

SAFe		
No.	Práctica	Definición
1	Desarrollar en colaboración la intención de la solución	Solution Intent es el repositorio para almacenar, administrar y comunicar el conocimiento del comportamiento actual y previsto de la solución. Las soluciones complejas y altamente confiables requieren y crean grandes cantidades de información técnica. Gran parte de ella refleja los requisitos y diseño de la solución: características, capacidades, historias, requisitos no funcionales, arquitectura del sistema, modelos y diseños a nivel de dominio, interfaces de sistema, especificaciones del cliente, pruebas, resultados de pruebas y trazabilidad. Una de las entradas de la intención de solución es la reutilización de elementos de sistemas previamente desarrollados, esto con el propósito de que el conocimiento fijo no comience desde cero.
2	Abordar la implementación de SAFe en el gobierno y personas influyentes clave	Los líderes deben realizar el curso SAFe para el gobierno, este puede ayudar a abordar impedimentos frecuentes para adoptar Lean-Agile en el gobierno, alinear el presupuesto, el pronóstico, la contratación, la gobernanza, el cumplimiento y muchos más principios y prácticas Lean-Agile. Los líderes deben tomarse el tiempo, como grupo, para explorar, analizar y validar en colaboración los desafíos que enfrenta la organización. Deben evaluar cómo el sistema actual contribuye a esos desafíos y aprender la mentalidad, los principios y las prácticas que deberán adoptar para lograr los resultados transformadores que imaginan
3	Juntar, entrenar y estimular a un grupo con el poder suficiente para liderar un esfuerzo de cambio	Juntar, entrenar y estimular a un grupo con el poder suficiente para liderar un esfuerzo de cambio, se necesitan líderes que puedan establecer la visión, mostrar el camino y eliminar impedimentos para cambiar.
4	Capacitar a ejecutivos, gerentes y líderes	Se debe capacitar y entrenar a profesionales, gerentes y agentes del cambio que puedan implementar cambios de proceso específicos
5	Crear un centro de excelencia Lean-Agile	El Centro de Excelencia Lean-Agile (LACE) es un pequeño equipo de personas dedicado a implementar la forma de trabajo SAFe Lean-Agile. La creación de un LACE es a menudo uno de los diferenciadores clave entre las empresas que practican Agile solo de nombre y aquellas totalmente comprometidas con la adopción de prácticas Lean-Agile y obtener los mejores resultados comerciales. Este grupo de trabajo se convierte en el punto focal y la fuente continua de inspiración y energía para las actividades de gestión del cambio.
6	Identificar el backlog de solución (proyecto)	El Backlog de la solución contiene las próximas capabilities (comportamientos de la solución) y Enablers (respalda las actividades de código, los componentes e infraestructura técnica necesarias para proporcionar funcionalidad comercial futura) del incremento de programa, y está destinado a avanzar la solución y construir su pista arquitectónica. Los elementos son el resultado de actividades de investigación y colaboración activa con diversas partes interesadas: Clientes, Propietarios de Negocios, Gestión de Productos, Propietarios de Productos, Arquitectos de Sistemas e ingenieros de soluciones.
7	Refinar y priorizar el backlog de solución	Refinar (Revisar y actualizar la definición de los elementos de trabajo pendiente y desarrollo de criterios de aceptación e hipótesis de beneficios, trabajar con los equipos para establecer estimaciones de viabilidad técnica y alcance, analizar formas de dividir los elementos de trabajo pendiente en trozos más pequeños de valor incremental, identificar los habilitadores necesarios para admitir nuevas características y capacidades, y establecer su asignación de capacidad) y priorizar el trabajo pendiente.
8	Identificar el backlog de programa	El backlog de programa contiene las próximas características destinadas a satisfacer las necesidades del cliente y ofrecer beneficios empresariales para un único tren de liberación ágil (ART). También contiene las características del habilitador necesarias para crear la pista de acceso arquitectónico. Este backlog de programa se construye en el evento de planificación del incremento de programa, donde se establecen los equipos y estos crean sus planes de entrega de tareas futuras de desarrollo de productos

9	Refinar y priorizar el backlog de programa	Refinar (Revisar y actualizar la definición de los elementos de trabajo pendiente y desarrollo de criterios de aceptación e hipótesis de beneficios, trabajar con los equipos para establecer estimaciones de viabilidad técnica y alcance, analizar formas de dividir los elementos de trabajo pendiente en trozos más pequeños de valor incremental, identificar los habilitadores necesarios para admitir nuevas características y capacidades, y establecer su asignación de capacidad) y priorizar el trabajo pendiente.
