
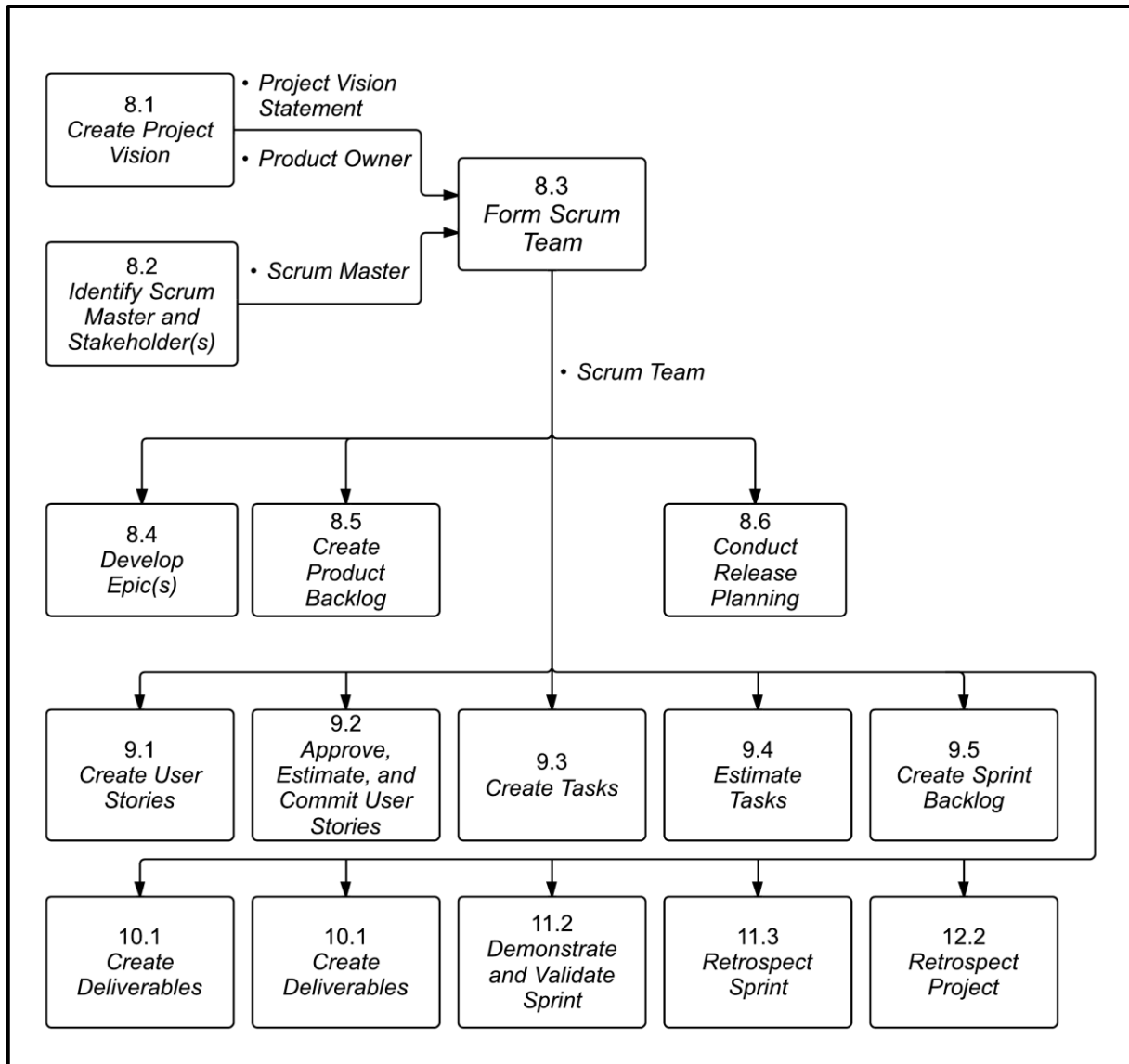


Figura 8-8: Form Scrum Team—Entradas, Herramientas, y Salidas

Nota: Los asteriscos () denotan una entrada, herramientas o salida "obligatoria" para el proceso correspondiente.*

Figura 8-9: *Form Scrum Team*—Diagrama de flujo de datos

8.3.1 Entradas

8.3.1.1 *Product Owner**

Descrito en la sección 8.1.3.1.

8.3.1.2 *Scrum Master**

Descrito en la sección 8.2.3.1.

8.3.1.3 *Project Vision Statement**

Descrito en la sección 8.1.3.2.

8.3.1.4 *Chief Product Owner*

Descrito en la sección 8.1.1.5.

8.3.1.5 *People Requirements*

Descrito en la sección 8.2.1.8.

8.3.1.6 *People Availability and Commitment*

Descrito en la sección 8.2.1.9.

8.3.1.7 *Organizational Resource Matrix*

Descrito en la sección 8.2.1.10.

8.3.1.8 *Skills Requirement Matrix*

Descrito en la sección 8.2.1.11.

8.3.1.9 Requisitos de recursos

Estos requisitos incluyen todos los recursos, sean personas o no, necesarios para que el *Scrum Team* funcione con eficacia. Estos recursos incluyen infraestructura de oficinas, espacios de reunión, los equipos de trabajo, *Scrumboards*, etc. En el caso de los equipos virtuales, recursos adicionales, tales como herramientas de *collaboration*, videoconferencia, repositorios de documentos compartidos, servicios de traducción, etc. tienen que ser considerado.

8.3.1.10 Recomendaciones del Scrum Guidance Body

Descrito en la sección 8.1.1.12.

8.3.2 Herramientas

8.3.2.1 Selección del Scrum Team*

El *Scrum Team* es la base de cualquier proyecto de Scrum y tener a los miembros adecuados para el equipo es vital para la entrega exitosa de los proyectos Scrum. Los miembros del *Scrum Team* son generalistas/ especialistas ya que cuentan con conocimiento de diversos campos y son expertos en al menos uno. Más allá de la experiencia en la materia, son las habilidades interpersonales de cada miembro del equipo que determinará el éxito de los equipos auto-organizados.

Los miembros ideales del *Scrum Team* son independientes, auto-motivados, se enfocan en el *customer*, y tienen un sentido alto de la responsabilidad y la colaboración. El equipo debe ser capaz de fomentar un ambiente de reflexión independiente y de tomar decisiones con el fin de extraer los mayores beneficios de la estructura.

8.3.2.2 Expert Advice from HR

Los gerentes de *Expert Advice from Human Resource (HR)* pueden ser valiosos para la formación de un *Scrum Team*. El departamento de Recursos Humanos posee un conocimiento especializado sobre los empleados de una organización y de las numerosas técnicas que pueden ayudar a los *Product Owners*, *Scrum Masters*, y a los patrocinadores para identificar a los mejores miembros del equipo.

8.3.2.3 Costo asociado con el personal

Todos los costos asociados con los requisitos de las personas deben ser evaluados, analizados, aprobados y presupuestados.

8.3.2.4 Costos de formación y capacitación

Es posible que los miembros del equipo no cuenten con las habilidades o conocimientos necesarios para llevar a cabo tareas especializadas. En ese caso, el *Product Owner* debe evaluar las necesidades de capacitación de los miembros potenciales del equipo y proporcionar capacitación, cuando se ha encontrado una carencia de habilidades o conocimientos.

Para que una implementación de Scrum sea realmente eficaz, tiene que haber un alto nivel de conciencia dentro de la organización de los principios y valores de Scrum. Este conocimiento ayudará a la ejecución exitosa de Scrum. El *Scrum Team* tiene que estar capacitado en las prácticas de Scrum, y es el *Scrum Master* quien debe desempeñar el papel de entrenador de este tema para el equipo. Dado a que la planificación de los *Sprints* es un factor muy importante para el éxito, el entrenamiento le ayudará a los miembros del equipo a entender cómo discutir e identificar metas alcanzables en un *Sprint*. El *Scrum Master* tiene que sacar lo mejor del *Scrum Team* motivándolos y facilitando el proceso de desarrollo. Con el entrenamiento correcto, el *Scrum Master* puede ayudar al equipo a articular *issues* y desafíos que puedan enfrentar. Normalmente, cualquier *issue* o conflictos en el equipo se resuelven por el mismo equipo con entrenamiento y la asistencia del *Scrum Master*, según sea necesario. El *Scrum Master* debe atender los *issues* como la baja moral o falta de coordinación dentro del equipo. Él/ella es responsable de la eliminación de *impediments* para el equipo. Cuando sea necesario, el *Scrum Master* puede presentarles estos *issues* externos e *impediments* de gestión a la dirección para llegar a una resolución.

Los costos de formación y capacitación son discutidos en el proceso de *Identify Scrum Master and Stakeholder(s)*, sección 8.2.2.3.

8.3.2.5 Costo de los recursos

Los costos asociados con todos los requisitos no relacionados con las personas deben ser evaluados, analizados, aprobados y presupuestados. Un recurso en el entorno del proyecto es identificado como aquello que se usa para realizar una tarea o actividad, incluyendo-pero no limitado a equipos, materiales, servicios externos, y el entorno físico.

8.3.3 Salidas

8.3.3.1 *Scrum Team* identificado*

El *Scrum Team*, a veces conocido como *Development Team*, es un grupo o equipo de personas que son responsables de la comprensión de los *business requirements* especificados por el *Product Owner*, la estimación de *user stories* y la creación definitiva de los *Entregables* del proyecto. Los *Scrum Teams* son multi-funcionales y auto-organizados. El equipo decide la cantidad de trabajo que se comprometerá en un *Sprint* y determina la mejor manera de realizar el trabajo. El *Scrum Team* se compone de miembros multi-

funcionales, que llevan a cabo todo el trabajo involucrado en la creación de Entregables que potencialmente se puedan enviar, incluyendo el desarrollo, la prueba, *quality assurance*, etc.

La identificación del *Scrum Team* es la responsabilidad del *Product Owner*, a menudo en consulta con el *Scrum Master*.

El papel del *Scrum Team* se describe con más detalle en la sección 3.6.

8.3.3.2 Personas de respaldo

Al seleccionar a los equipos, otro aspecto importante es la creación de substitutos para cada miembro del *Scrum Team*. Aunque la disponibilidad y compromiso por los miembros del equipo se confirman por adelantado, pueden surgir *issues* tales como una enfermedad, emergencia familiar, o simplemente el hecho de que un miembro se marche de la organización. Los *Scrum Team* trabajan en pequeños grupos de seis a diez personas. El tener reemplazantes asegura que no haya una disminución importante en la productividad debido a la falta de un miembro del equipo.

