

# VERIFICACIÓN Y VALIDACIÓN DE SOFTWARE

---

# Bibliografía

---

FARRELL-VINAY, P. Manage Software Testing. Auerbach Publications, 2008.

PERRY, W. Effective Methods for Software Testing. John Wiley & Sons, Inc., 1995.

O'DOCHERTY, M. Object Oriented Analysis and Design. Understanding System Development with UML 2.0, John Wiley & Sons, Inc., 2005.

# Validación

# Validación

---

- Clase anterior
  - Conceptos Verificación Validación
  - Proceso de desarrollo y proceso de validación de forma concurrente
  - Validación informal
  - Validación formal

# Validación

---

- Agenda
  - Pruebas Alfa.
  - Pruebas Beta.
  - Pruebas de Aceptación.

# Pruebas Alfa

# Validación

---

- Con el testeo Alpha ponemos en evidencia que el trabajo en el sistema de software está progresando.
- El objetivo es asegurar que se está desarrollando “algo” que probablemente es correcto y usable para el usuario.
- Por ejemplo, podemos implementar alguna funcionalidad central dentro del framework de nuestro diseño del sistema, y ponerlo a prueba de clientes seleccionados cuidadosamente y algunos colegas disponibles.

# Validación

---

- Si la retroalimentación es buena indicará que vamos por el camino correcto (obtendremos información correcta de mejoras).
- Si la retroalimentación es negativa debemos retroceder.
- A menudo se realiza contra un prototipo, una versión del sistema que es producida de forma rápida y barata.
- El prototipo de la prueba Alfa es desechado y reemplazado por una solución más elegante que se producirá durante un desarrollo a mayor escala.
- Si bien el código es desechado, se retiene el aprendizaje obtenido del problema y su solución, hasta este punto.



# Validación

---

- Liberación Alfa: Cualquier versión del software que no ha pasado la Prueba de sistema. Puede incluir liberaciones parciales. (Farrell-Vinay)

# Pruebas Beta

# Validación

---

- Las pruebas Beta ocurren cuando se desarrolla un incremento completo del software y después de todas las fases del desarrollo.
- Involucra solicitar a personas probar el software en su ambiente real.
- El objetivo es encontrar los errores más obvios y solucionarlos antes de la liberación final.

# Validación

---

- Sin importar el proceso de desarrollo de software y su excelente implementación, siempre existirán defectos que pasaron por alto a los ojos de los desarrolladores y el equipo de pruebas.
- Las pruebas exhaustivas son imposibles; los testers usan su habilidad y juicio para anticipar la manera en la que el software será utilizado

# Validación

---

- El testeo Beta es utilizado para cubrir las brechas entre el uso anticipado y el uso real.
- El testeo Beta debe ser realizado por usuarios finales y no por miembros del equipo de pruebas

(O'Docherty)

# Validación

---

- Las pruebas beta es el momento en el que el producto de software elaborado, finalmente llega a los usuarios “reales”, llamados “Beta testers”. Dependiendo de la naturaleza del software:
  - ✓ Colegas. (*Si testearas mi producto, yo testearé el tuyo*)
  - ✓ Clientes reales. (*Descuentos en la versión final liberada*)
  - ✓ Ambos
  
- Los “Beta testers” tienen las siguientes características:

# Validación

---

- Son lo más cerca que estaremos de los usuarios en el mundo real.
- En ocasiones pertenecen a compañías que conocen a fondo el mercado del producto.
- Tienen otras actividades además de “jugar” con el producto de software, por lo que es necesario analizar la razón por la cual realizarán esta tarea:
  - Porque actualmente usan el producto y desean saber:
    - Si vale la pena continuar usándolo.
    - Si las funcionalidades que han estado solicitando durante los últimos 18 meses están ahí.
  - Porque posiblemente lo recomienden para compra y tomar parte de la prueba Beta es la manera más barata de que obtengan una copia.

# Validación

---

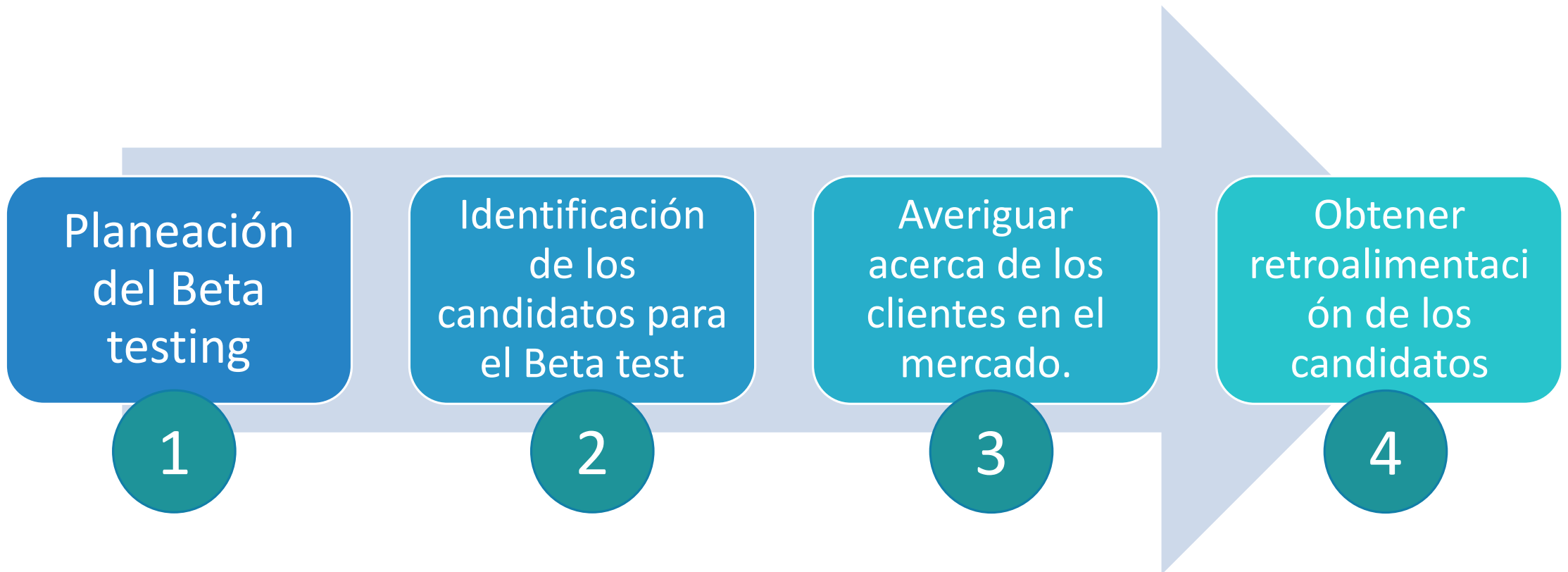
- No son testers. En su mayoría no tienen idea de cómo se reportan los defectos y estarían encantados de decirte que todo funciona de maravilla, aún cuando se darán por vencidos al ver la amplitud del manual.
- Tienen software extraño en sus computadoras que hace de la instalación un problema.
- Pueden encontrar defectos en los que nunca pensaste.



# Validación

---

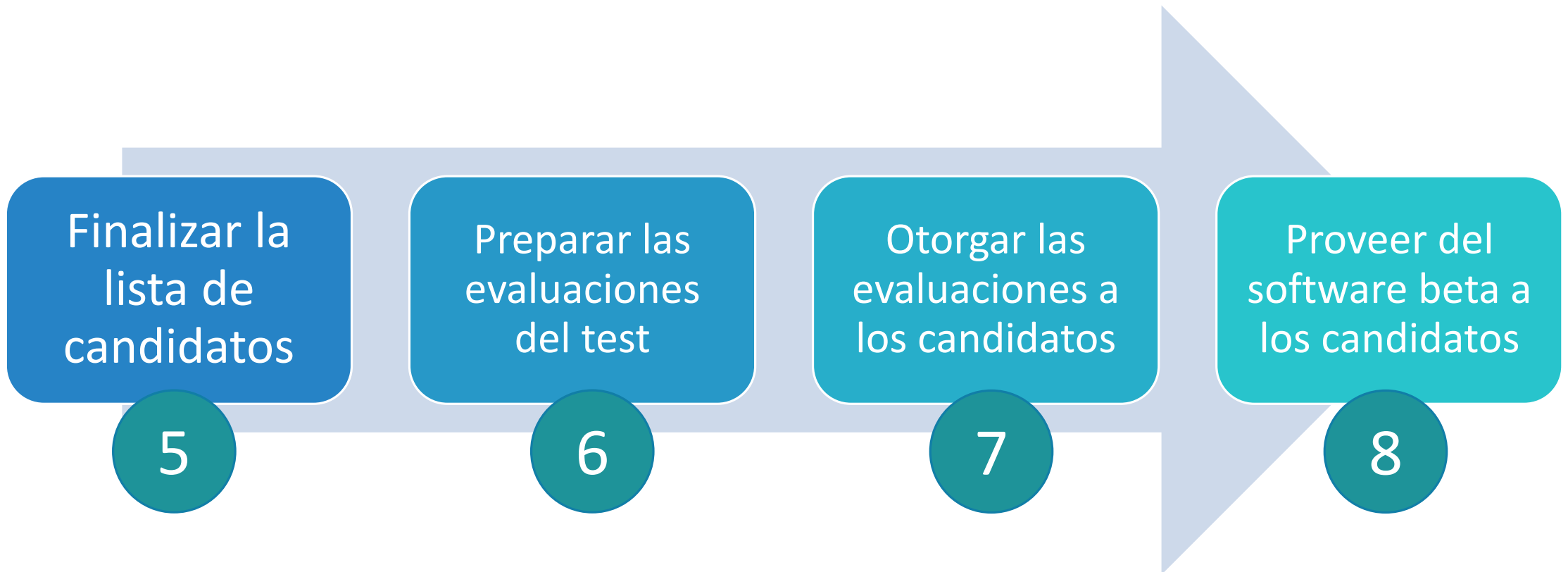
- El proceso de testeo beta contiene las siguientes etapas:



# Validación

---

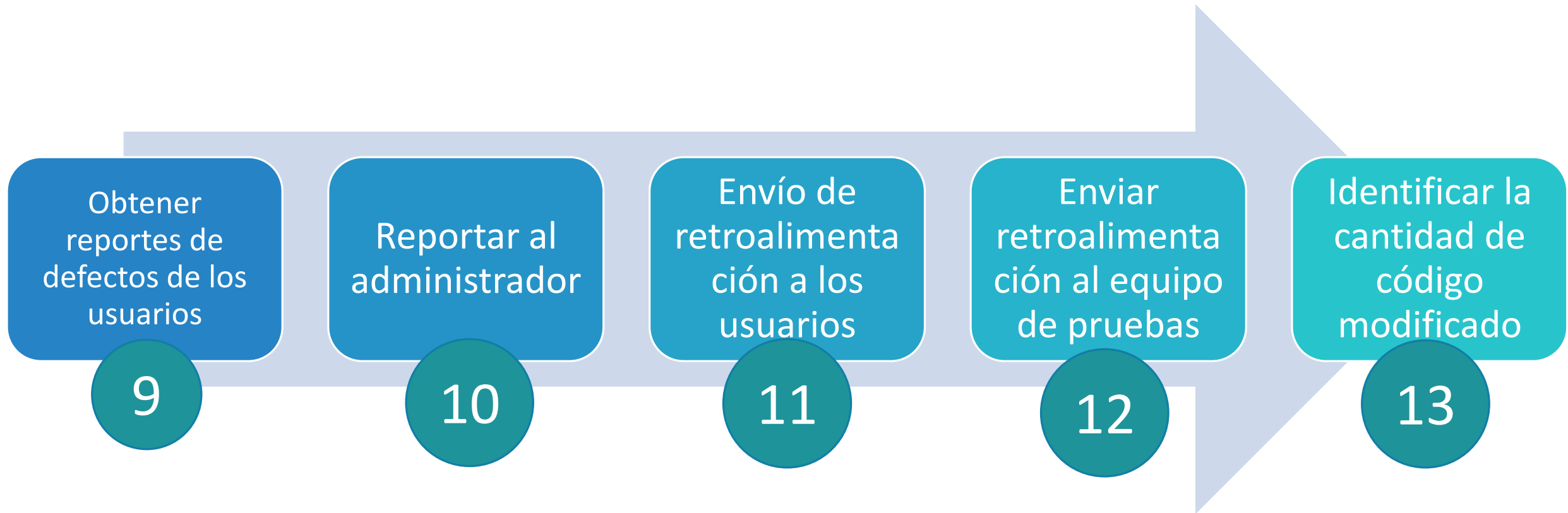
- El proceso de testeo beta contiene las siguientes etapas:



# Validación

---

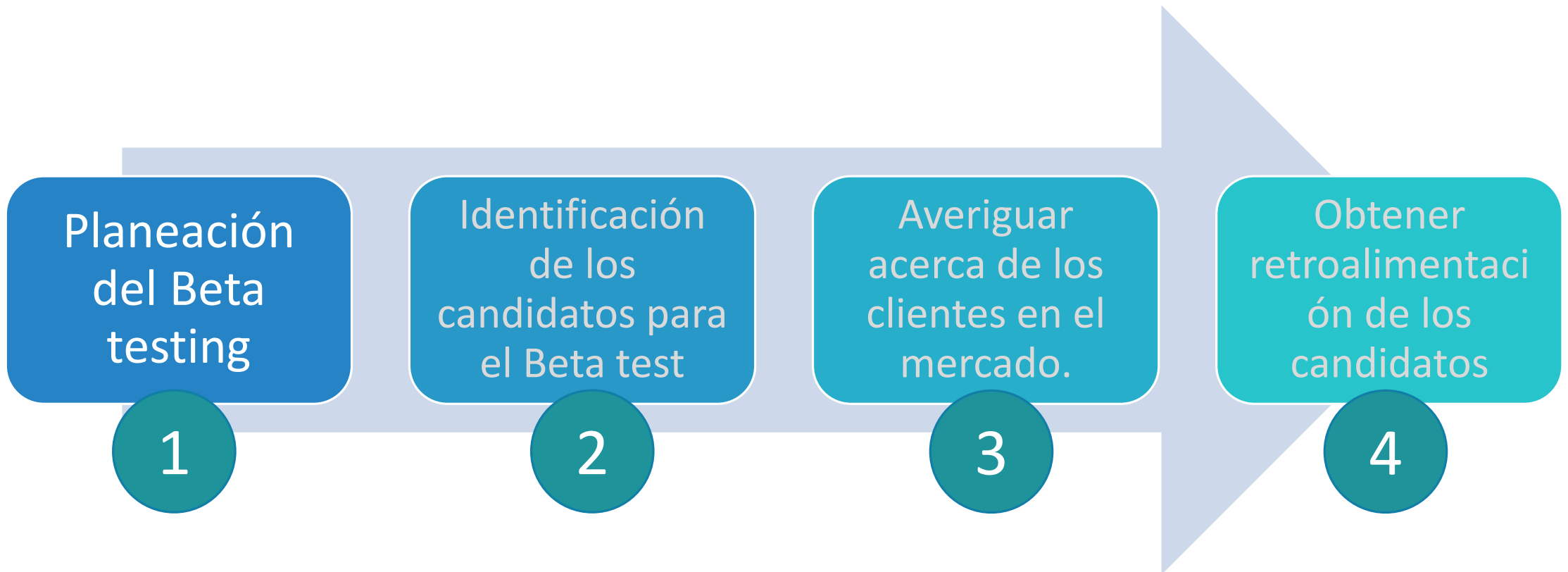
- El proceso de testeo beta contiene las siguientes etapas:



# Validación

---

- El proceso de testeo beta contiene las siguientes etapas:



1

# Validación

---

## Planificación del Testing.

- a) Quién está a cargo de la ejecución del Test.
- b) Qué datos se espera obtener.
- c) Qué información se le proporcionará a los beta testers.
- d) Cómo será la interfaz con los usuarios (specially-trained o staff de soporte existente).