10	Construir, refinar y priorizar el backlog de equipo	El Backlog del equipo es un repositorio, el cual comprende Historias de usuarios, habilitadores, así como historias que surgen del contexto local del equipo y otros elementos de trabajo que representan todas las cosas que un equipo necesita hacer para cumplir con los objetivos del incremento de programa.
11	Identificar los ART	ART es un grupo de personas (típicamente de entre 50 a 125) trabajando de forma coordinada y alineada a los objetivos de negocio en un flujo constante de entrega de valor para un Value Stream definido para la organización. Identificar a un equipo formado de varios equipos ágiles, equipos de operaciones y especialistas (por ejemplo UX y Arquitectura) y un equipo coordinador a nivel de programa llamado el Release Management Team (RTM).
12	Identificar los ART de soluciones	Es la construcción organizativa utilizada para construir soluciones grandes y complejas que requieren la coordinación de múltiples trenes de liberación ágil (ART) , así como las contribuciones de los proveedores . Alinea los ART con una misión empresarial y tecnológica compartida utilizando la solución Visión , Backlog y Roadmap , y un Incremento de Programa (PI) alineado.
13	Llevar a cabo el evento de planificación antes y después del incremento de programa (nivel de soluciones)	Apoyan y coordinan los diversos ART involucrados en el Tren de soluciones. La planificación en este nivel superior ayuda a alinear el desarrollo de la Solución en su conjunto y proporciona dirección y visibilidad hacia dónde van los trenes al próximo PI.
14	Llevar a cabo la planificación del incremento de programa (programa y equipo)	Es un evento cara a cara llevado a cabo entre todos los implicados, con un timebox establecido y cuya finalidad es alinear a todos los equipos en el ART con una misión y visión compartidas, donde cada equipo identifica sus tareas o trabajo pendiente de iteración, dependencias y planes de iteración para ejecutar dicho trabajo bajo los objetivos iniciales del incremento de programa y los riesgos asociados a estos.
15	Planificación de iteraciones (equipo)	Es un evento en el cual todos los miembros del equipo determinan la cantidad del Backlog de Equipo que pueden comprometerse a entregar durante la próxima Iteración. El equipo resume el trabajo como un conjunto de Objetivos de iteración a los que están comprometidos. Estos objetivos de iteración alinean a los equipos entre sí, ayudan a identificar las dependencias y proporcionan el contexto necesario para desarrollar una resolución.
16	Construir y mantener la tubería de entrega continua	Continuous Delivery Pipeline (CDP) representa los flujos de trabajo, las actividades y la automatización necesarios para guiar una nueva pieza de funcionalidad desde la ideación hasta el lanzamiento bajo demanda de valor para el usuario final. Comprende 4 aspectos: (i) Continuous Deployment (CD) es el proceso que toma las Features validadas en un entorno de staging y las implementa en el entorno de producción, en donde se prueban y se las deja listas para el release. (ii) Continuous Exploration (CE) es el proceso que impulsa la innovación y fomenta la alineación con lo que se debe construir mediante la exploración continua de las necesidades del mercado y del customer, y la definición de una Visión, un Roadmap y un conjunto de Features para una solución que satisfaga esas necesidades. (iii) Continuous Integration (CI) es el proceso de tomar Features del Program Backlog y desarrollar, probar, integrar y validarlos en un entorno de representación donde están listos para la implementación y el release, y (iv) lanzamiento bajo demanda; que es el proceso que implementa una nueva funcionalidad en la producción y la libera de manera progresiva o inmediata a los Clientes según la demanda

17	Ejecutar el incremento de programa e iteraciones de equipos	Los equipos ágiles gestionan su trabajo a lo largo del período de ejecución del incremento de programa, donde cada equipo colabora para definir, construir y aprobar las historias a las que se comprometieron durante la planificación de iteraciones, además de realizar un seguimiento del progreso de la iteración y mejorar el flujo de valor mediante el uso de tableros de historia y Kanban y eventos diarios de pie (DSU), todo esto a nivel de equipos individuales. De igual manera, los equipos deben coordinarse entre sí llevando a cabo reuniones como Scrum of Scrums para gestionar las dependencias de los ART y proporcionar visibilidad sobre el progreso y los impedimentos; además del evento de sincronización PO, en la que participan product owners y administradores de productos y cuyo propósito es obtener la visibilidad de lo bien que el ART está progresando hacia el cumplimiento de los objetivos del incremento de programa, discutir problemas u oportunidades y también puede ser útil para prepararse en el siguiente incremento de programa. Ofrecer soluciones complejas y de alta calidad requiere una intensa cooperación y colaboración entre equipos. Para admitir esto, los equipos trabajan en una cadencia ART común y publican y comunican objetivos de iteración al principio de cada iteración. También actualizan otros equipos durante la sincronización de ART y administran activamente las dependencias interactuando con los miembros del equipo de otros equipos.