8

8.3.3.3 Plan de Collaboration

Collaboration es un elemento muy importante en Scrum. La planificación de cómo las distintas personas que toman decisiones, los *stakeholders*, y miembros del equipo participan y colaboran entre sí es vital. *Collaboration Plan* es una salida (*output*) opcional que puede ser formal o informal. A veces, puede ser simplemente un entendimiento verbal entre los diversos *stakeholders*, ya en que Scrum se evita toda la documentación necesaria. Sin embargo, para los proyectos más grandes y complejos, especialmente aquellos con equipos distribuidos, puede ser necesario poner en marcha un acuerdo más formal. El plan puede abordar cómo los miembros de Scrum Core Team, *stakeholder(s)* y otras personas involucradas en el proyecto Scrum se comunican y colaboran durante todo el proyecto y también puede definir las herramientas o técnicas específicas que se utilizarán para este fin. Por ejemplo, en el caso de equipos distribuidos, puede haber una necesidad de un acuerdo sobre cuándo y cómo se llevarán a cabo las reuniones, qué tipo de herramientas de comunicación se utilizará, y quién debe estar involucrado en las diversas reuniones.

8.3.3.4 Team Building Plan

Debido a que un *Scrum Team* es multi-funcional, cada miembro debe participar activamente en todos los aspectos del proyecto. El *Scrum Master* debe identificar los *issues* con los miembros del equipo y encararlos con el fin de mantener un equipo eficaz.

Para tener cohesión en el equipo, el *Scrum Master* debe asegurarse que las relaciones entre los miembros del equipo sean positivas y que estén unificados en la consecución de la totalidad del proyecto y de los objetivos de la organización, lo que conduce a una mayor eficiencia y una mayor productividad.

En este contexto, es importante estudiar la sección 3.10, que habla sobre las teorías populares de recursos humanos y su relevancia para Scrum.

8.4 Develop Epic(s)

La figura 8-10 muestra todas las entradas, herramientas y salidas para el proceso de *Develop Epic(s)*.

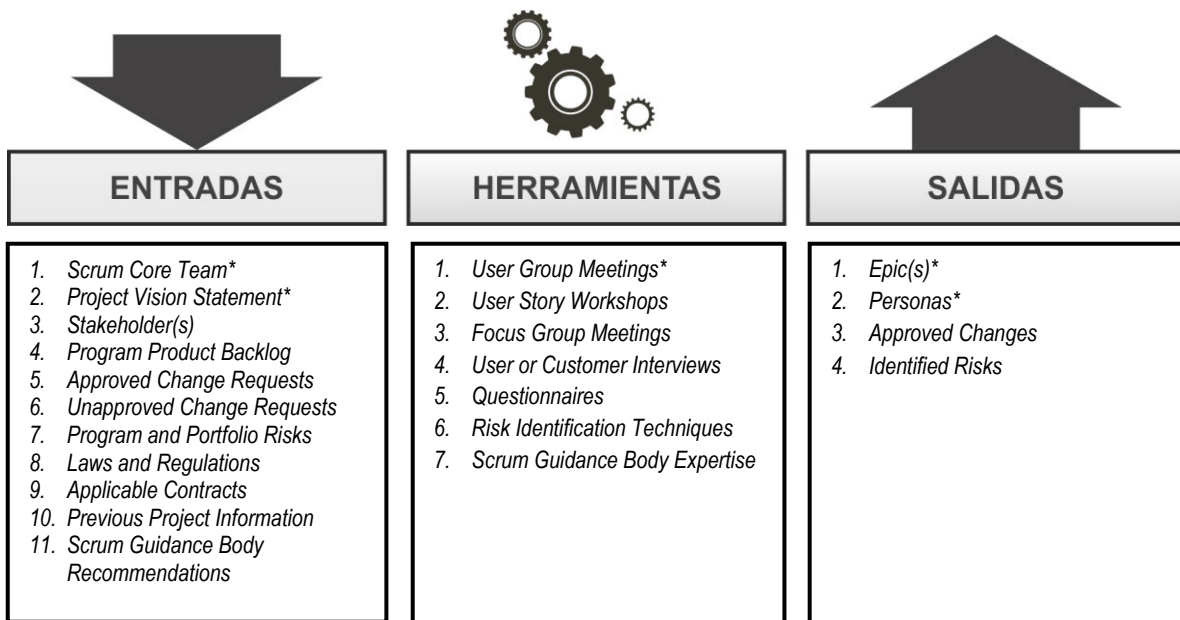
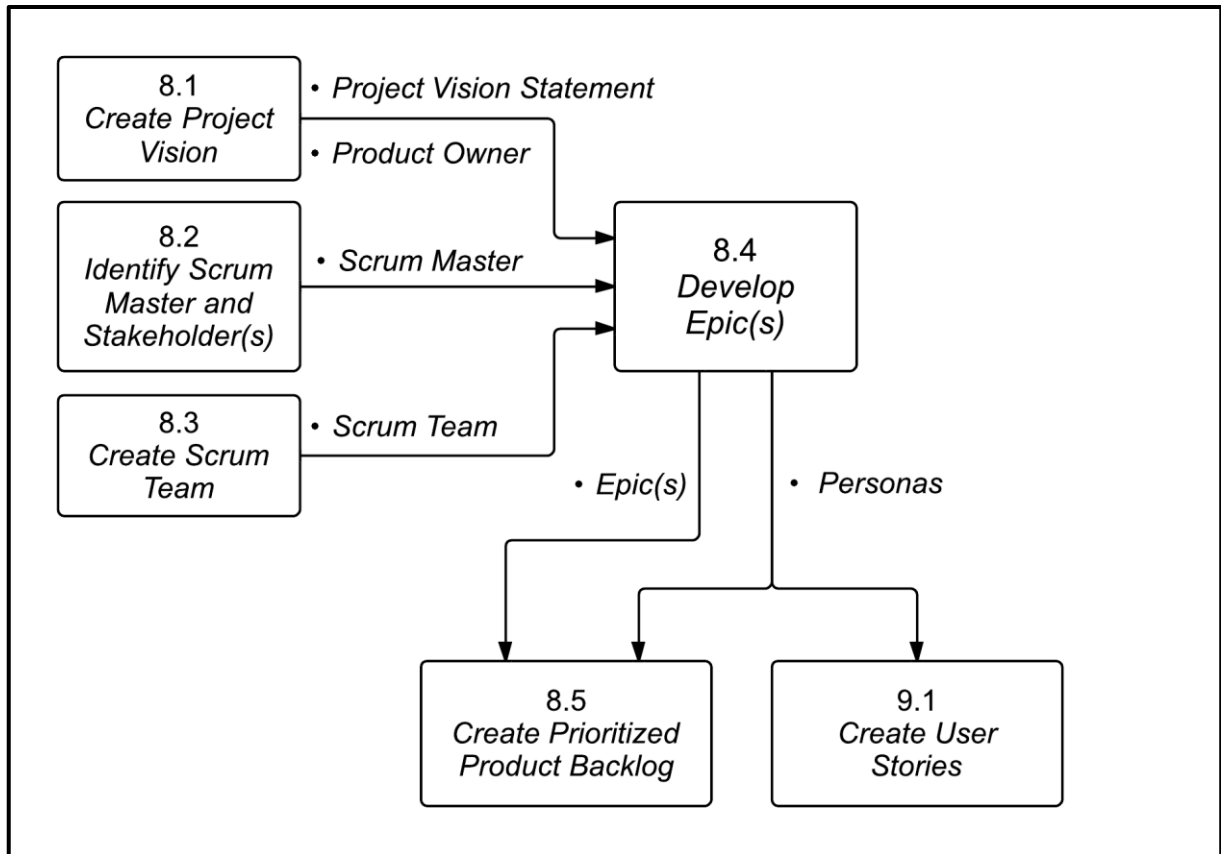


Figura 8-10: *Develop Epic(s)*—Entradas, Herramientas, y Salidas

Figura 8-11: *Develop Epic(s)*—Diagrama de Flujo

8.4.1 Entradas

8.4.1.1 *Scrum Core Team**

El equipo central de Scrum consiste en el *Scrum Team*, el *Scrum Master* y el *Product Owner* como se describe en la sección 3.3.1.

8.4.1.2 *Project Vision Statement**

Descrito en la sección 8.1.3.2.

8.4.1.3 *Stakeholder(s)*

Descrito en la sección 8.2.3.2.

8.4.1.4 *Program Product Backlog*

Descrito en la sección 8.1.1.6.

8.4.1.5 *Approved Change Requests*

Approved changed requests que vienen del programa o *portfolio* son entradas que se añadirán a la lista de cambios de proyecto aprobados para su ejecución en *Sprints* futuros. Cada cambio puede requerir su propio *Epic* o *User Story* y podría convertirse en un aporte al proceso de *Develop Epic(s)*. *Approved Change Requests* en este proceso también podría ser el resultado de otros procesos de Scrum.

Change Requests y *Approved Change Requests* se discuten en las secciones 6.3.1, 6.4.2.1 y 6.6.

8.4.1.6 *Unapproved Change Requests*

Los pedidos de cambios se presentan por lo general como *Change Requests* y permanecer en un estado de *no aprobado* hasta que sean formalmente aprobados. *Unapproved Change Requests* para el proceso *Develop Epic(s)* podría provenir de *Create Deliverables*, *Conduct Daily Standup* y otros procesos.

Change requests y *Unapproved Change Requests* se discuten en las secciones 6.3.1, 6.4.2.1 y 6.6.

8.4.1.7 **Program and Portfolio Risks**

Risks relacionados con un *portfolio* o el programa también tendrán un impacto en los proyectos que forman parte del respectivo *portfolio* o programa. Durante *risk assessment* de *portfolios* y programas, si se determina que el riesgo puede afectar a un proyecto individual, la información relevante sobre el riesgo debe ser comunicada al *Product Owner* y *Scrum Team*. *Program and Portfolio Risks* podrían ser entradas para el proceso *Develop Epic(s)* y pueden tener un impacto global sobre cómo se lleva a cabo este proceso.

Program and Portfolio Risks se describen en la sección 7.5.1.