# 1

# Validación

---

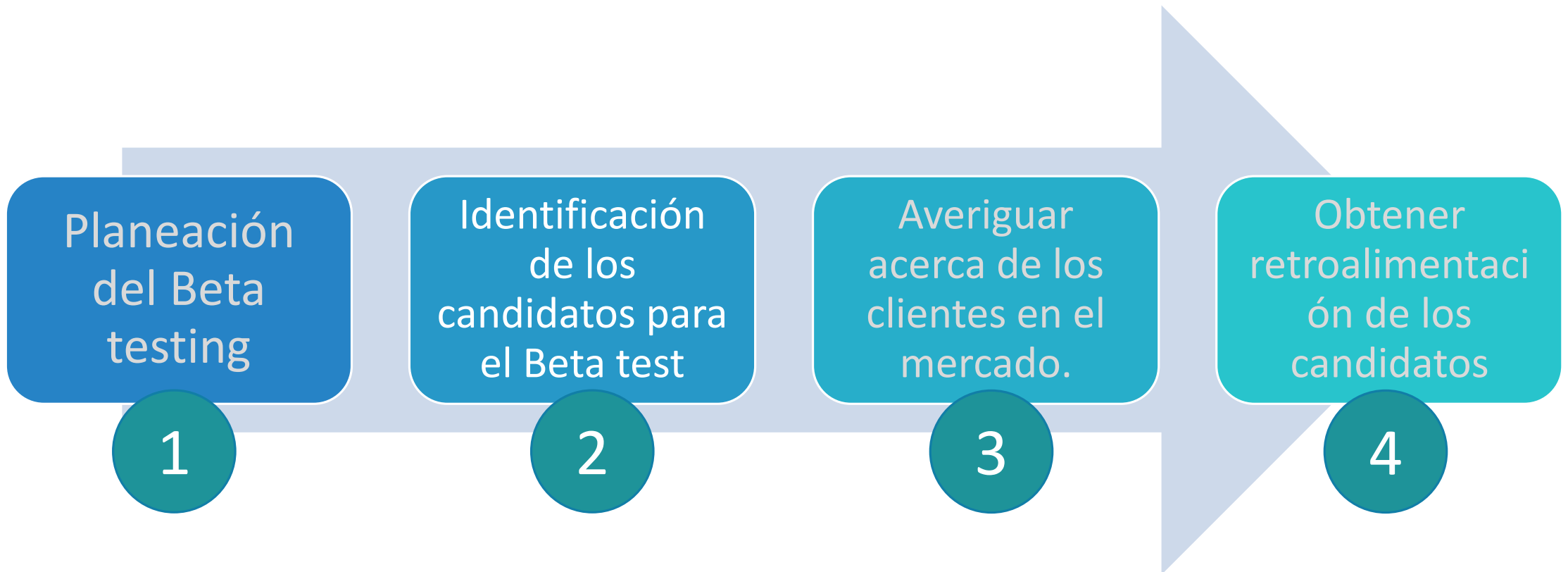
## Planificación del Testing.

- e) Cómo se manejarán los reportes de defectos.
  - Comunicación con el equipo de prueba
  - Comunicación con el equipo de desarrolladores
- f) Cómo se reportará el progreso a la administración.
- g) Cómo se recompensará a los beta testers
- h) Identificar el conjunto de características mínimas del test (p. e. Si el tiempo es corto es preferible tener vistas de usuario en una liberación temprana y parcial pero usable con tiempo de correcciones a una liberación completa pero sin tiempo para mejoras)

# Validación

---

- El proceso de testeo beta contiene las siguientes etapas:



2

## Validación

---

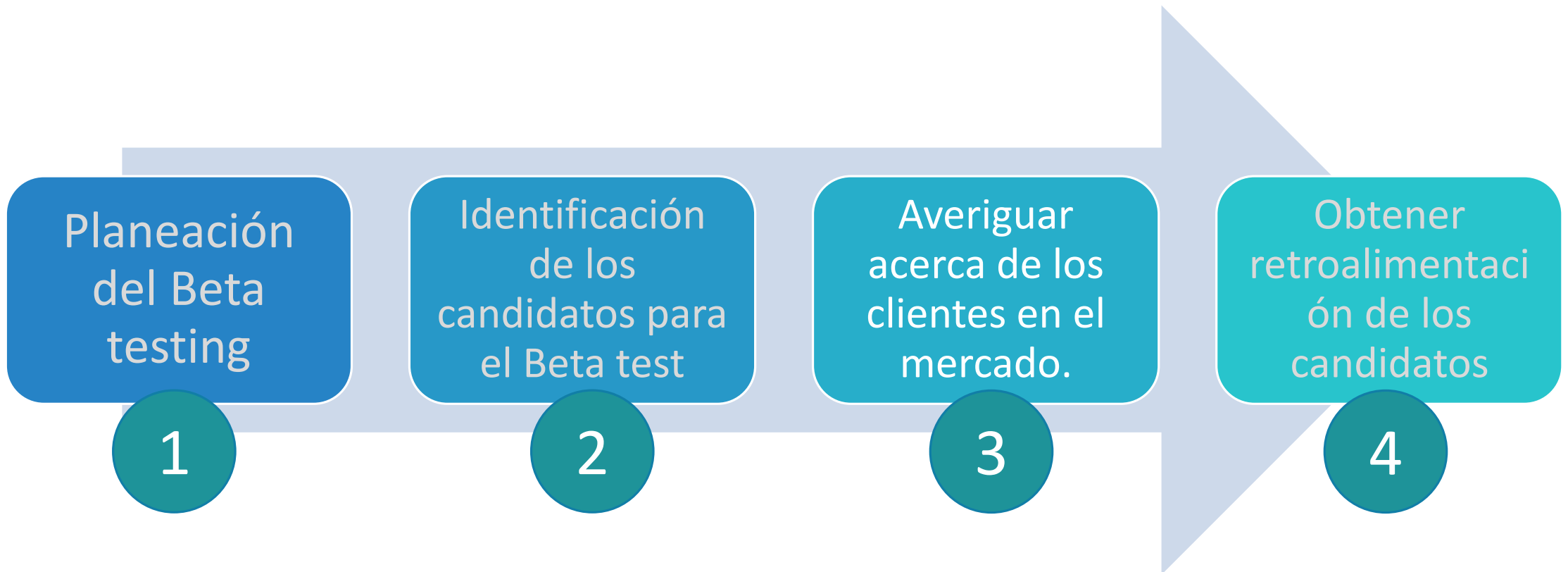
Identificar los candidatos para el Test Beta.



# Validación

---

- El proceso de testeo beta contiene las siguientes etapas:



# Validación

---

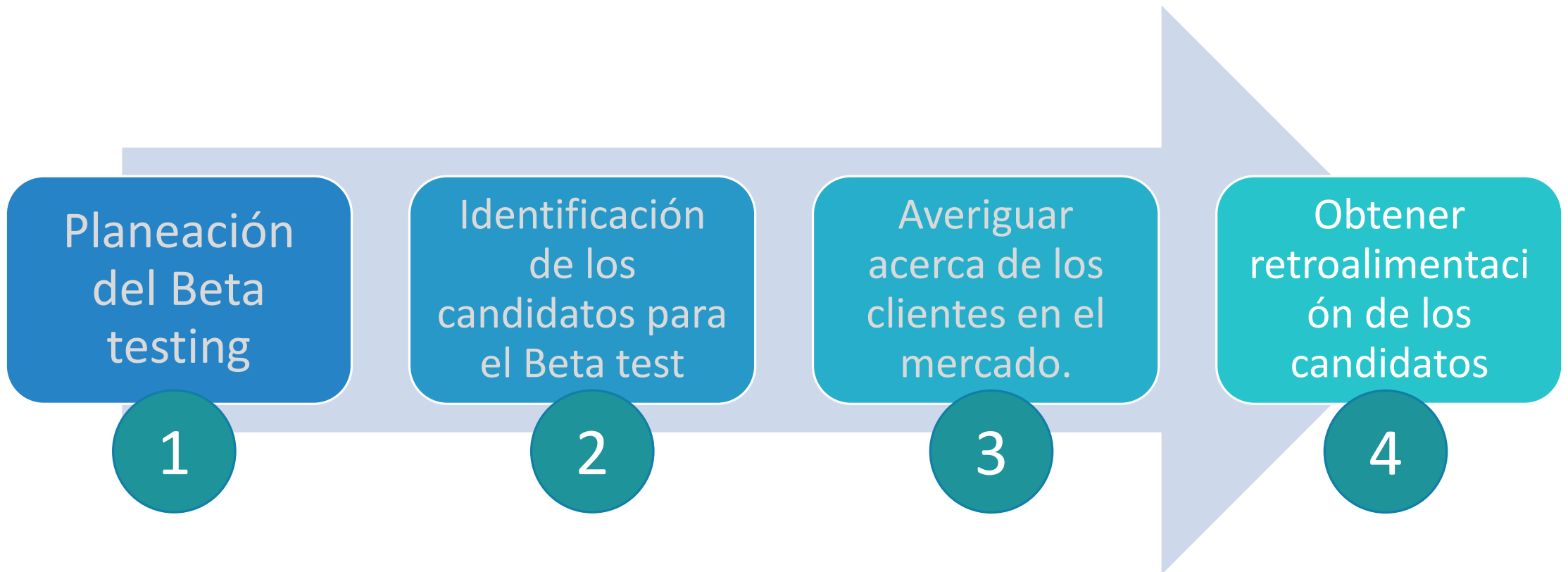
## Averiguar quienes son los clientes en el mercado

- a) Lo más típico en el sector del mercado.
- b) Las quejas públicas menos probables si el software falla.
- c) Los más acostumbrados a beneficiarse de las pruebas beta.
- d) Los más probables de tomar pruebas beta.
- e) Los más probables de enviar reportes de defectos utilizables.