18	Demostración de la solución	La demostración de la solución es donde los resultados de los esfuerzos de desarrollo combinados de múltiples trenes de liberación ágil (ART), junto con las contribuciones de los proveedores y otros participantes, la solución muestra a los clientes y otras partes interesadas. Esta demostración es un momento crítico para el Tren de soluciones, una oportunidad para la evaluación objetiva y la retroalimentación. También es un momento para celebrar los logros del último IP y proporciona información esencial para las decisiones de inversión a corto plazo. Como única medida tangible de progreso, mitiga el riesgo de inversión.
19	Demostración del sistema	La demostración del sistema se produce al final de cada iteración, donde se prueba y evalúa la solución completa en un contexto similar a la producción (a menudo escenificación) para recibir comentarios de las partes interesadas. La demostración del sistema es un evento importante que proporciona una vista integrada de las nuevas características para la iteración más reciente entregada por todos los equipos en el tren de lanzamiento ágil (ART). Cada demostración brinda a las partes interesadas de ART una medida objetiva del progreso durante un Incremento de Programa (PI)
20	Revisión de iteración	Es un evento basado en la cadencia, donde cada equipo inspecciona el incremento al final de cada iteración para evaluar el progreso y luego ajusta su trabajo pendiente para la próxima iteración
21	Inspeccionar y adaptar	Inspect and Adapt (I&A) es un evento significativo que se realiza al final de cada Program Increment (PI), el cual consta de tres partes: (i) Demostración del sistema, donde el tren demuestra y evalúa el estado actual de la Solution, (ii) Medición cuantitativa y cualitativa, donde los equipos revisan colectivamente cualquier métrica cuantitativa y cualitativa que hayan acordado recopilar y, a continuación, discutan los datos y las tendencias y (iii) Taller retrospectivo y de resolución de problemas donde los equipos reflexionan e identifican elementos en el backlog para mejorar a través de este taller. El resultado es un conjunto de elementos de trabajo pendiente de mejora que entran en el trabajo pendiente del programa para el siguiente evento de PI Planning.
22	Retrospectiva de iteración	Es una reunión en la que los miembros del Agile Team evalúan los resultados de la Iteration, revisan sus prácticas e identificar maneras de mejorar

LeSS		
No.	Práctica	Definición
1	Sprint Planning one	Sprint Planning One se enfoca en: la selección de elementos de mayor prioridad que cumplen la definición de 'listo' (elementos ofrecidos por el Propietario del producto), identificar oportunidades para el trabajo multi-equipo, resolver dudas y preguntas, y definir el Objetivo Sprint. El Sprint Planning One es una reunión para todos los equipos juntos donde deciden qué equipo trabajará en qué elementos. Cuando son menos de 8 equipos, se manejan como equipos de características, de lo contrario se crea un estructura adicional llamada 'área de requisitos', la cual conduce a diferentes vistas del trabajo pendiente y cada una de estas tiene varios equipos de características las cuales trabajan bajo un backlog de área que es priorizado por un propietario de producto de área especialista en el producto.
2	Sprint Planning two de equipo y multi-equipo	Sprint Planning two es una reunión separada por cada equipo donde se enfocan en crear un plan de trabajo para "hacer" para cada elemento. Los elementos y el plan de acción o tareas comprenden el Backlog de Sprint. Los equipos que identificaron dependencias en sus historias construyen sus backlogs de equipo en conjunto con el fin de acordar puntos de integración, coordinación y sincronización para ellas.