8.4.1.8 **Laws and Regulations (Leyes y regulaciones)**

Dependiendo del proyecto, hay *Laws and Regulations*, impuestos por un consejo de administración, que afectan la planificación y ejecución. Las leyes son externas a la organización e impuesta por una entidad gubernamental. Las regulaciones pueden ser o bien interna o externa. Las regulaciones internas son aquellas que son aplicables dentro de la empresa, por lo general basadas en la política de la empresa. Estas regulaciones pueden estar relacionadas con los sistemas de *quality management*, regulaciones financieras, reglamentos de personal, etc. Las regulaciones externas son las relativas a los criterios, las normas y los requisitos gubernamentales establecidos.

Laws and Regulations deben ser considerados al desarrollar *Epics*. Los *Epics* se basan en *business requirements*. Para cumplir con estos requisitos el equipo del proyecto tiene que cumplir con ambas leyes y reglamentos internos y externos.

A veces, algunas de las leyes y reglamentos que afectan a múltiples proyectos Scrum pueden incluirse como parte de los *Scrum Guidance Body Recommendations*, como se discutió en la sección 8.1.1.12.

8.4.1.9 **Los contratos aplicables**

Si todo el proyecto o partes del mismo se cumplen en virtud de un contrato, el contrato ha de definir el alcance del trabajo y las condiciones específicas del mismo. El tipo de contrato utilizado tendrá influencia en el riesgo del proyecto.

Algunos de los tipos más comunes de contratos utilizados en proyectos Scrum son los siguientes:

Incremental Delivery Contract—Este contrato incluye puntos de *inspección* en intervalos regulares. Ayuda a que el *customer* o *stakeholders* tomen decisiones sobre el desarrollo de los productos periódicamente a lo largo del proyecto en cada punto de inspección. El *customer* puede aceptar el desarrollo del producto, decidir parar el desarrollo del producto, o solicitar modificaciones del producto.

Joint Venture Contract—Este contrato se utiliza generalmente cuando dos o más socios llevan a cabo el trabajo de un proyecto. Las partes involucradas en el proyecto logran algún *Return on Investment*, porque los ingresos o beneficios generados serán compartidos entre todos.

Development in Phases Contract—Este contrato permite que los fondos estén disponibles cada mes o cada trimestre después de que un *release* se complete con éxito. De esta forma, se les incentiva a el *customer* y a los proveedores y se asegura de que el riesgo monetario para el *customer* se limite a ese período de tiempo determinado, ya que no son financiados los lanzamientos sin éxito.

Incentive and Penalty Contract—Estos contratos se basan en el acuerdo de que el proveedor se verá recompensado con un incentivo financiero si los productos del proyecto se entregan a tiempo, y acumulará sanciones financieras si la entrega se retrasa.

Otros tipos de contratos más populares incluyen el pago por las características del contrato, el tiempo y los materiales del contrato, precio fijo y contrato de alcance fijo, y el contrato de beneficio fijo.

Los *Epics* deberían desarrollarse teniendo en cuenta los términos y condiciones del tipo de contrato que se utilice.

8.4.1.10 Previous Project Information

Información y conocimientos adquiridos en proyectos anteriores y similares dentro de la organización son insumos valiosos para el desarrollo de *Epics* y de la evaluación de riesgos. *Previous Project Information* podría incluir las notas del director del proyecto, registros de proyectos y los comentarios por el *stakeholder*.

Algunos detalles y las mejores prácticas relacionadas con información de proyectos anteriores pueden estar disponibles a través de *Scrum Guidance Body Recommendations*.

8.4.1.11 Scrum Guidance Body Recommendations

Discutidos en la sección 8.1.1.12

Scrum Guidance Body Recommendations pueden incluir información sobre las normas, los reglamentos, las reglas y las mejores prácticas para el desarrollo de *Epics*.

8.4.2 Herramientas

8.4.2.1 *User Group Meetings**

User Group Meetings implica a relevantes *stakeholders* (principalmente usuarios o *customers* del producto). Ellos le proporcionan al Scrum Core Team con información de primera mano acerca de las expectativas del usuario. Esto ayuda en la formulación de los *Acceptance Criteria* para el producto y proporciona información valiosa para el desarrollo de *Epics*. *User Group Meetings* son de vital importancia en la prevención del trabajo costoso que puedan deberse a la falta de claridad con respecto a las expectativas y exigencias. Estas reuniones también promueven que se crea en el proyecto y crea un entendimiento común entre el Scrum Core Team y el/los *Stakeholder(s)* *pertinente(s)*.

8.4.2.2 *User Story Workshops*

User Story Workshops se llevan a cabo como parte del proceso de *Develop Epic(s)*. El *Scrum Master* facilita estas sesiones, en las que todo el *Scrum Core Team* interviene, y en ocasiones, es deseable incluir a otro(s) *Stakeholder(s)*. Estos talleres ayudan al *Product Owner* a dar prioridad a los requisitos y permitir que el *Scrum Core Team* tenga una perspectiva compartida de los *Acceptance Criteria*. Se aseguran de que los *Epics* y *User Stories* describan la funcionalidad desde el punto de vista de los usuarios, sean fáciles de entender, y que puedan ser calculados de forma fiable. *User Story workshops* son útiles en la comprensión de las expectativas del usuario para los entregables y son excelentes para la formación de equipos. También facilitan la preparación para la planificación del próximo *Sprint*. Un Taller de historia de usuario es una buena plataforma para discutir y aclarar todos los elementos de un producto y a menudo profundizar en los detalles más pequeños para garantizar la claridad.

8.4.2.3 *Focus Group Meetings*

En los *Focus Group Meetings* se reúnen las personas en una sesión guiada para proporcionar sus opiniones, percepciones o valoraciones de un producto, servicio o resultado deseado. Los miembros del grupo de enfoque tienen la libertad de hacerse preguntas el uno al otro y de obtener aclaraciones sobre temas o conceptos específicos. A través de cuestionamiento, la crítica constructiva y la retroalimentación, los grupos focales conducen a un producto de mejor calidad y con ello contribuyen a la satisfacción de las expectativas de los usuarios. En estas reuniones, los miembros del grupo de enfoque a veces llegan a un consenso en ciertas áreas, mientras que en otras áreas las opiniones pueden ser diferentes. Cuando los miembros del grupo tienen diferentes opiniones o puntos de vista, no se escatiman esfuerzos para resolver las diferencias con el fin de llegar a un consenso.

Las sesiones de grupos focales pueden ayudar a los equipos a crear ideas innovadoras, resolver problemas, y dar sugerencias para mejorar. Estas reuniones generan ideas y retroalimentación de los

usuarios potenciales y desarrolladores de productos. Estas reuniones se realizan normalmente para la planificación, la evaluación, o la mejora de un producto o servicio. Perspectivas obtenidas a partir de estas reuniones también pueden ayudar con *Develop Epics*, y con los *User Stories*. A veces, los *focus group meetings* se llevan a cabo para resolver los *issues* que puedan surgir durante el desarrollo de los *Epics*.

8.4.2.4 Entrevistas con usuario o *customers*

El incorporar a los *stakeholders*, incluyendo al patrocinador, los usuarios, y los *customers* del producto, es importante para ganar el contexto y la visión necesaria para desarrollar *Epics*. Este tiempo compartido para entrevistar a los usuarios y *customers* resultará en un mejor alineamiento de los requisitos de *Epics* con la visión general de proyectos, ofreciendo de esta forma un mayor valor.

Estas entrevistas ayudan a:

- Identificar y entender las necesidades y expectativas del *stakeholder*
- Reunir opiniones y hechos
- Comprender la perspectiva del *stakeholder* sobre el producto final
- Recopilar la retroalimentación sobre el producto iterado o parcialmente desarrollado

8.4.2.5 Questionario

Una forma económica de obtener una perspectiva estadística cuantitativa y cualitativa de un gran número de usuarios o *customers* es el uso de encuestas o cuestionarios. Un cuestionario es un instrumento de investigación con el fin de recopilar información sobre un asunto o tema específico. Los cuestionarios pueden ser auto-administrados o administrados por un entrevistador.

Gran cuidado debe ser ejercido en el diseño de cuestionarios, la selección del público debe ser adecuada, y la determinación de un método apropiado de implementación de encuestas es también de gran importancia para evitar errores y prejuicios.

Durante el desarrollo de *Epics*, el *Product Owner* o el *Scrum Master* podrían llevar a cabo una encuesta para recopilar la información pertinente de *stakeholders* o el *Scrum Team*.

8.4.2.6 Técnicas de *Risk Identification*

Descrito en la sección 7.4.1.1

8.4.2.7 *Scrum Guidance Body Expertise*

Descrito en la sección 3.3.2

Al crear *Epics*, el *Scrum Guidance Body Expertise* podría relatar las normas y reglamentos documentados; o los estándares y mejores prácticas para la creación de *Epics*. También puede haber un equipo de expertos en la materia que pueda ayudar al *Product Owner* a crear *Epics*. Este equipo podría incluir analistas de negocios, arquitectos, desarrolladores, expertos de Scrum u otras personas con experiencia. Generalmente este grupo de expertos no es el mismo equipo que se quedará y trabajará en un proyecto en particular, ya que tienen que pasar de un proyecto a otro durante la "fase de venta" o "fase cero" con *customs* o usuarios.

8.4.3 Salidas

8.4.3.1 *Epic(s)**

8

Epics están escritos en las etapas iniciales del proyecto, cuando la mayoría de los *User Stories* son funcionalidades de alto nivel o descripciones de productos que están ampliamente definidas. *Epics* son *user stories* grandes sin refinar en el *Prioritized Product Backlog*.

Una vez que estos *Epics* vienen en el *Prioritized Product Backlog* para ser terminados en el próximo *Sprint*, se convierten en *User Stories* más pequeños. Estos *User Stories* más pequeñas son generalmente funcionalidades simples, cortas y fáciles de implementar, o bloques de tareas que deben completarse en un *Sprint*.

8.4.3.2 *Personas**

Personas son personajes de ficción altamente detallados. Estos representan a la mayoría de los usuarios y otros *stakeholders* que puede ser no utilicen directamente el producto final. *Personas* se creó para identificar las necesidades de los usuarios. La creación de *Personas* específicas puede ayudar al equipo a entender mejor a los usuarios y sus necesidades y metas. Basado en una *Persona*, el *Product Owner* puede priorizar de manera más efectiva las funciones para crear el *Prioritized Product Backlog*.

Creación de una *Persona*: Esto implica la asignación de un nombre ficticio y, preferentemente, una imagen de valores, al personaje. La *Persona* incluirá atributos muy específicos como la edad, el género, la educación, el medio ambiente, los intereses y objetivos. Una cita que ilustra las necesidades de la persona puede ser incluida también. A continuación se muestra un ejemplo de una *Persona* para un sitio web de viajes.