# Validación

---

- El proceso de testeo beta contiene las siguientes etapas:



# 4

## Validación

---

### Obtener retroalimentación de los usuarios

- a) Llamar a los candidatos por teléfono.
- b) Identificar quienes son, qué hacen, cuánto tiempo esperan dedicar al test.
- c) Obtener otros medios de localización.
- d) Confirmar como, si aplica, el test es de valor para el candidato.
- e) Identificar si el candidato es una persona confiable para realizar cualquier tipo de test.

# Validación

---

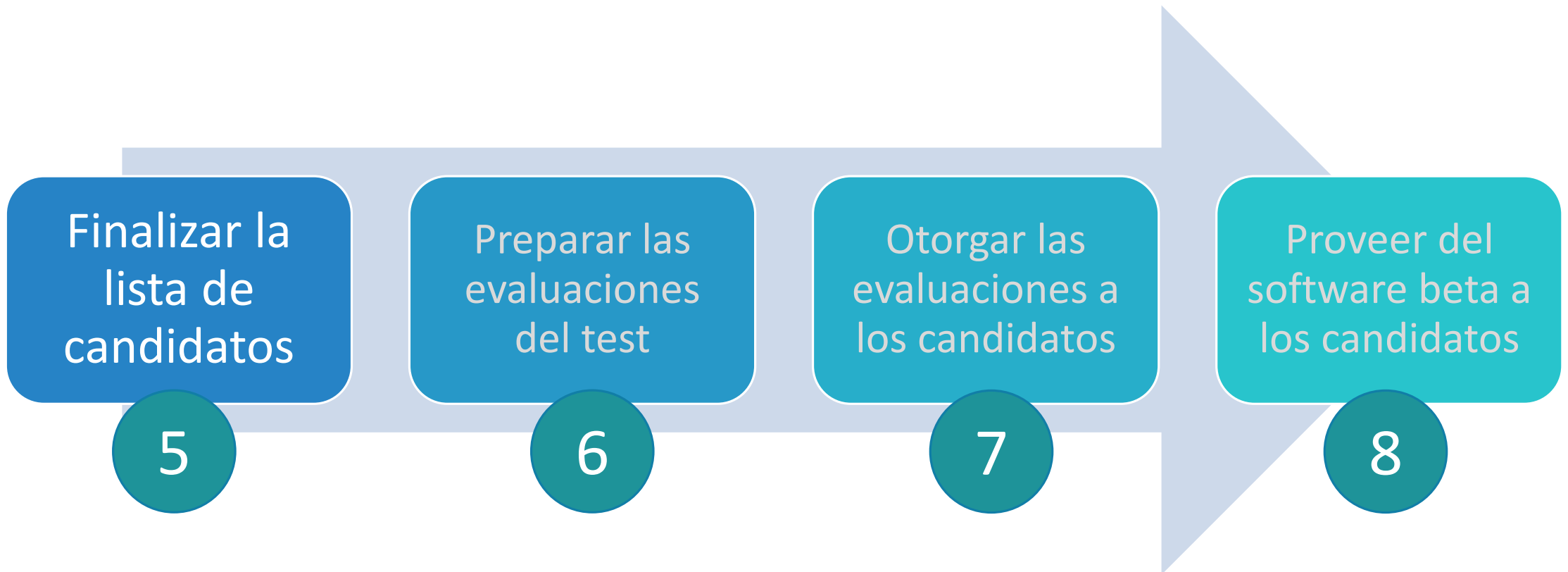
## Obtener retroalimentación de los usuarios

- f) Identificar las funcionalidades que usará con mayor probabilidad.
- g) Obtener cualquier tipo de documentación de usuario que muestre la forma en la que usarán el producto.
- h) Asegurar que el candidato no dañará la compañía usando el sistema pensando que preferiría “jugar” en un ambiente completo y aislado.
- i) Alentar al beta tester a enfocarse en aspectos y defectos de usabilidad. Realzar la importancia con frases como *“Creemos que funciona, solo tu nos puedes decir si funciona bien”*

# Validación

---

- El proceso de testeo beta contiene las siguientes etapas:



# 5

## Validación

---

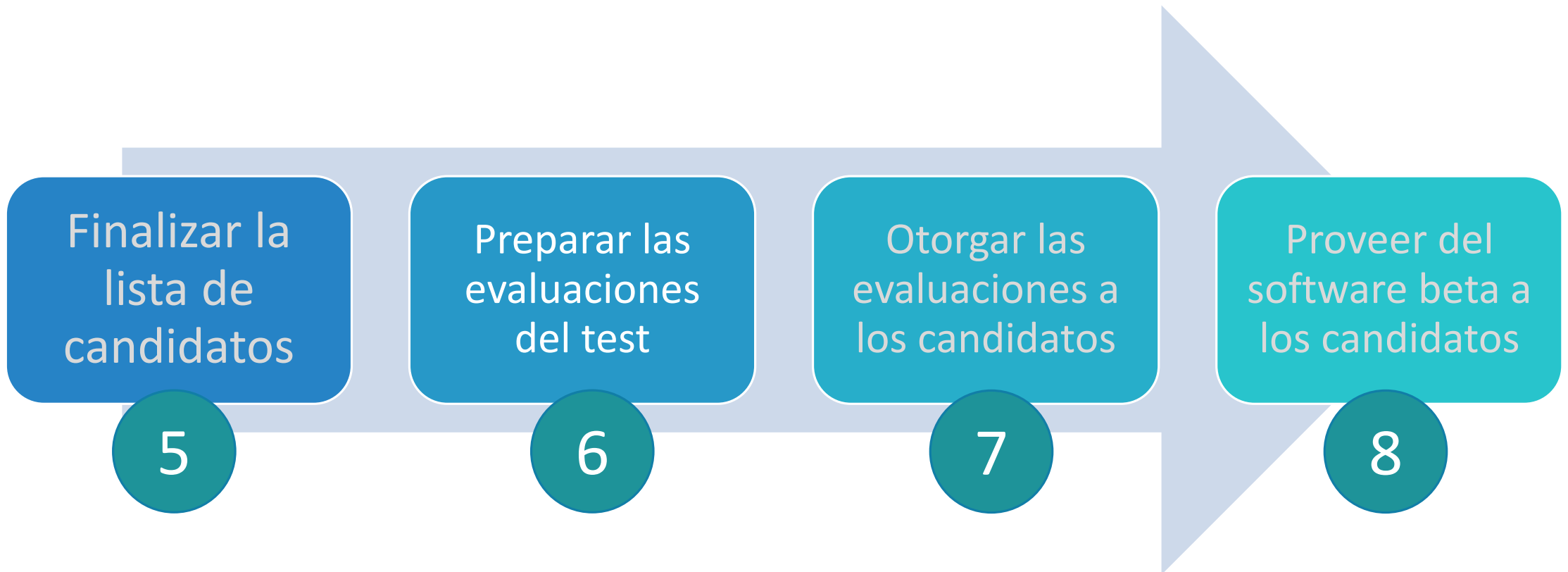
### Finalizar la lista de candidatos al test beta.

- 1) Asegurar que se cuenta con el número suficiente de candidatos en caso de que alguno deserte.
- 2) Asegurar que el sector del mercado y la cobertura de características es suficiente.
- 3) Hablar con los candidatos para confirmar que han sido elegidos.

# Validación

---

- El proceso de testeo beta contiene las siguientes etapas:





## 6

# Validación

---

## Preparar las evaluaciones para el test Beta.

### Considere lo siguiente:

- 1) No espere que los usuarios beta tengan la habilidad en preparaciones de test.
- 2) Será la primera vez que usarán una nueva versión del software.
- 3) Apoye a los usuarios Beta con listas de verificación. Por ejemplo:

1. Instale	6. Inicie
2. Desinstale	7. Use las características d, e, f,..
3. Instale	8. Cierre
4. Use las funcionalidades a, b c...	9. Instale
5. Cierre	10. Use las características A, C, E, G...

6

# Validación

---

Preparar las evaluaciones para el test Beta.