3	Actividades de desarrollo que apoyan la coordinación y la entrega continua	Los equipos se dedican a desarrollar los ítems seleccionados y que se encuentran en su sprint backlog de equipo. Se maneja una integración constante de código, cargado con las pruebas automatizadas y control de versiones, con el fin de encontrar y resolver a tiempo conflictos de integración. Se maneja una filosofía de pruebas de aceptación automatizadas e integración continua para evitar tener un equipo de integración y pruebas. La coordinación y comunicación entre los equipos con dependencias es vital, se recomienda la coordinación descentralizada e informal a la centralizada (comunicación directa entre desarrolladores del mismo equipo o inter equipo). Se maneja la definición de hecho general con el fin de tener siempre un incremento potencialmente entregable.
4	Retrospectiva general	La Retrospectiva Sprint, implica inspeccionar y adaptar con respecto al proceso LeSS y el entorno. Es un espacio para que los equipos discutan las oportunidades de mejora encontradas en sus retrospectivas de equipo. Posteriormente, los participantes se centran los problemas y se discuten soluciones de mejora.
5	Actividades de coordinación	Los equipos hacen la reunión de Scrum daily por equipo, cuando existen dependencias, un integrante del equipo con el que tienen dependencias participa de dicho daily scrum y luego comenta con su equipo lo aprendido. Se celebra también una reunión general semanal (o según se acuerde) (tipo Scrum de scrum u open space) para coordinar los equipos, se hace con representantes de los equipos y el (o los) Scrum master que se encarga de guiar dicha reunión. Además, se hacen reuniones de las comunidades de práctica, si son necesarias, por ejemplo, si la comunidad de testing (compuesta por voluntarios de todos los equipos) quiere proponer una nueva forma de automatización de pruebas.
6	Refinación general del product backlog (PBR Product Backlog Refinement)	Se revisa la priorización del backlog y se actualiza en caso de ser necesario teniendo en cuenta el impacto de marketing o impacto al cliente o riesgos técnicos, del negocio, etc. Una oportunidad para la adaptación. Puede ser hecha durante un sprint en curso. También permite aumentar el conocimiento compartido ya que se discuten los ítems que están estrechamente relacionados. Se realizan estimaciones de Scrum Póker. Pueden surgir un evento PBR multi-equipo cuando los equipos noten ítems que estén estrechamente relacionados o para aumentar el conocimiento compartido entre ellos. El propietario del producto no trabaja solo en el refinamiento del product backlog sino que recibe el apoyo de los integrantes de los otros equipos los cuales trabajan en conjunto con los clientes / usuarios y otras partes interesadas.

7	PBR Multi-equipo y de equipo	Los equipos realizan un refinamiento del product backlog con expertos y varios equipos para aumentar la comprensión compartida, aprovechar las oportunidades de coordinación, alinear estimaciones y aumentar la adaptabilidad entre los equipos. Los asistentes incluyen a todos los miembros de todos los equipos participantes, y también pueden incluir expertos en la materia, usuarios, clientes y el Propietario del producto
8	Más actividades de desarrollo	Los equipos desarrollan código que se integra de forma continua combinada con la automatización completa de pruebas. Se detienen y arreglan cuando se rompe la construcción, trabajando hacia su objetivo de perfección de tener un producto listo para enviar que puedan entregar continuamente a los clientes. Por lo tanto, cuando Sprint está a punto de terminar y los equipos se están preparando para unirse a la Revisión de Sprint, no hay una oleada de esfuerzos tardíos para integrar y probar un gran lote de código: se ha integrado y probado todo el tiempo.
9	Revisión de Sprint (Sprint Review)	Los miembros de los equipos realizan la revisión del sprint completo (o entrega) junto con los stakeholders (clientes, representantes de servicio al cliente, usuarios), acompañados del PO y el o los SM. Cuando muchos elementos por explorar se realiza a manera de feria de ciencias, unos exponen otros revisan y otros toman notas.
10	Retrospectiva de equipo	Finalizada la jornada de revisión de Sprint, cada equipo realiza su propia retrospectiva del sprint e identifican las oportunidades de mejora, como por ejemplo, una mejor PBR o aplicar tempranamente un taller de diseño para no tener complicaciones durante el desarrollo de un sprint por problemas de diseño.