Ejemplo:

Vanessa tiene 39 años de edad y es residente de San Francisco. Ella es apasionada por viajar y después de haber tenido una exitosa carrera como abogado, ha decidido disfrutar de su pasión por el viaje. Le gusta tener opciones al seleccionar sus viajes aéreos y servicios de alojamiento para que ella pueda elegir el mejor a un precio asequible. Ella se siente frustrada con los sitios web lentos y desordenados.

8.4.3.3 *Approved Changes*

Unapproved Change Requests pueden ser aprobados por el *Product Owner* durante el proceso de *Develop Epic(s)*, a veces con sugerencias proporcionadas por los *stakeholders* relevantes. Tales cambios se clasifican como las modificaciones aprobadas (*approved changes*) y pueden ser priorizadas e implementadas en los futuros *Sprints*.

Change Requests y *Approved Change Requests* se discuten en las secciones 6.3.1, 6.4.2.1 y 6.6.

8.4.3.4 *Risks identificados*

Al crear *Epics*, nuevos *risks* pueden ser identificados y estos *risks* identificados constituyen una salida importante de esta etapa. Estos *risks* contribuyen al desarrollo del *Prioritized Product Backlog* (también se conoce como *Risk Adjusted Product Backlog*).

Risk identification se describe en la sección 7.4.1.

8.5 Create Prioritized Product Backlog

Figura 8-12 muestra todas las entradas, las herramientas y las salidas para el proceso *Create Prioritized Product Backlog*.

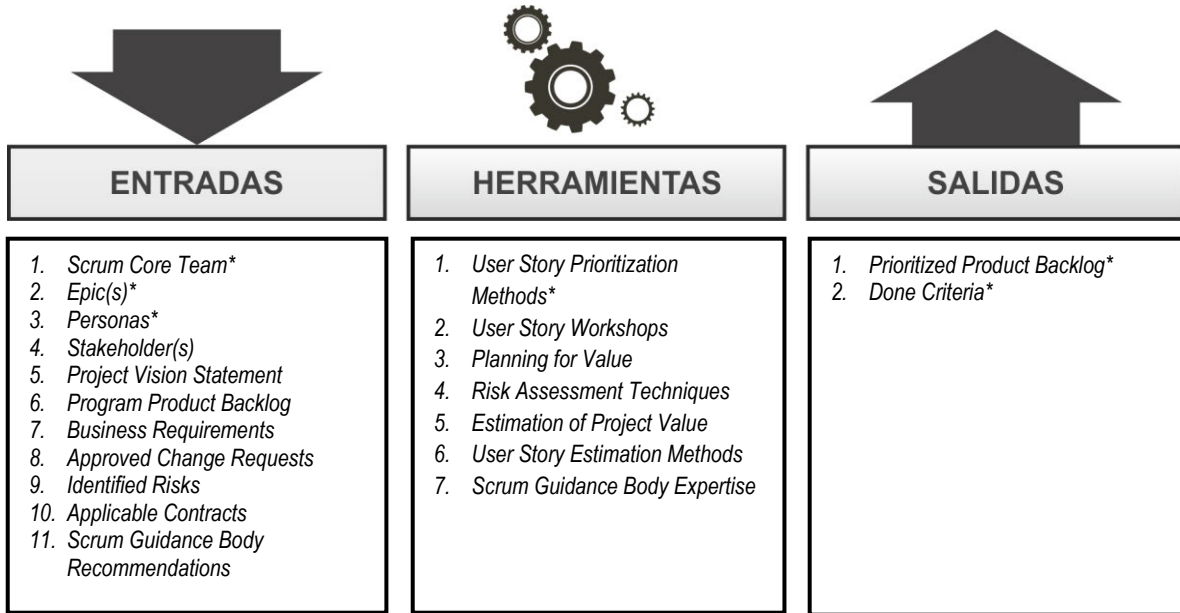
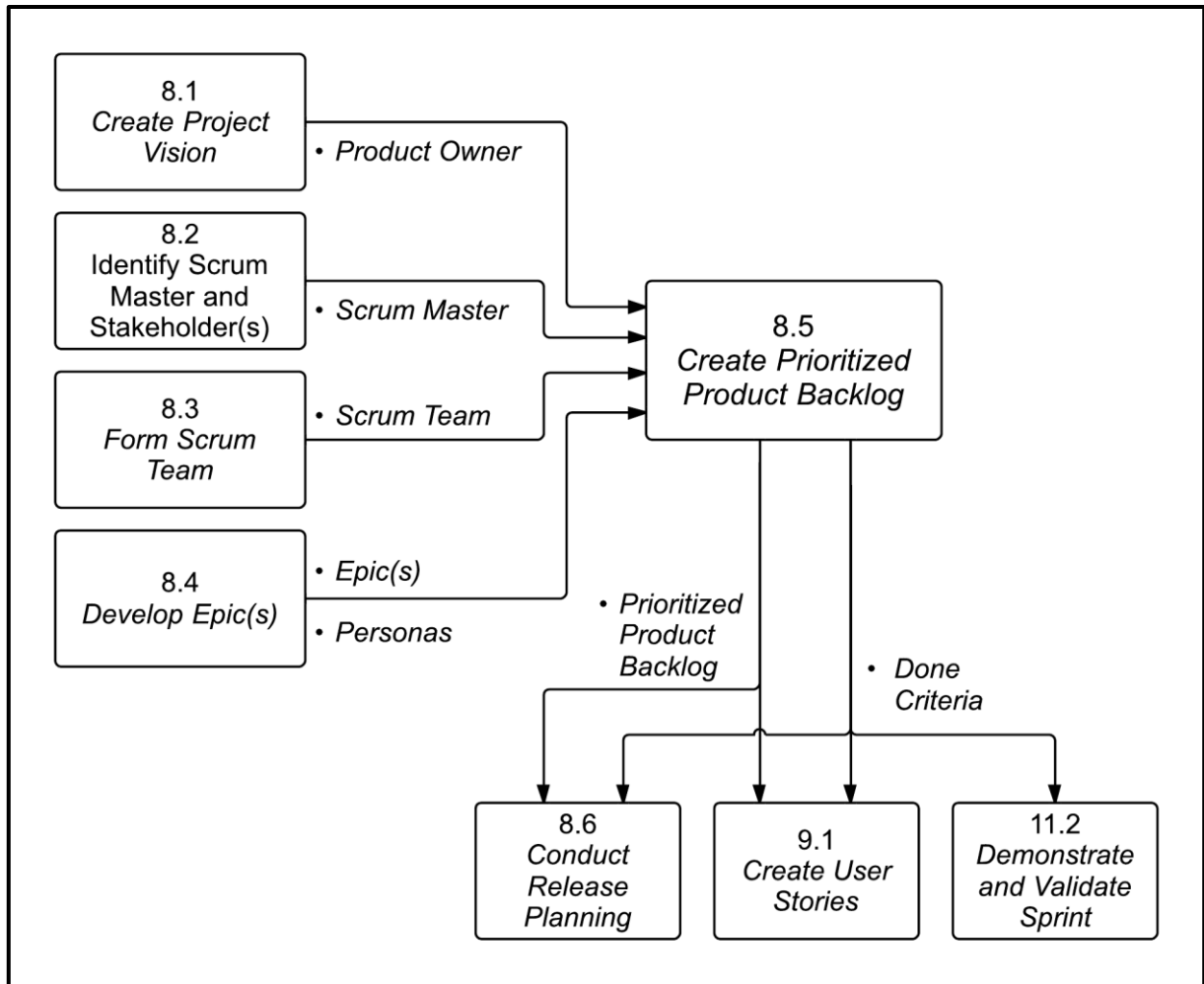


Figura 8-12: *Create Prioritized Product Backlog*—Entradas, Herramientas, y Salidas

Figura 8-13: *Create Prioritized Product Backlog*— Diagrama de flujo de datos

8.5.1 Entradas

8.5.1.1 *Scrum Core Team**

Descrito en la sección 8.4.1.1.

8.5.1.2 *Epic(s)**

Descrito en la sección 8.4.3.1.

8.5.1.3 *Personas**

Descrito en la sección 8.4.3.2.

8.5.1.4 *Stakeholder(s)*

Descrito en la sección 8.2.3.2.

8.5.1.5 *Project Vision Statement*

Descrito en la sección 8.1.3.2.

8.5.1.6 *Program Product Backlog*

Descrito en la sección 8.1.1.6.

8.5.1.7 *Business Requirements*

La suma de todos los conocimientos adquiridos a través de diversas herramientas como entrevistas a los *customers* o usuario o, Cuestionarios, JAD Sessions, Gap Analysis, SWOT Analysis, y otras reuniones, ayuda a desarrollar una mejor perspectiva sobre los *business requirements* y ayuda en la creación del *Prioritized Product Backlog*.

8.5.1.8 **Approved Change Requests**

Descrito en la sección 8.4.3.3.

8.5.1.9 **Risks identificados**

Descrito en la sección 8.4.3.4.

8.5.1.10 **Los contratos aplicables**

Descrito en la sección 8.4.1.9.

8.5.1.11 **Scrum Guidance Body Recommendations**

Descrito en la sección 8.1.1.12.

Durante la creación del *Prioritized Product Backlog*, *Scrum Guidance Body* puede incluir información sobre las normas, los reglamentos, los procedimientos y las mejores prácticas para el desarrollo del *Prioritized Product Backlog*.

8.5.2 **Herramientas**

8.5.2.1 **User Story Prioritization Methods***

Algunas de las técnicas que se utilizan para dar prioridad a los *User Stories* o requisitos en el *Prioritized Product Backlog*, sobre la base del valor de negocio se presentan a continuación:

- **MoSCoW Prioritization scheme**—El MoSCoW prioritization scheme deriva su nombre de las primeras letras de las frases "must have" (debe tener), "should have" (debería tener), "could have" (podría tener), y "will not have" (no tendrá)". Este método de *priorization* es generalmente más efectivo que un *simple scheme*. Las etiquetas están en orden decreciente de prioridad. "Debe tener" *user stories* son aquellos sin los que el producto no tendrá valor y "no tendrá" *User Stories* son aquellos que, a pesar de que sería bueno tener, no son necesarios para ser incluidos.
- **Paired Comparison**—esta técnica se prepara una lista de todas las *user stories* en el *Prioritized Product Backlog*. A continuación, cada historia de usuario se toma de forma individual y se compara con las otros *User stories* en la lista, uno a la vez. Cada vez que dos *User stories* se

comparan, se toma una decisión en cuanto a cuál de los dos es más importante. A través de este proceso, una lista de prioridades de las *User stories* se puede generar.