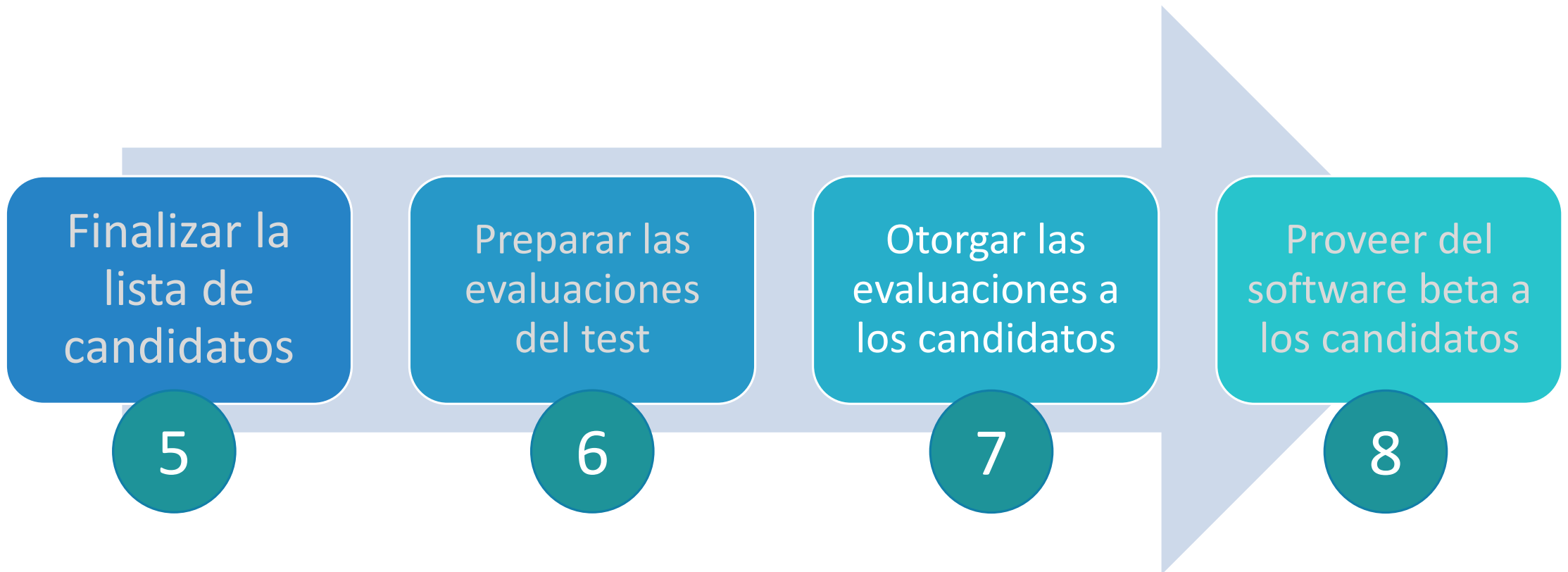
Considere lo siguiente:

4) Elabore una hoja de reporte de fallas.

# Validación

---

- El proceso de testeo beta contiene las siguientes etapas:



# 7

## Validación

---

Otorgar las evaluaciones a los candidatos

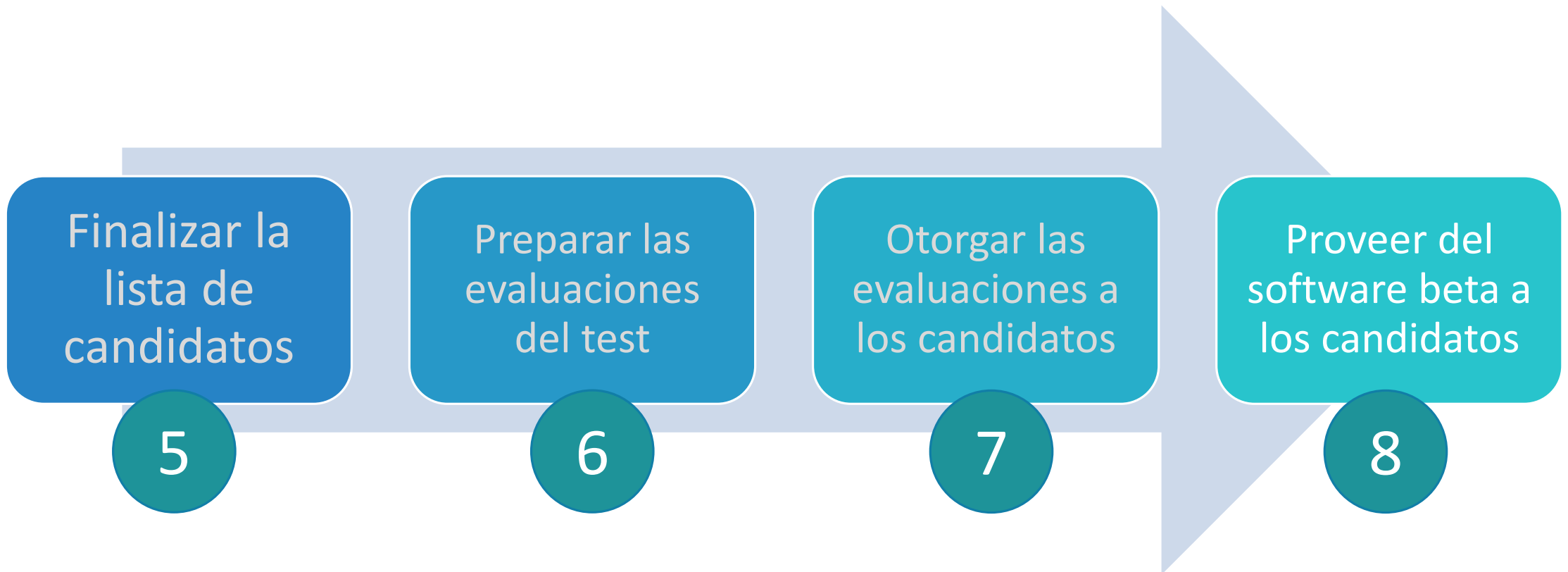
Considere:

- 1) Mantener contacto continuo con los usuarios beta, de esta forma sabrá que le interesan.

# Validación

---

- El proceso de testeo beta contiene las siguientes etapas:



# Validación

---

## Proveer del software beta a los usuarios

### Considere lo siguiente:

- i. Contacte a los usuarios beta dos días antes de la liberación para advertirles.
- ii. Asegure que cuentan con el ambiente de prueba disponible.
- iii. Asegure que cuentan con la documentación necesaria.
- iv. Establezca comunicación nuevamente para saber si instalaron correctamente el software.

# Validación

---

- El proceso de testeo beta contiene las siguientes etapas:



# Validación

---

## Obtener reportes de defectos de los usuarios

### Considere lo siguiente:

Agradezca personalmente de forma breve.

Comparar los defectos con aquellos ya encontrados:

- i. Alguna funcionalidad es más defectuosa de lo que se pensaba?
- ii. Alguna característica no se cubrió en su totalidad?



# Validación

---

- El proceso de testeo beta contiene las siguientes etapas:



# Validación

---

## Reportar al administrador

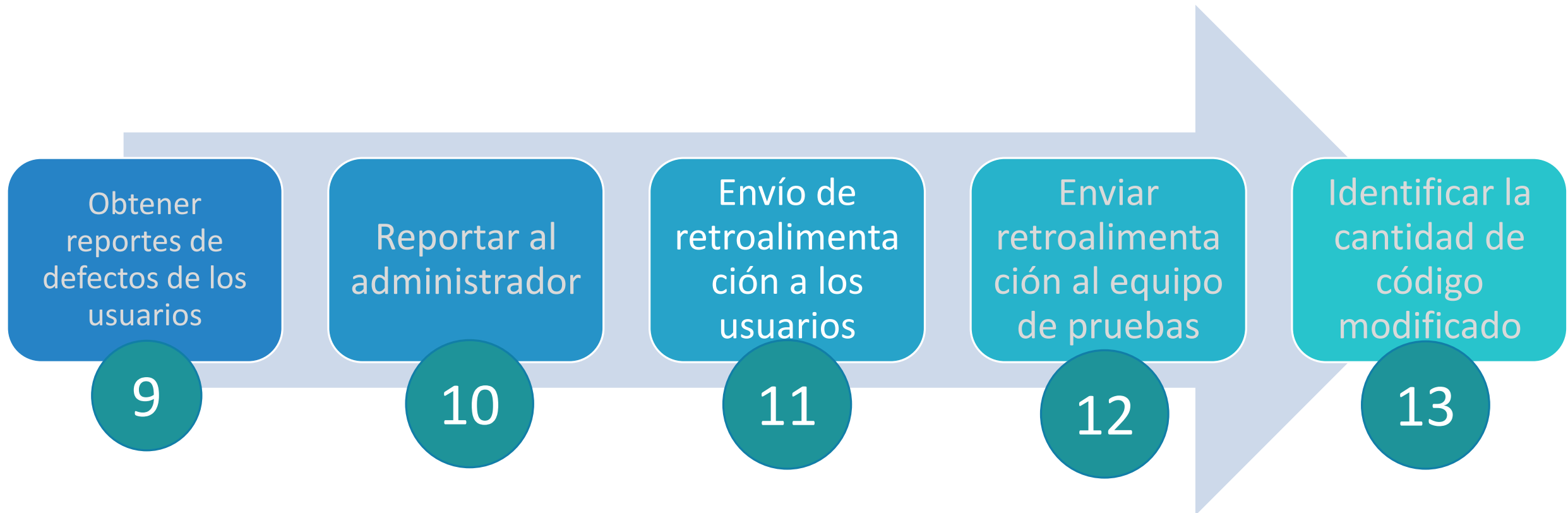
### Responda lo siguiente:

- i. En qué medida el programa de prueba beta se mostró en el programa de prueba alfa.
- ii. Hubo problemas específicos de hardware?
- iii. Hubo problemas de rendimiento?
- iv. Cuáles son las implicaciones de la prueba Alfa y la fecha de liberación final?
- v. El número de defectos y su severidad son peor de lo que se esperaba?