11	Priorización del Backlog de Área	El propietario del producto (PO) agrupa cada elemento de la cartera de productos bajo exactamente una categoría de requisitos: su área de requisitos. Después de esto, genera diferentes vistas sobre el Backlog general del producto, llamado Backlog de área. Las tareas pendientes del área son priorizadas por un propietario del producto del área que se especializa en parte del producto, desde la perspectiva del cliente.

Nexus		
No.	Práctica	Definición
1	Refinar la lista de producto	Descomponer la lista de producto de tal forma que las dependencias que se identifiquen, y se remuevan o minimicen. "Crear finas rodajas". Los elementos de la Pila del Producto se dividen en rodajas finas de funcionalidad y el equipo más adecuado para hacer el trabajo debería identificarse tan pronto como sea posible.
2	Planificación del Sprint Nexus (General)	Representantes apropiados de cada Equipo Scrum se reúnen para discutir y revisar la Pila del Producto refinada. El Propietario del Producto proporciona el conocimiento del dominio y guía las decisiones de selección y priorización. Ellos seleccionan los elementos de la Pila del Producto para cada equipo. A continuación, cada Equipo Scrum planea su propio Sprint, interactuando con los otros equipos en la medida en que sea apropiado. El resultado es un conjunto de Objetivo del Sprint que se alinean con el Objetivo Nexus general, con la Pila del Producto del Sprint de cada Equipo Scrum y con la única Pila del Producto Nexus. Esta última hace que los elementos seleccionados de la Pila del Producto de cada Equipo Scrum y cualquier dependencia que haya entre ellos, sean transparentes. Todos los elementos de la Pila del Producto seleccionados para el Sprint y sus dependencias deberían visualizarse en la Pila del Producto del Sprint Nexus.

3	Planificación del Sprint Nexus (Equipos)	Representantes apropiados de cada Equipo Scrum se reúnen para discutir y revisar la Pila del Producto refinada. El Propietario del Producto proporciona el conocimiento del dominio y guía las decisiones de selección y priorización. Ellos seleccionan los elementos de la Pila del Producto para cada equipo. A continuación, cada Equipo Scrum planea su propio Sprint, interactuando con los otros equipos en la medida en que sea apropiado. El resultado es un conjunto de Objetivo del Sprint que se alinean con el Objetivo Nexus general, con la Pila del Producto del Sprint de cada Equipo Scrum y con la única Pila del Producto Nexus. Esta última hace que los elementos seleccionados de la Pila del Producto de cada Equipo Scrum y cualquier dependencia que haya entre ellos, sean transparentes. Todos los elementos de la Pila del Producto seleccionados para el Sprint y sus dependencias deberían visualizarse en la Pila del Producto del Sprint Nexus.
4	Trabajo de desarrollo	Todos los equipos desarrollan software, integrando frecuentemente su trabajo en un ambiente común que puede probarse para asegurar que la integración se haya realizado. Las actividades de desarrollo son apoyadas por el equipo de integración Nexus, que está compuesto por el P.O., el S.M. y un conjunto de profesionales que tienen habilidades en el uso de herramientas, distintas prácticas y en el campo general de la ingeniería de sistemas. Los miembros del equipo de integración Nexus se aseguran de que las prácticas y herramientas se implementen, se entiendan y se usen para detectar dependencias y para integrar frecuentemente todos los artefactos hasta una definición de "Hecho".. Además, ayudan a resolver cualquier restricción técnica y no técnica entre equipos que pueda impedir la capacidad de un Nexus de entregar un Incremento Integrado constantemente. El Equipo de Integración Nexus sirve también de entrenador (Coach) a los Equipos Scrum en los estándares de desarrollo, infraestructura y arquitectura requeridos por la organización para asegurar el desarrollo de Incrementos Integrados de calidad.