- **100-Point Method**—El *100-Point Method (Método de 100 puntos)* fue desarrollado por Dean Leffingwell y Don Widrig (2003). Se trata de darle al *customer* 100 puntos que pueden usar para votar por los *User Stories* que son más importantes. El objetivo es dar más peso a las *User Stories* que son de mayor prioridad en comparación con las otras *User Stories* disponibles. Cada miembro del grupo asigna puntos a las diversas *user stories*, dando más puntos a los que se sienten son más importantes. Al finalizar el proceso de votación, la prioritization se determina calculando el total de puntos asignados a cada *User Story*.
- **Kano Analysis**

Descrito en la sección 4.5.2

8.5.2.2 *User Story Workshops*

Descrito en la sección 8.4.2.2.

8.5.2.3 *Planning for Value*

Descrito en la sección 4.5.2

8.5.2.4 *Técnicas de Risk Assessment*

Descrito en la sección 7.4.2.1.

8.5.2.5 *Estimación Valor del Proyecto*

Descrito en la sección 4.5.1.

8.5.2.6 Métodos de estimación del *User Story*

Todas las herramientas estimadas usadas para los procesos *Approve*, *Estimate* and *Commit User Stories* (como se describe en la sección 9.2.2) se puede utilizar para crear estimaciones de alto nivel para Epic(s) cuando creamos el *Prioritized Product Backlog*. Algunas herramientas importantes son:

1. *User Group Meetings*
2. *Planning Poker*
3. *Fist of Five*
4. Puntos para la Estimación de Costos
5. Otras técnicas de estimación

8.5.2.7 *Scrum Guidance Body Expertise*

Descrito en la sección 8.4.2.7

Durante la creación del *Prioritized Product Backlog*, el *Scrum Guidance Body Expertise* podría relacionarse con las normas y reglamentos, y las mejores prácticas documentadas para la creación de *Epics*. También puede haber un equipo de expertos en la materia que podría ayudar al *Product Owner* en la opción *Create Prioritized Product Backlog*. Este equipo podría incluir analistas de negocios, arquitectos, desarrolladores, Scrum Expertos, y otras personas con experiencia. Generalmente este grupo de expertos no es el mismo equipo que se quedará trabajando en este proyecto, ya que tienden a pasar de un proyecto a otro durante la "fase de venta" o "fase cero" con los *customers* o usuarios.

8.5.3 Salidas

8.5.3.1 *Prioritized Product Backlog**

El *Product Owner* desarrolla una *Prioritized Product Backlog*, que contiene una lista priorizada de los requerimientos del negocio y de los proyectos escritos en forma de *Epic(s)*, que son de altos niveles de *User Stories*. El *Prioritized Product Backlog* se basa en tres factores principales: el valor, riesgo o incertidumbre, y dependencias. También se le conoce como *Risk Adjustment Product Backlog* dado a que incluye *risks* identificados y evaluados relacionados con el proyecto. También incluye cambios aprobados que pueden ser priorizados adecuadamente en el *Prioritized Product Backlog* (tal como se describe en la sección 6.3.1).

- **Valor**—Es la responsabilidad del *Product Owner* asegurar en primer lugar la entrega de los productos que ofrezcan el mayor valor. Incluso un producto de gran valor no puede ser parte del primer *release* si hay otros productos, incluso de mayor valor que son suficientes para un primer lanzamiento.

- **Riesgo e Incertidumbre**—Cuanta más incertidumbre existe, más riesgoso es el proyecto. Por lo tanto, es importante que los productos de mayor riesgo en el *Prioritized Product Backlog* se les de mayor prioridad. Los productos que llevan un mayor nivel de riesgo también requerirán acciones de *riks mitigation*. Cuando estas acciones de *risk mitigation* se priorizan contra el atraso, el resultado es un Risk Adjusted Product Backlog. Tratar con *risks* al principio del proyecto no garantiza que el proyecto será un éxito, pero sí mejorará la capacidad del equipo para hacer frente a los riesgos. Esto se describe en la sección 7.4.3.
- **Dependencias**—Por lo general, no es posible crear un *Prioritized Product Backlog* en la que no existen dependencias entre los User Stories. Los requisitos funcionales a menudo dependen de otros requisitos funcionales e incluso no funcionales. Estas dependencias pueden afectar cómo se priorizan los *user stories* en el *Prioritized Product Backlog*. Dos de las formas más comunes para resolver dependencias son o bien dividir una sola historia en varias partes, o combinar historias interdependientes.
- **Estimaciones**—Las estimaciones de alto nivel para *Epic(s)*, también están disponibles en el *Prioritized Product Backlog*.

8.5.3.2 Done Criteria*

Done Criteria es un conjunto de reglas que se aplican a todos los *User Stories*. Una definición clara de Done es crítica, ya que elimina la ambigüedad de los requisitos y ayuda a que el equipo se adhiera a las normas de calidad obligatorias. Esta clara definición se utiliza para crear los *Done Criteria*, que son un resultado del proceso de *Create Prioritized Product Backlog*. Una historia de usuario se considera *Done* cuando se le demuestra y es aprueba por el *Product Owner* que juzga sobre la base de los *Done Criteria* y los *User Story Acceptance Criteria*.

Ejemplo de Done Criteria:

Proyecto: Diseñar las nuevas variantes de un coche deportivo popular en LRA Ltd.

Done Criteria:

- El diseño es aprobado por la división de Excelencia Técnica.
- El prototipo pasa todas las pruebas de túnel de viento mandado por la División de Aerodinámica.
- El diseño se despeja para la producción de la división de propiedad intelectual.
- Expectativas el diseño de la seguridad son corroboradas por el informe de seguridad Diseño de la División de Seguridad.
- El informe de estimación de costos para el diseño es aprobado por la división de finanzas.

8.6 Conduct Release Planning

Figura 8-14 muestra todos las entradas, herramientas y productos para el proceso de *Conduct Release Planning*.

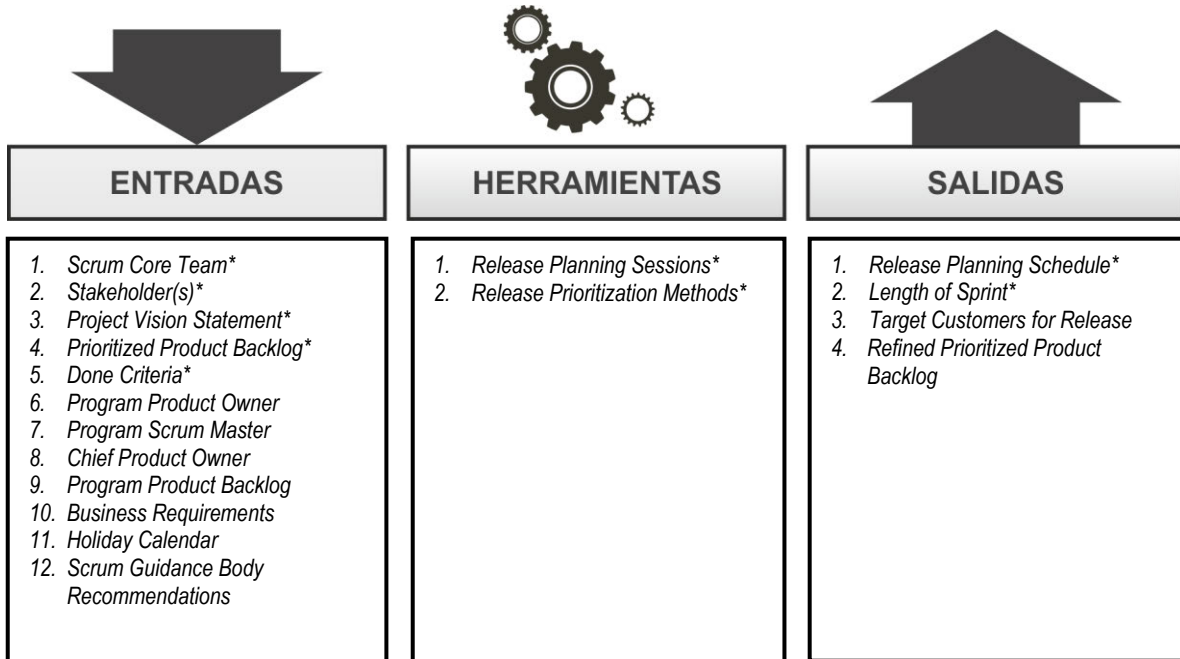


Figura 8-14: *Conduct Release Planning*—Entrada, Herramientas, y Salidas

Nota: Los asteriscos (*) denotan una entrada, herramientas o salida "obligatoria" para el proceso correspondiente.