# Validación

---

- El proceso de testeo beta contiene las siguientes etapas:



# Validación

---

## Envío de retroalimentación a los usuarios

### Considere lo siguiente:

- i. Reporte a los usuarios beta el progreso regularmente.
- ii. Si se liberará una nueva versión, hágales saber que no perdieron el tiempo testeando la versión anterior.

# Validación

---

- El proceso de testeo beta contiene las siguientes etapas:



# Validación

---

## Enviar retroalimentación al equipo de pruebas

### Considere lo siguiente:

- i. Enviar el conjunto de pruebas a revisar.
- ii. Cuáles ambientes de prueba requieren cambios.
- iii. Los regímenes de rendimiento que requieren cambio.

# Validación

---

- El proceso de testeo beta contiene las siguientes etapas:



# Validación

---

## Identificar la cantidad de código modificado

- i. Compare la cantidad de código modificado como resultado de las pruebas Alfa con el código modificado en las pruebas Beta.



# Pruebas de Aceptación

# Validación

---

- Las pruebas de aceptación como un subconjunto de las pruebas de sistema, están diseñadas para demostrar al cliente que el sistema cumple con sus requerimientos.
- Ocurren cuando los defectos encontrados en las pruebas de sistema han sido corregidos, clarificados o aceptados por el cliente.
- Finaliza cuando el certificado de prueba de aceptación es firmado formalmente por cliente.

(Farrell-Vinay)

# Validación

---

- Las pruebas de Aceptación son un testeo formal conducido para determinar si el sistema satisface su *criterio de aceptación* y para posibilitar al comprador a determinar si acepta el sistema.
- Las pruebas de aceptación a la entrega, son usualmente la última oportunidad del comprador para examinar el software y en su caso, redirigirlo al desarrollador por ser insuficiente o incorrecto.

# Validación

---

- Frecuentemente las pruebas de aceptación son la única oportunidad que el cliente tiene para identificar las deficiencias del software.
- Lo que trae un importante riesgo para el cliente, con la probabilidad de que el sistema no opere de forma confiable (por la falta de un control de calidad adecuado durante el proceso de desarrollo).

# Validación

---

- Para mitigar este riesgo, el cliente debe involucrarse con la aceptación del software de manera temprana a la adquisición del software.
- Como un proceso de ciclo de vida, la aceptación del software permite:
  - a) Detección temprana de problemas.
  - b) Preparación de instalaciones apropiadas para el test.
  - c) Consideración temprana de las necesidades del usuario durante el desarrollo del software.

# Validación

---

¿Qué significa la aceptación del software?

# Validación

---

- La aceptación del software es un proceso incremental de aprobación o rechazo de los sistemas de software durante el desarrollo o mantenimiento; de acuerdo a qué tanto el software satisface el criterio predefinido.

# Validación

---

- Las decisiones de aceptación suceden en tiempos predefinidos, cuando los procesos, herramientas de soporte, documentación provisional, segmentos del software y el sistema de software en su totalidad cumplen los criterios predefinidos de aceptación.
- La decisión final:



# Validación

---

- ✓ La documentación entregada es adecuada y consistente con el sistema en ejecución.
- ✓ Y el sistema de software completo cumple con los requerimientos del cliente.

**Esta decisión es usualmente basada en la Prueba de Aceptación del Software**

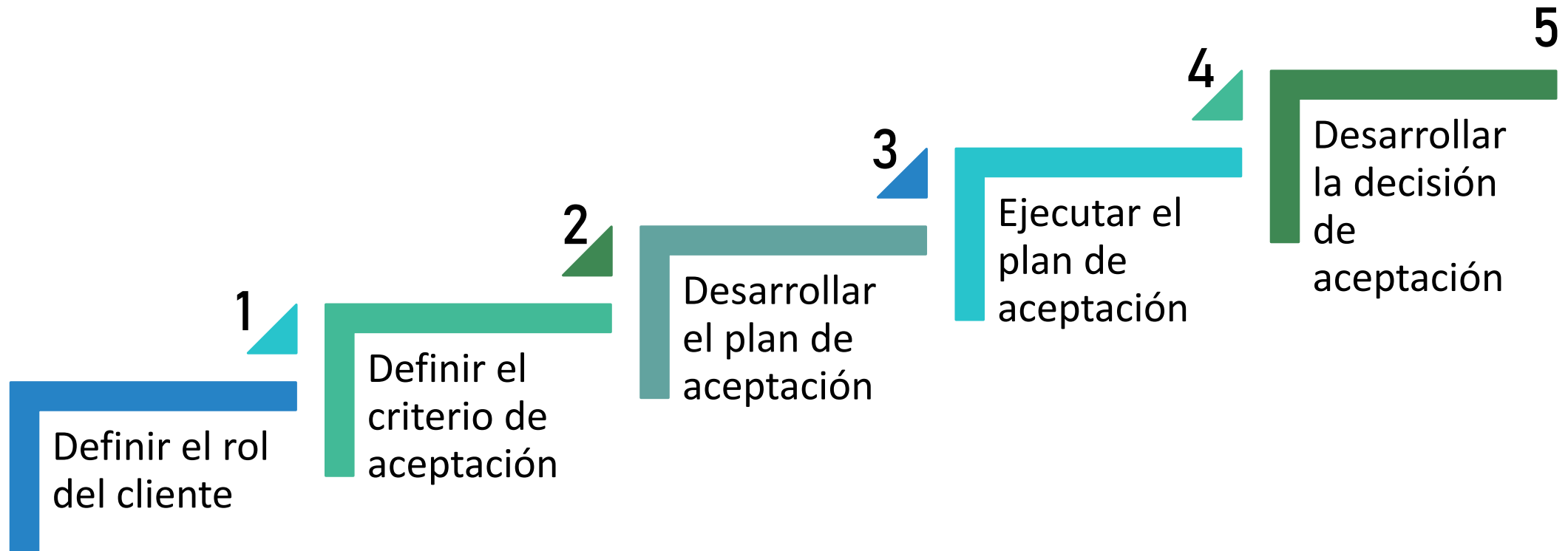
# Validación

---

- La prueba de aceptación de software final y formal debe ocurrir al final del proceso de desarrollo.
- Consiste en probar para determinar si el sistema desarrollado cumple con la **funcionalidad** predeterminada, **rendimiento**, **calidad** y criterios de la **interfaz**.