5	Revisión del sprint Nexus	La Revisión del Sprint Nexus se realiza al final del Sprint para proporcionar retroalimentación sobre el Incremento Integrado que el Nexus ha construido durante el Sprint y para adaptar los Elementos de la Pila del Producto en caso de ser necesario. Una Revisión del Sprint Nexus reemplaza las Revisiones de Sprint de los Equipos Scrum individuales porque el Incremento Integrado completo es el foco para obtener retroalimentación de los interesados. Puede no ser posible mostrar todo el trabajo terminado en detalle. Podrían necesitar técnicas para maximizar la retroalimentación de las partes interesadas. El resultado de la Revisión del Sprint Nexus es una Pila del Producto revisada
6	Retrospectiva del Sprint Nexus	La Retrospectiva del Sprint Nexus es un evento en el cual se reúnen los representantes apropiados de cada Equipo Scrum para identificar desafíos compartidos. Es una oportunidad formal para que el Nexus se inspeccione y adapte a sí mismo y cree un plan de mejoras que se ejecuten a partir del próximo Sprint para asegurar así la mejora continua. Los representantes se reúnen para identificar los desafíos compartidos al final de cada Sprint. Luego, cada equipo hace su retrospectiva individual, y finalmente los representantes se reúnen nuevamente para discutir soluciones y acciones de mejora al Nexus.

DAD		
No.	Práctica	Definición
1	Organizar equipos	Un programa de TI es un gran equipo de entrega de TI compuesto por dos o más subequipos (también llamados escuadrones). El propósito de la gestión de programas es coordinar los esfuerzos de los subequipos para asegurar que trabajen juntos eficazmente hacia el objetivo común de producir una solución consumible para sus partes interesadas. Hay tres estrategias comunes para organizar los equipos de entrega dentro de un programa (equipos de características; donde cada equipo es responsable de implementar un requisito funcional, como una historia de usuario o un caso de uso, de extremo a extremo, equipos de componentes; donde cada subequipo es responsable de uno o más subsistemas o módulo, y código abierto interno; en el cual un componente o subsistema se desarrollará a través de un método de código abierto), cada uno de los cuales tiene ventajas y desventajas. Además de los equipos de entrega, en un programa grande es probable que encuentre la necesidad de equipos de liderazgo (el equipo propietario del producto, el equipo propietario de la arquitectura y el equipo de coordinación/gestión de productos) compuesto por los propietarios de productos, propietarios de arquitectura y jefes de equipo de los equipos de entrega, respectivamente. Estos equipos de liderazgo son responsables de la coordinación de trabajo/requisitos, coordinación técnica y coordinación de gestión dentro del programa, respectivamente.
2	Priorizar el trabajo	Es necesario priorizar el trabajo realizado por los equipos, incluidos los nuevos requisitos y la corrección de defectos. Hay varias maneras de priorizar el trabajo, como por valor empresarial, por riesgo, por gravedad (en el caso de defectos de producción), o por trabajo más corto ponderado primero (wsjf) por nombrar algunas estrategias. La priorización es una actividad continua a lo largo del ciclo de vida y es responsabilidad de los propietarios de sus productos.
3	Asignar el trabajo	Los elementos de trabajo deben asignarse a los equipos de entrega o a los esfuerzos de código abierto en el caso de programas que incluyan componentes internos de código abierto a lo largo del ciclo de vida. El tipo de trabajo y el enfoque del subequipo son los principales determinantes de cómo se asigna el trabajo. Sin embargo, la capacidad del equipo y las preocupaciones de equilibrio de carga, por ejemplo, un equipo se ha quedado sin trabajo o un equipo actualmente tiene demasiado trabajo, también se tendrán en cuenta al asignar nuevo trabajo. La asignación de trabajo es responsabilidad de los propietarios de productos, aunque la supervisión de la capacidad del equipo suelen ser realizadas por el director del programa y los clientes potenciales del equipo. A pesar de todo, estas actividades deben ser realizadas de forma colaborativa por las personas disponibles en ese momento.
4	Plan de programa	Los programas tradicionales a menudo se planifican anualmente o incluso ad hoc. Programas ágiles, al menos los disciplinados, tienden a usar Rolling Wave Planning o Planificación Gradual, la cual hace referencia a una técnica de planificación que consiste en planificar en detalle el trabajo que se tiene que realizar en el corto plazo, mientras que el trabajo a más largo plazo se planifica a un nivel más alto de la EDT (Estructura de Desglose del Trabajo, WBS en inglés). A medida que avanza el desarrollo del proyecto, y se conocen más detalles de la siguiente fase, se va planificando en detalle lo que anteriormente solo estaba planificado a alto nivel. Esta técnica es muy útil en proyectos en los que hay un grado de incertidumbre elevado y a medida que se van conociendo detalles de las siguientes fases podemos ir planificando los detalles.