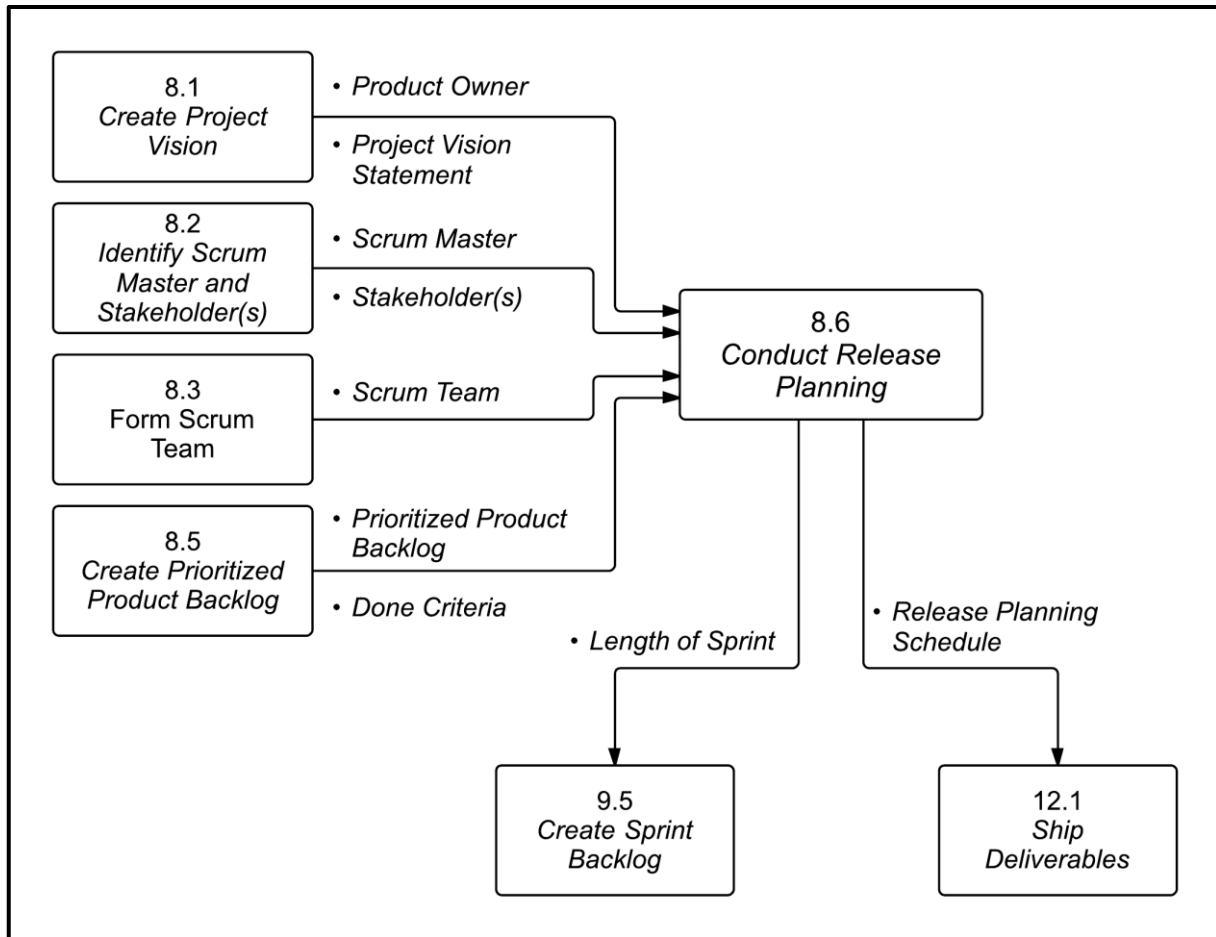


Figura 8-15: Conduct Release Planning—Diagrama de flujo de datos

8.6.1 Entradas

8.6.1.1 *Scrum Core Team**

Descrito en la sección 8.4.1.1.

8.6.1.2 *Stakeholder(s)**

Descrito en la sección 8.2.3.2.

8.6.1.3 *Project Vision Statement**

Descrito en la sección 8.1.3.2.

8.6.1.4 *Prioritized Product Backlog**

Descrito en la sección 8.5.3.1.

8.6.1.5 *Done Criteria**

Descrito en la sección 8.5.3.2.

8.6.1.6 *Program Product Owner*

Descrito en la sección 8.1.1.2.

8.6.1.7 *Program Scrum Master*

Descrito en la sección 8.1.1.3.

8.6.1.8 *Chief Product Owner*

Descrito en la sección 8.1.1.5.

8.6.1.9 **Program Product Backlog**

Descrito en la sección 8.1.1.6.

8.6.1.10 **Business Requirements**

Descrito en la sección 8.5.1.7.

8.6.1.11 **Calendario de vacaciones**

Es importante para el *Scrum Team* estar al tanto de las fechas claves y la disponibilidad de todos los miembros del equipo. Esto se puede lograr mediante el uso de un calendario compartido que proporciona información sobre los días feriados, licencias, faltas al trabajo, los planes de viaje, eventos, etc. Este calendario ayudará al equipo en la planificación y ejecución de *Sprints*.

8

8.6.1.12 **Scrum Guidance Body Recommendations**

Descrito en la sección 8.1.1.12

En el proceso de *Conduct Release Planning*, las recomendaciones de *Scrum Guidance Body* pueden relacionarse con las normas, reglamentos, estándares y mejores prácticas para el desarrollo del Plan de Lanzamiento. El Cuerpo de Orientación puede ser la mejor autoridad para definir las directrices relacionadas con el valor del negocio, las expectativas de liberación, estrategias de implementación, la calidad y la seguridad.

8.6.2 **Herramientas**

8.6.2.1 **Release Planning Sessions***

Release Planning Sessions se llevan a cabo para desarrollar un Plan de Lanzamiento. El plan define cuando varios conjuntos de funcionalidad o productos utilizables serán entregados al *customer*. En Scrum, el principal objetivo de una reunión de planificación de lanzamiento es hacer que el *Scrum Team* tenga una visión general de las emisiones y calendario de entrega para el producto que están desarrollando para que puedan alinearse con las expectativas del *Product Owner* y los relevantes *stakeholders* (principalmente el patrocinador del proyecto).

Muchas organizaciones tienen una estrategia en relación con la liberación de los productos. Algunas organizaciones prefieren despliegue continuo, donde se produce una liberación/un lanzamiento después de la creación de la funcionalidad útil especificada. Otras organizaciones prefieren despliegue por etapas, donde las liberaciones se hacen en intervalos predefinidos. Dependiendo de la estrategia de la organización, *Release Planning Session* en los proyectos puede ser impulsada por la funcionalidad, en el que el objetivo es hacer un lanzamiento una vez que un conjunto predeterminado de funcionalidad se ha desarrollado; o la planificación puede ser impulsada por la *fecha*, en la que ocurre la liberación en una fecha predefinida.

Dado a que el marco de Scrum promueve la planificación basada en información y la práctica de decisiones iterativas, sobre la planificación detallada por adelantado como, en la gestión de proyectos tradicional estilo cascada, no es necesario que *Release Planning Session* elabore un plan de lanzamiento detallado de todo el proyecto. En Scrum, el Plan de Lanzamiento (Release Plan) se puede actualizar continuamente a medida que la información relevante está disponible.

8.6.2.2 *Release Prioritization Methods**

Release Prioritization Methods se utilizan para desarrollar un *Release Plan*. Estos métodos son la industria y la organización específica y generalmente son determinados por la alta dirección de la organización.

8.6.3 Salidas

8.6.3.1 *Release Planning Schedule**

Un *Release Planning Schedule* es uno de los resultados más importantes del proceso llamado *Conduct Planning Schedule*. Un *Release Planning Schedule* indica que las entregas van a ser despachadas a los *customers*, junto con intervalos planificados, y fechas para los *releases*. Puede que no haya un lanzamiento previsto a finales de cada iteración *Sprint*. A veces, un lanzamiento puede ser planificado después que un grupo de iteraciones *Sprint* se ha completado. Dependiendo de la estrategia de la organización, *Release Planning Sessions* en los proyectos puede ser impulsado por la funcionalidad, en el que el objetivo es entregar una vez que un conjunto predeterminado de funcionalidad que se ha desarrollado, o la planificación puede ser impulsada por la *fecha*, en la que ocurre la liberación en una fecha predefinida. La entrega debe ser *released (liberada)* cuando ofrece valor empresarial suficiente para el *customer*.

8.6.3.2 *Length of Sprint**

Basado en las diversas entradas que incluyen *Business Requirements* y *Release Planning Schedule*, el *Product Owner* y el *Scrum Team* deciden sobre el *Length of Sprint* para el proyecto. Una vez determinado, el *Length of Sprint* a menudo sigue siendo el mismo durante todo el proyecto.

Sin embargo, el *Length of Sprint* puede ser cambiado si y como el *Product Owner* y el *Scrum Team* lo consideren apropiado. Puede ser que temprano en el proyecto todavía estén experimentando para encontrar la mejor longitud del *Sprint*. Si más adelante en el proyecto hay un cambio en *Length of Sprint* normalmente significa que puede ser reducido debido a las mejoras en el entorno del proyecto.

Un *Sprint* podría ser *Time-boxed* de 1 a 6 semanas. Sin embargo, para obtener los máximos beneficios de un proyecto Scrum, siempre se recomienda mantener el *Sprint Time-boxed* a 4 semanas, a menos que existan proyectos con requisitos muy estables, donde los *Sprints* pueden extenderse hasta 6 semanas.

Impacto del cambio esperado en el *Length of Sprint* se describe en la sección 6.5.1

8.6.3.3 *Target Customers for Release*

No todos los lanzamientos se dirigirán a todos los *stakeholders* o usuarios. El/los stakeholder(s) puede optar por limitar ciertos comunicados a un subconjunto de usuarios. El Release Plan debe especificar los *customers* en quienes se va a enfocar el lanzamiento (reléase).

8.6.3.4 *Refined Prioritized Product Backlog*

El *Prioritized Product Backlog*, desarrollado en el proceso de *Create Prioritized Product Backlog*, se puede refinar en este proceso. Es posible que haya más claridad sobre los *User Stories* en el *Prioritized Product Backlog* después de que el Equipo Principal de lleve a cabo *Release Planning Sessions* con el/los *Stakeholder(s)*.