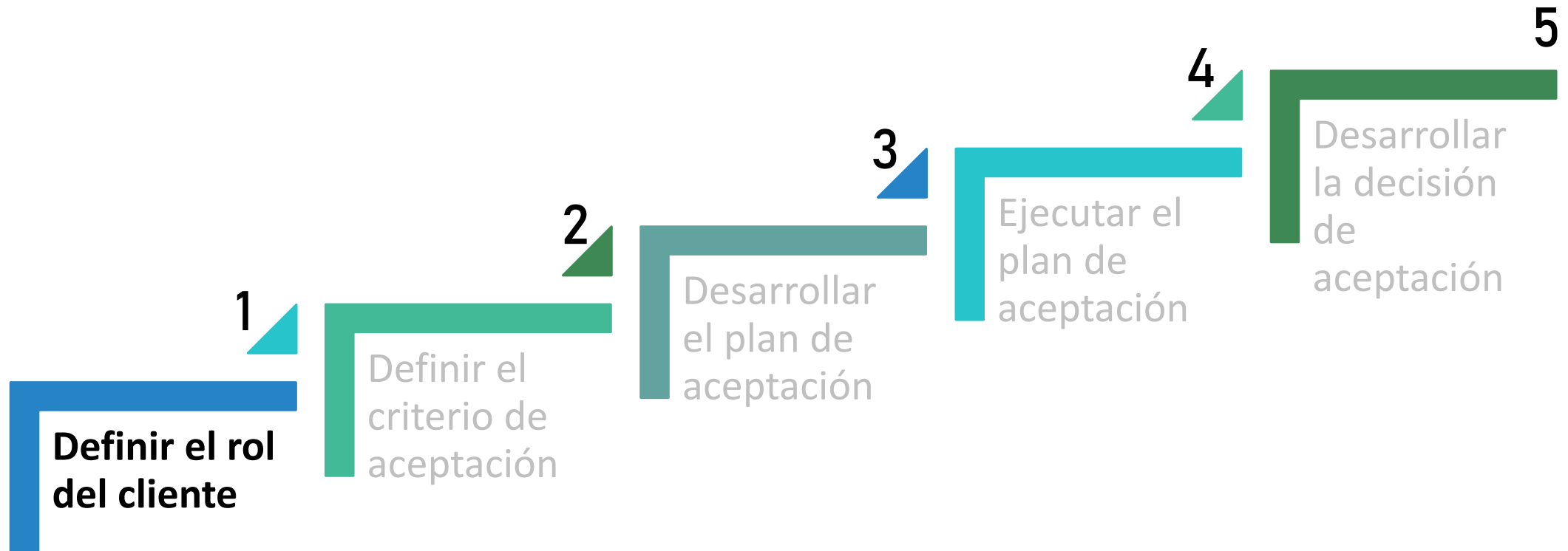
# Validación

## ■ Proceso de aceptación del software



# Validación

## ■ Proceso de aceptación del software



# Validación

## 1. DEFINIR EL ROL DEL CLIENTE

- La responsabilidad de la aceptación del software es del cliente o comprador y sus responsabilidades son:
  - a) Asegurar que el usuario se encuentra envuelto en el desarrollo de los requerimientos del sistema y el criterio de aceptación.
  - b) Identificar los productos provisionales y finales para la aceptación, su criterio de aceptación y calendario.
  - c) Planear cómo y quien realizará cada actividad de aceptación.
  - d) Proveer la información en la cual basar las decisiones de aceptación.

# Validación

## 1. DEFINIR EL ROL DEL CLIENTE

- La responsabilidad de la aceptación del software es del cliente o comprador y sus responsabilidades son:
  - e) Calendarizar de forma adecuada la recepción y examinación de productos y evaluaciones antes de la revisión de aceptación, por parte del staff del cliente/comprador.
  - f) Preparar el plan de aceptación.
  - g) Responder a los análisis de las entidades de proyecto antes de aceptar o rechazar.
  - h) Aprobar varios productos de software provisionales de acuerdo al criterio establecido.

# Validación

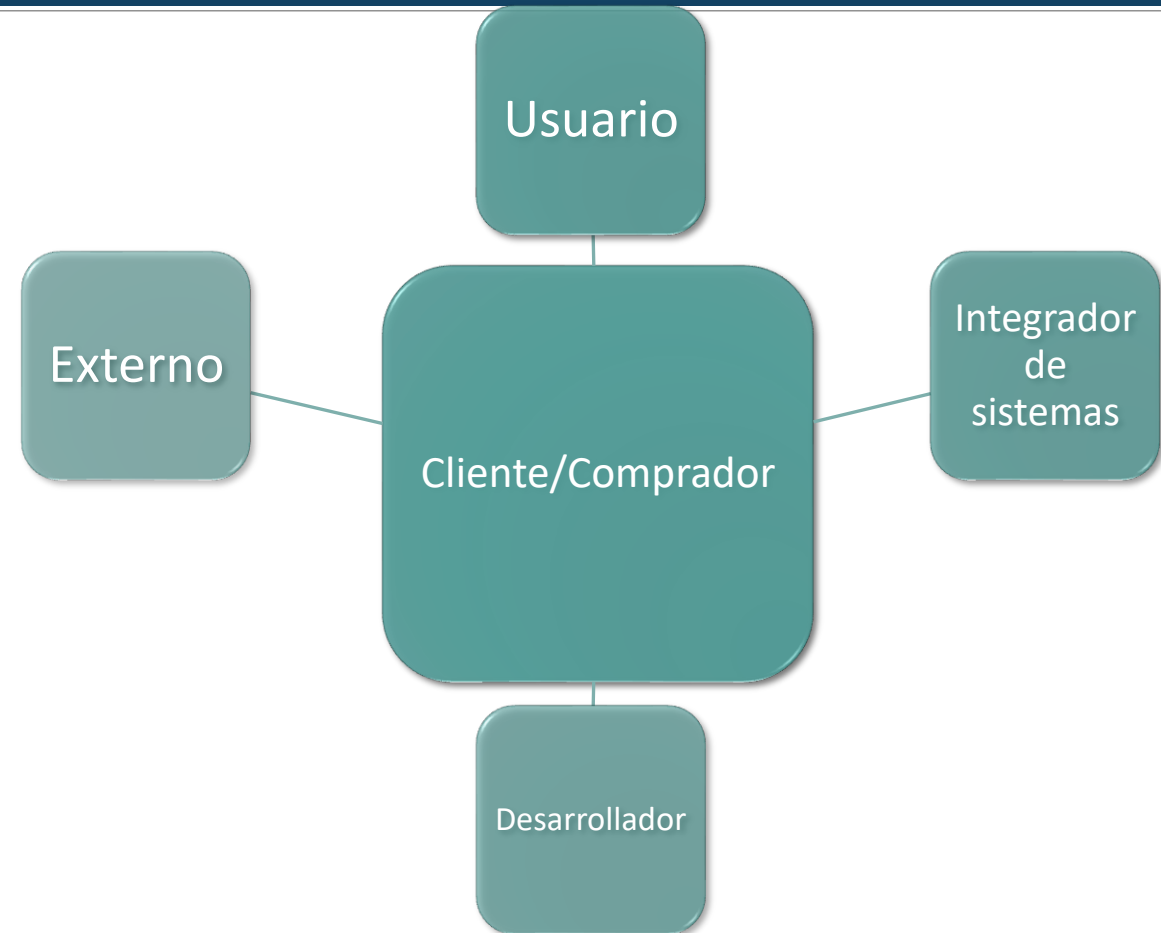
## 1. DEFINIR EL ROL DEL CLIENTE

- La responsabilidad de la aceptación del software es del cliente o comprador y sus responsabilidades son:
  - i) Elaborar las actividades de aceptación finales, incluyendo el testeo formal final a la entrega.
  - j) Tomar una decisión de aceptación para cada producto.

# Validación

## 1. DEFINIR EL ROL DEL CLIENTE

- El cliente/comprador puede delegar responsabilidades al administrador de aceptación:





# Validación

## 1. DEFINIR EL ROL DEL CLIENTE

- Por lo tanto el cliente tiene participación activa en:
  - a. La definición de la información requerida
  - b. Evaluar dicha información
  - c. Definir en diferentes puntos del proceso de las actividades de desarrollo, si el producto está listo para continuar a la siguiente actividad.
  - d. Identificar los productos de software y sus criterios de aceptación.
  - e. Examinar los requerimientos del sistema asegurando que representan las necesidades del usuario.
  - f. Asegurar que se completan todas las actividades de planeación y aceptación.
  - g. Realizar las pruebas de aceptación u observar los tests. (cliente con rol técnico)

# Validación

## 1. DEFINIR EL ROL DEL CLIENTE

- El criterio de aceptación es derivado de estos requerimientos que cuantitativamente describen las medidas funcionales, de interfaz, de rendimiento y calidad, que el software debe satisfacer.

# Validación

## 1. DEFINIR EL ROL DEL CLIENTE

- El **administrador de la aceptación** es el responsable de asegurar que los procedimientos de los tests se encuentran definidos e implementados completamente y que todos los recursos se encuentran listos para la prueba de aceptación.