5	Coordinar equipos	Hay varias maneras en que los subequipos pueden coordinarse entre sí. Por ejemplo, podrían elegir tener reuniones de coordinación entre equipos (también llamadas Scrum of Scrums (SoS)); podían visualizar el trabajo a través de paneles de tareas, paneles de control de equipo y otros radiadores de información, como un muro de modelado; podrían elegir tener sesiones de planificación de "sala grande" en las que todos los miembros del equipo participen o sesiones de modelado ágiles de "sala pequeña" en las que participen un subconjunto de personas; o incluso reuniones tradicionales (o ágiles) de puntos de control. Todas estas estrategias tienen sus ventajas y desventajas, y todas pueden ser aplicadas por los diversos tipos de equipos mencionados anteriormente.
6	Coordinar horarios	Hay varias estrategias que un programa puede adoptar para coordinar los horarios entre los subequipos. Lo más fácil conceptualmente, aunque a menudo más difícil de implementar en la práctica, es tener todos los subequipos en la misma cadencia (por ejemplo, cada subequipo tiene una iteración de dos semanas). Esto es lo que prescriben SAFe y LeSS. Otra opción es tener cadencias multiplicadoras donde los horarios de los subequipos se alinean cada tan a menudo. Por ejemplo, una vez trabajamos con un programa grande donde algunos subequipos tenían una iteración de una semana, algunos tenían una iteración de dos semanas y algunos tenían una iteración de cuatro semanas. También hemos visto otro equipo donde los subequipos tenían iteraciones de una, dos o tres semanas que proporcionaban alineación de terminaciones de iteración cada seis semanas. Lo más común, aunque rara vez se discute, es que los subequipos tengan cadencias dispares. Esto se garantiza que se produce cuando los equipos siguen diferentes ciclos de vida. Por ejemplo, cuando algunos subequipos siguen el ciclo de vida ágil/básico basado en Scrum que tiene iteraciones, sin embargo, otros subequipos siguen los ciclos de vida de entrega lean o continuos que no tienen iteraciones, entonces tiene un desafío de alineación. O si tienes subequipos que adoptan cualquier longitud de iteración que les guste (hemos visto algunos programas con subequipos con dos, tres, cuatro y a veces incluso longitudes de iteración de cinco semanas) entonces también en efecto tienen cadencias dispares.
7	Programar versiones de solución	Los programas deben programar sus propias versiones, de acuerdo con la estrategia de administración de versiones de su organización, que implica la coordinación entre los subequipos. Cuando las cadencias de los subequipos están (razonablemente) alineadas, entonces es más fácil coordinar las versiones de producción. Por ejemplo, cuando todos los subequipos tienen iteraciones de dos semanas (o al menos los subequipos con iteraciones lo hacen), podrían lanzarse a producción cada dos semanas. En el caso de las cadencias del multiplicador, existe la posibilidad de liberar en producción cada vez que se alinean los finales de iteración.
8	Negociar dependencias funcionales y técnicas	Una responsabilidad importante del equipo de propietarios del producto es administrar las dependencias funcionales entre el trabajo que realizan varios subequipos. Existen estrategias para gestionar las dependencias entre dos subequipos ágiles, entre un subequipo ágil y un subequipo magro, e incluso entre un subequipo ágil/lean y un subequipo tradicional (esto no es ideal, pero a veces sucede). Del mismo modo, una responsabilidad importante del equipo de Propietarios de Arquitectura es trabajar a través de dependencias técnicas dentro de la solución que está desarrollando el programa.
9	Gobernar el programa	El programa debe regirse, tanto internamente como dentro del mismo programa, mientras sigue operando bajo los auspicios de la estrategia general de gobierno de TI de su organización. El gobierno de TI al que refiere Disciplined Agile se define como "el liderazgo, las estructuras organizativas y los procesos simplificados para permitir que TI trabaje como socio para mantener y ampliar la capacidad de la organización de producir un valor significativo para sus clientes". Las métricas a nivel de programa, particularmente aquellas que siguen el progreso de los sub-equipos y la calidad que se entrega, son vitales para una coordinación exitosa dentro del programa. Los programas, debido a su tamaño y a que generalmente son de mayor riesgo, a menudo tienen requisitos de informes más rigurosos para la alta gerencia a fin de proporcionarles una mayor transparencia, por ende el tablero de un programa a menudo tiene una colección más robusta de medidas en exhibición.