8.7 Fase Diagrama de flujo de datos

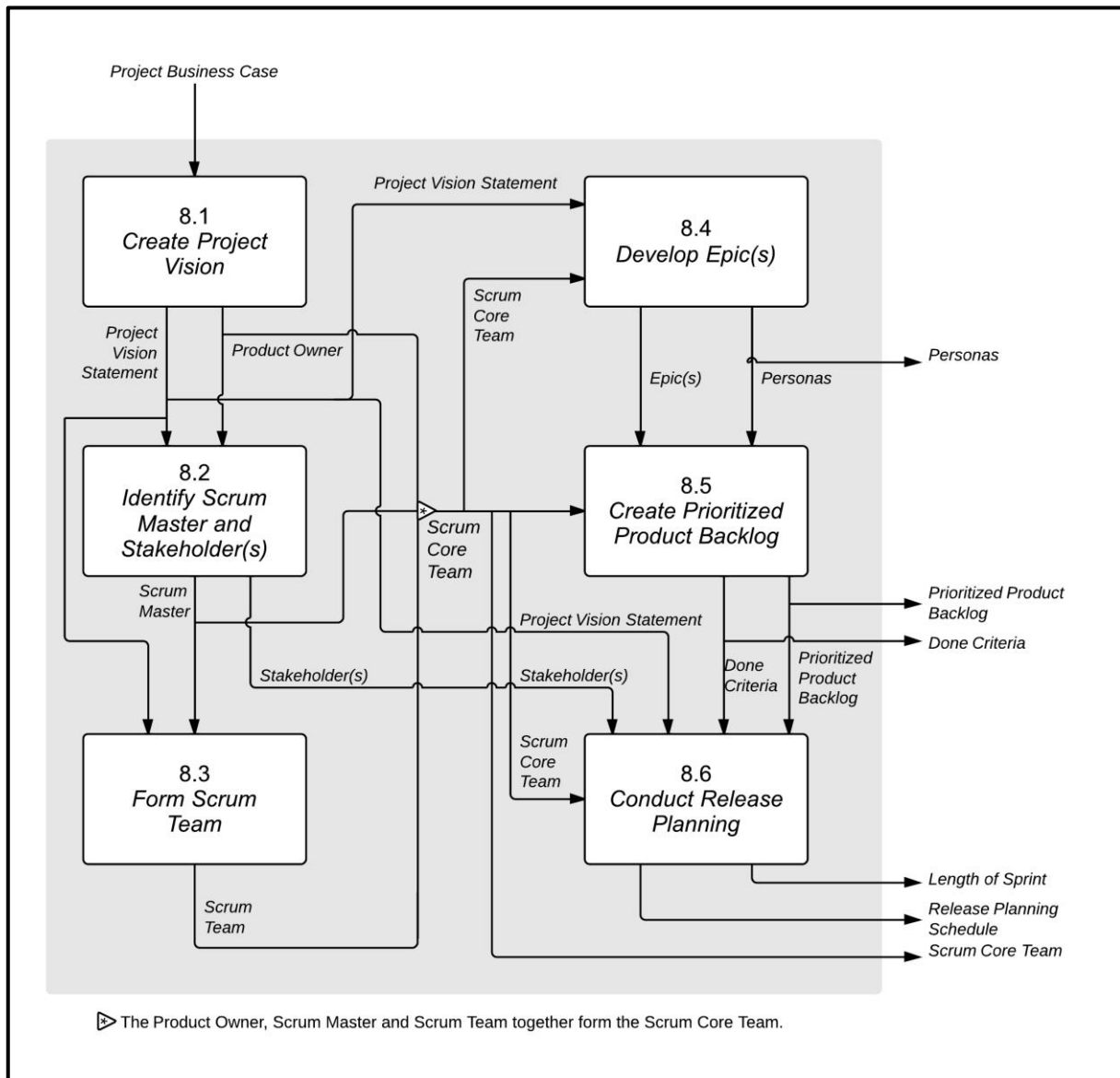


Figura 8-16: *Initiate Phase*—Diagrama de flujo de datos

9. PLANEAR Y ESTIMAR

La fase de *Plan and Estimate* consiste en procesos relacionados con la planificación y las tareas de estimación, que incluyen *Create User Stories*, *Approve*, *Estimate*, *and Commit User Stories*, *Create Tasks*, *Estimate Tasks*, y *Create Sprint Backlog*.

Plan and Estimate, tal como se define en *Una guía para el conocimiento de Scrum (Guía SBOK™)*, es aplicable a los siguientes:

- *Portfolios*, *programs* y/o *projects* de cualquier sector
- *Products*, servicios o cualquier otro resultado que se les entregará a los *stakeholders*
- *Projects* de cualquier tamaño y complejidad

El término "producto" en la *Guía SBOK™* puede referirse a un producto, servicio, o cualquier otra prestación. Scrum se puede aplicar de manera efectiva a cualquier proyecto en cualquier industria - desde pequeños proyectos o equipos con tan sólo seis miembros por equipo, hasta proyectos grandes y complejos que cuentan con cientos de miembros por equipo.

A fin de facilitar la mejor aplicación del marco de Scrum, en este capítulo se identifican las entradas, herramientas y salidas de cada proceso, ya sea como "obligatorio" u "opcional". Las entradas, herramientas y salidas indicadas por *asteriscos (*)* son obligatorias, mientras que las que no tienen *asteriscos* son opcionales.

Se recomienda que el *Scrum Team* y aquellas personas que recién comienzan a aprender sobre el marco y los procesos de Scrum, se centran principalmente en las aportaciones obligatorias, las herramientas y los productos; mientras que los *Product Owners*, *Scrum Masters*, y otros practicantes con experiencia de Scrum se deberían de esforzar por alcanzar un conocimiento más profundo de la información ofrecida en este capítulo. También es importante darse cuenta de que, aunque todos los procesos se definen de forma única en la *Guía SBOK™*, no necesariamente se llevan a cabo de forma secuencial o por separado. A veces, puede ser más apropiado combinar algunos procesos, dependiendo de los requisitos específicos de cada proyecto.

Este capítulo está escrito desde la perspectiva de un *Scrum Team* trabajando en un *Sprint* para producir entregables (*deliverables*) como parte de un proyecto más amplio. Sin embargo, la información que se describe es igualmente aplicable a proyectos enteros, programas y *portfolios*. Información adicional relacionada con el uso de Scrum para proyectos, programas y *portfolios* está disponible en los capítulos 2 al 7, que cubren los principios y aspectos de Scrum.

Figura 9-1 Proporciona una descripción general de los procesos de *Plan and Estimate Phase*, que son los siguientes:

9.1 Create User Stories—En este proceso, *User Stories* y sus afines *User Story Acceptance Criteria* se crean. Los *User Stories* son generalmente escritos por el *Product Owner* y están diseñados para asegurar que los requisitos del *Customer* estén claramente representados, y que puedan ser plenamente comprendidos por todos los *stakeholders*. *User Story Writing Workshops* se pueden llevar a cabo lo cual implica que los miembros del *Scrum Team* creen *User Stories*. Estos *User Stories* se incorporan en el *Prioritized Product Backlog*.

9.2 Approve, Estimate, and Commit User Stories—En este proceso, el *Product Owner* aprueba los *User Stories* para un *Sprint*. Luego, el *Scrum Master* y el *Scrum Team* estiman el esfuerzo necesario para desarrollar la funcionalidad descrita en cada *User Story*. Por último, el *Scrum Team* se compromete a entregar los requisitos del *Customer* mediante *Approve, Estimate, and Commit User Stories*.

9.3 Create Tasks—En este proceso, los *Approved, Estimated, and Committed User Stories* se dividen en tareas específicas y se compilan en un *Task List*. A menudo, un *Task Planning Meeting* se lleva a cabo con este fin.

9.4 Estimate Tasks— En este proceso, el *Scrum Core Team*, en las reuniones de *Task Estimation*, estima el esfuerzo necesario para realizar cada tarea del *Task List*. El resultado de este proceso es un *Effort Estimated Task List*.

9.5 Create Sprint Backlog—En este proceso, el *Scrum Team* tiene un *Sprint Planning Meeting* donde el grupo crea un *Sprint Backlog* que contiene todas las tareas que deben completarse en el *Sprint*.

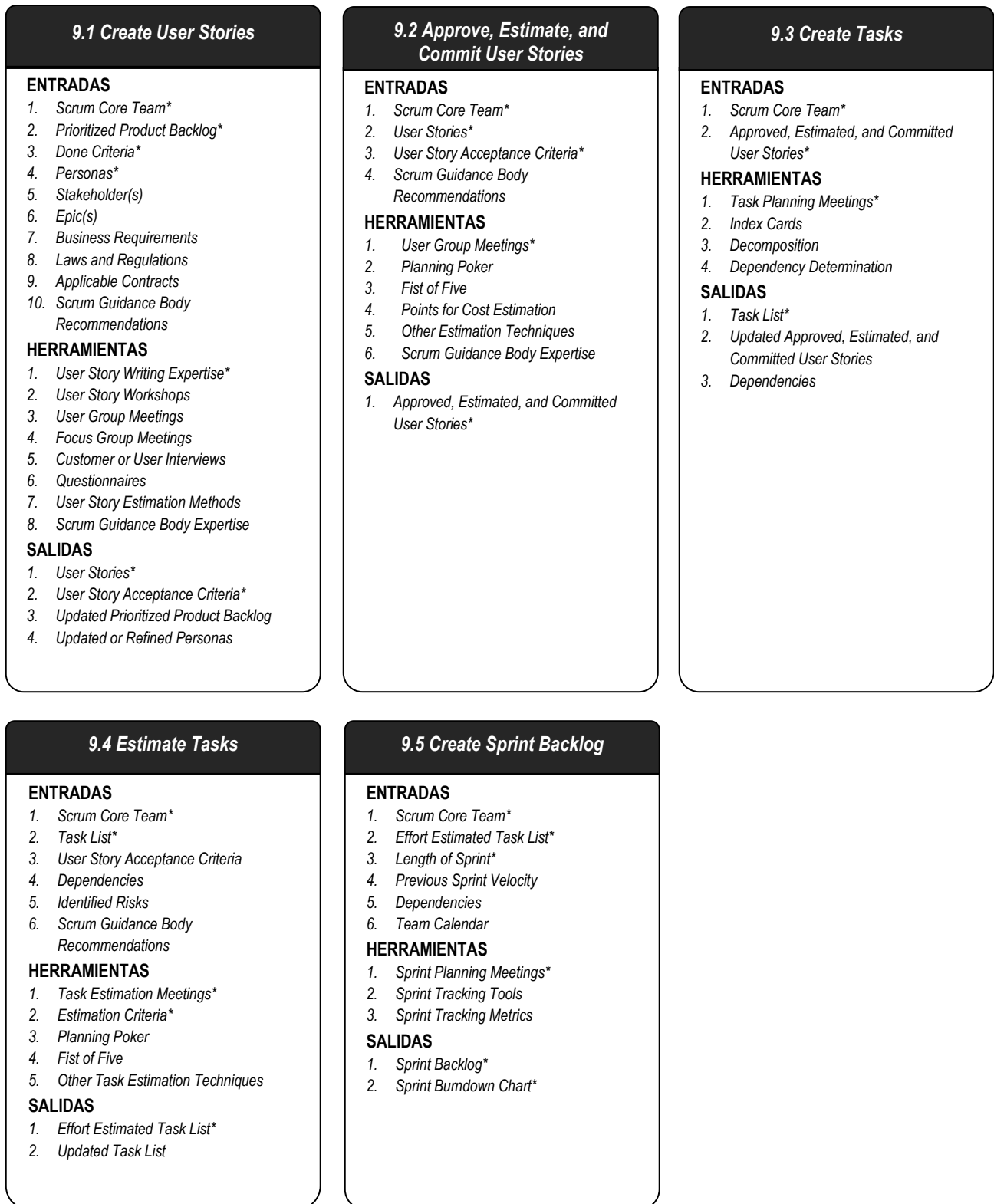


Figura 9-1: Plan and Estimate General

Nota: Los asteriscos (*) denotan una entrada, herramientas o salida "obligatoria" para el proceso correspondiente.

La figura 9-2 muestra las entradas obligatorias, las herramientas y las salidas de los procesos en la fase *Plan and Estimate*.

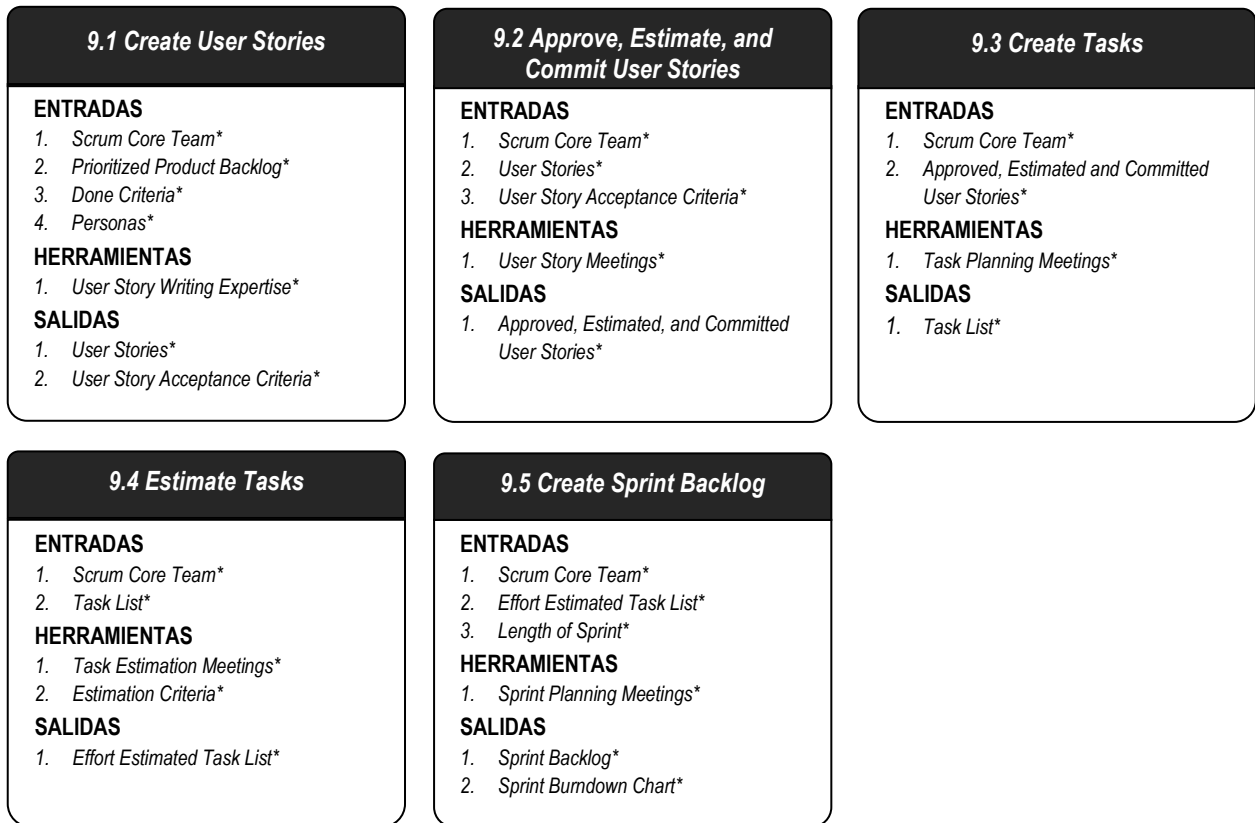


Figura 9-2: *Plan and Estimate* General (Esenciales)

Nota: Los asteriscos (*) denotan una entrada, herramienta o salida "obligatoria" para el proceso correspondiente.

9.1 Create User Stories

La figura 9-3 muestra todas las entradas, las herramientas y las salidas para *Create User Stories*.

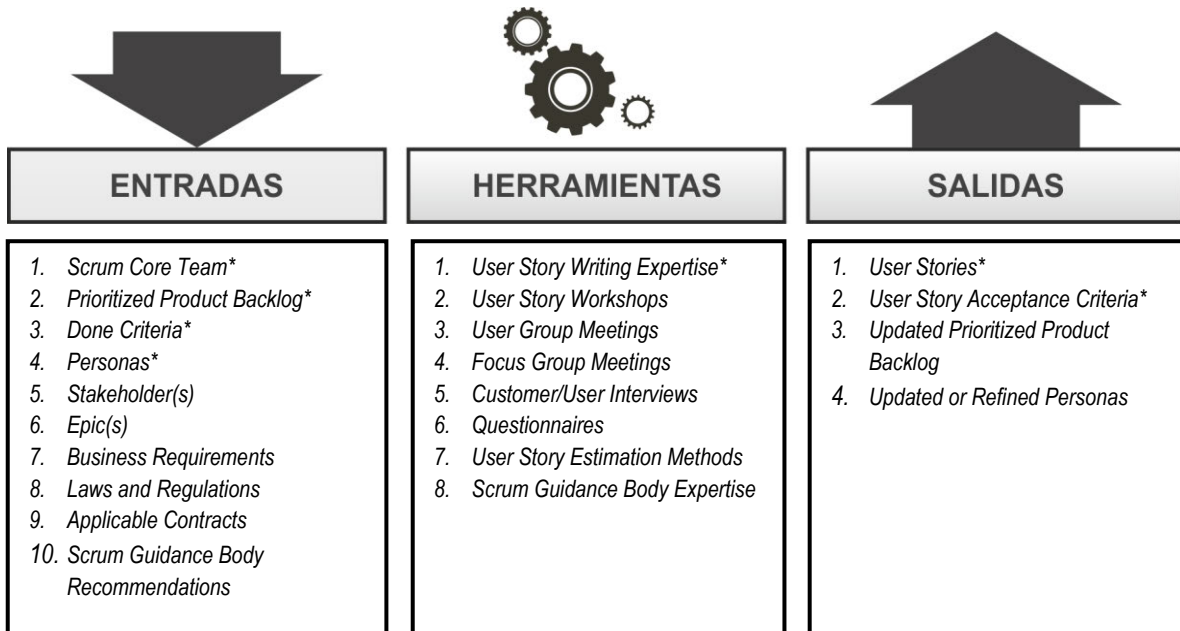
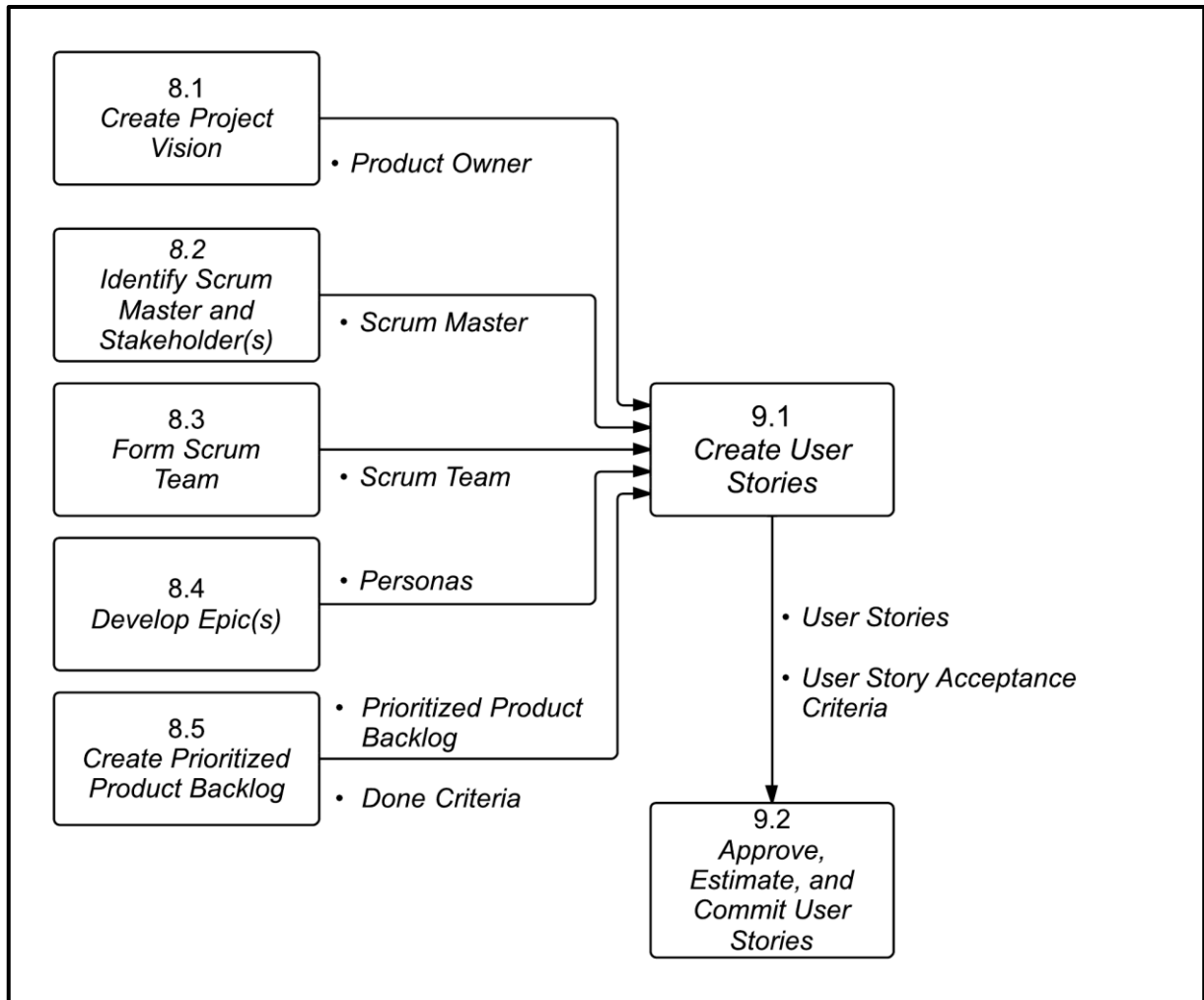


Figura 9-3: Create User Stories—Entradas, Herramientas, y Salidas

Nota: Los asteriscos (*) denotan una entrada, herramienta o salida "obligatoria" para el proceso correspondiente.

Figura 9-4: *Create User Stories*—Diagrama de Flujo de Datos