10	Asegurar la preparación para la producción	Se realiza con el fin de cumplir el objetivo de transición, cuyo objetivo principal es implementar con éxito su solución en la producción (o en el mercado). Asegurar la preparación para la producción busca determinar si una solución está lista para desplegarse o enviarse. Asegurar la preparación para la producción tiene dos objetivos: (i) Asegurar la preparación técnica, donde se realizan pruebas alfa, preparación de migración de datos, finaliza el ciclo de vida de testing y reparaciones, se finaliza el plan de desarrollo y se finaliza la documentación; y (ii) Asegurar la preparación de los interesados donde se comunica a los stakeholders el despliegue, se prepara el ambiente de soporte y se entrena/educa a los stakeholders en la solución que se entrega.
11	Implementar la solución	Se realiza con el fin de cumplir el objetivo de transición, cuyo objetivo principal es implementar con éxito su solución en la producción (o en el mercado). Desplegar/Implementar la solución, describe el enfoque para lanzar / enviar la solución. Desplegar/Implementar la solución tiene 4 objetivos: (i) Automatizar el despliegue, donde se aplican técnicas de desarrollo continuo (CD); (ii) Definir la estrategia de lanzamiento, como por ejemplo las soportadas por Disciplined Devops: Liberar ventanas, Liberar tren, Lanzamiento continuo; (iii) Lanzamiento en producción, donde se verificará el qué se debe hacer para implementar la solución (backups, migración de datos, despliegue); y (iv) Validación del lanzamiento, donde se plantea el cómo determinar si la implantación fue exitosa.
12	Apoyar a los miembros del equipo	Ayudar a las personas a mejorar su conjunto de habilidades con la ayuda de entrenamiento, mentorías, comunidades de práctica, etc.; retroalimentación continuo, autoevaluación y revisión del líder del equipo
13	Aproveche y mejore la infraestructura existente	Aprovechar y mejorar los activos organizativos existentes, adopte y desarrolle una orientación empresarial, trabaje con funcionalidades, datos y activos heredados.
14	Evolucionar la forma de trabajo	Identificar y evolucionar la forma cómo los equipos trabajarán juntos. (Formar entorno de trabajo y mejorar el proceso de equipo y el entorno). Inicialmente se debe organizar un espacio de trabajo físico de acuerdo a la localización entre equipos, seguido, se debe escoger los estilos de comunicación y colaboración, puede ser cara a cara, videoconferencias, chats grupales y/o correos. Por otra parte, cada equipo puede elegir la forma de trabajo; ya sea ágil o tradicional, también se debe tener en cuenta que el trabajo de cada equipo debe ser visible para todos, identificar, implementar y compartir posibles mejoras, finalmente se debe organizar el entorno; análisis de código, gestión de la configuración, integración continua, pruebas unitarias, wikis, gestión de elementos de trabajo, etc.
15	Mejora continua	Permitir que las personas dentro de su organización compartan fácilmente sus aprendizajes de mejora entre sí de una manera sistemática. Se deben realizar actividades potenciales asociadas con la mejora continua ágil disciplinada. Estas actividades pueden ser realizadas por, o al menos apoyado por, un equipo de mejora de procesos (a veces denominado Grupo de Procesos de Ingeniería de Software, o SEPG). Algunas de estas prácticas serán realizadas por los Centros de Excelencia (CdE) y apoyadas por sus Comunidades de Práctica (CoPs) (si las hubiera). Una Comunidad de Práctica (CoP) es una colección de personas que comparten un oficio o profesión que se han unido para "aprender" unos de otros para desarrollarse y a menudo incluso la organización; Los CoP a menudo realizan las actividades llamadas por los factores de proceso Identificar mejoras, Compartir mejoras, Mejoras de captura y Equipos de soporte. Un Centro de Excelencia (CoE) es un grupo de personas con habilidades y experiencia especializadas cuyo trabajo es proporcionar liderazgo y difundir deliberadamente ese conocimiento dentro de su organización