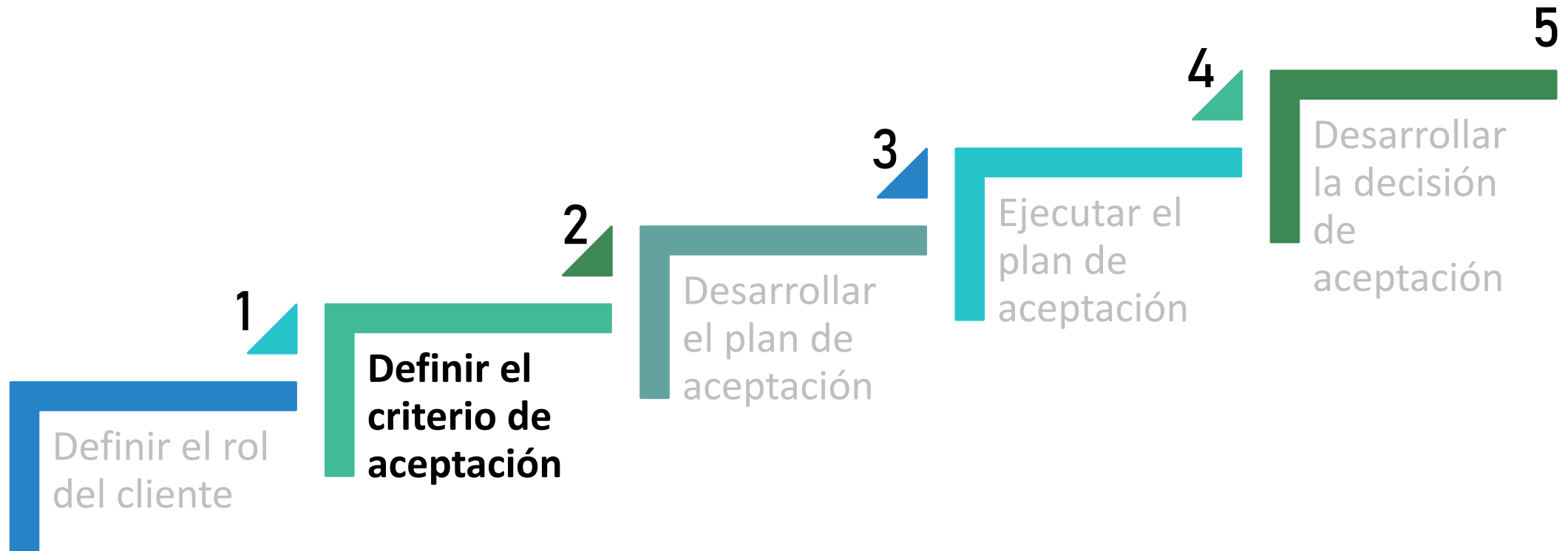
# Validación

## 1. DEFINIR EL ROL DEL CLIENTE

- En el caso de la prueba de aceptación final, las pruebas deben ser realizadas en el ambiente operacional.

# Validación

## ■ Proceso de aceptación del software



# Validación

## 2. DEFINIR EL CRITERIO DE ACEPTACIÓN

- El cliente/comprador debe definir el criterio bajo el cual el software es aceptable.
  - De manera ideal, este debería incluirse en la especificación de requerimientos del software.
- Para la preparación del desarrollo del criterio de aceptación, el cliente/comprador debería:

# Validación

## 2. DEFINIR EL CRITERIO DE ACEPTACIÓN

- a) Conocer profundamente la aplicación para la cual el sistema servirá.  
(Conocer el proceso de negocio)
- b) Familiarizarse por completo con la forma en la que actualmente se lleva a cabo el proceso para el cual servirá el sistema.
- c) Identificar aplicaciones similares en otras organizaciones.
- d) Entender los riesgos y beneficios de la metodología de desarrollo de software empleada.
- e) Entender por completo las consecuencias de agregar nuevas funcionalidades para mejorar un sistema existente.

# Validación

## 2. DEFINIR EL CRITERIO DE ACEPTACIÓN

- Los requerimientos de aceptación que un sistema debe cumplir se dividen en seis categorías:
  1. Funcionalidad
  2. Rendimiento
  3. Calidad de la interfaz
  4. Calidad general del software
  5. Seguridad
  6. Seguridad del software



# Validación

## 2. DEFINIR EL CRITERIO DE ACEPTACIÓN

- Los **requerimientos de funcionalidad** se relacionan con las reglas de negocio que el sistema debe ejecutar.
- Los **requerimientos de rendimiento**, están relacionados con los requerimientos operacionales, como tiempo o restricciones de recursos.
- Los **requerimientos de calidad** de la interfaz se relacionan con la conexión entre componentes de procesamiento (humano-computadora, módulo-módulo)

# Validación

## 2. DEFINIR EL CRITERIO DE ACEPTACIÓN

- Los **requerimientos de calidad general del sistema** son aquellos que especifican límites para factores o atributos como confiabilidad, testeabilidad, exactitud y usabilidad.

Por ejemplo el criterio:

**“El documento de requerimientos debe tener menos de 5 enunciados que describan información faltante”**

Es un ejemplo factor de calidad de completitud

# Validación

## 2. DEFINIR EL CRITERIO DE ACEPTACIÓN

- Los **requerimientos de seguridad** están relacionados a accesos autorizados a los recursos del sistema y a procesar la integridad.
- Los **requerimientos de seguridad** de software son necesarios cuando puede ocurrir una lesión o la muerte como resultado de la falla del sistema.
- El nivel de criticidad de un sistema (p.e. Las fallas pueden resultar en pérdida de vidas humanas) es importante para determinar el criterio de aceptación cuantitativa.

# Validación

## 2. DEFINIR EL CRITERIO DE ACEPTACIÓN

- El usuario debe determinar el grado de criticidad de los requerimientos en las seis categorías.
- Algunos factores típicos que afectan la criticidad son:
  - ✓ Importancia del sistema dentro de la organización o la industria.
  - ✓ Consecuencia de los fallos.
  - ✓ Complejidad del proyecto.
  - ✓ Riesgo tecnológico
  - ✓ Complejidad del ambiente del usuario.

# Validación

## 2. DEFINIR EL CRITERIO DE ACEPTACIÓN

- Productos o partes de productos con requerimientos críticos no califican para *Aceptación* si no satisfacen el criterio.
- Un producto con fallas en un requerimiento no crítico, puede ser Aceptado, dependiendo del criterio de aceptación cuantitativo para los factores de calidad; sin embargo si falla en varios criterios no críticos, su calidad es cuestionable.

# Validación

## 2. DEFINIR EL CRITERIO DE ACEPTACIÓN

- El cliente tiene la responsabilidad de asegurar que los requerimientos funcionales, de rendimiento, de seguridad de software, de seguridad y de requerimientos, contengan criterios numéricos.
- Es importante considerar que un sistema con valores absolutos en lugar de un rango de valores en los criterios de aceptación, puede resultar muy caro o nunca satisfacer dichos criterios.

# Validación

## 2. DEFINIR EL CRITERIO DE ACEPTACIÓN

- La siguiente tabla muestra algunos problemas o preocupaciones de aceptación para las seis categorías genéricas.

# Validación

## 2. DEFINIR EL CRITERIO DE ACEPTACIÓN

### Funcionalidad

La documentación y el código mantienen una consistencia interna y entre fases; trazabilidad de funcionalidad; verificación adecuada de lógica; evaluación funcional y testing; preservación de funcionalidad en el ambiente operacional.

### Rendimiento

Análisis de factibilidad de requerimientos de rendimiento; herramientas correctas de simulación e instrumentación; análisis de rendimiento en el ambiente de operación

Aspectos de aceptación por categoría



# Validación

## 2. DEFINIR EL CRITERIO DE ACEPTACIÓN

### Calidad de la interfaz

Documentación de la interfaz; complejidad de la interfaz; planes de testeo de integración y de interfaz; ergonomía de la interfaz; pruebas de interfaz en el ambiente operacional.

### Calidad general del sistema

Cuantificación de las medidas de calidad; Criterios de aceptación para todos los productos de software; Adecuación de la documentación y del desarrollo del sistema de software a estándares; criterio de calidad para testing operacional.

Aspectos de aceptación por categoría

# Validación

## 2. DEFINIR EL CRITERIO DE ACEPTACIÓN

### Seguridad

Identificación de requerimientos de seguridad; planes de testeo de seguridad; verificación formal de la seguridad.

### Seguridad del software

Identificación de los requerimientos de seguridad; construcción y trazo del árbol de fallas; eliminación de métodos o tecnología de desarrollo inapropiadas para los sistemas de seguridad crítica; incorporación de código fallo-seguridad para prevención y recuperación de estados inseguros potenciales.

Aspectos de aceptación por categoría

# Validación

## 2. DEFINIR EL CRITERIO DE ACEPTACIÓN

- Después de identificar los problemas y preocupaciones , el cliente/comprador debe establecer el criterio de aceptación, tanto para elementos individuales como para el sistema final.
- Este criterio deben ser los valores numéricos aceptables o un rango de valores.
- El cliente debe comparar los valores de aceptación establecidos contra el número de problemas presentados en la fase de aceptación.

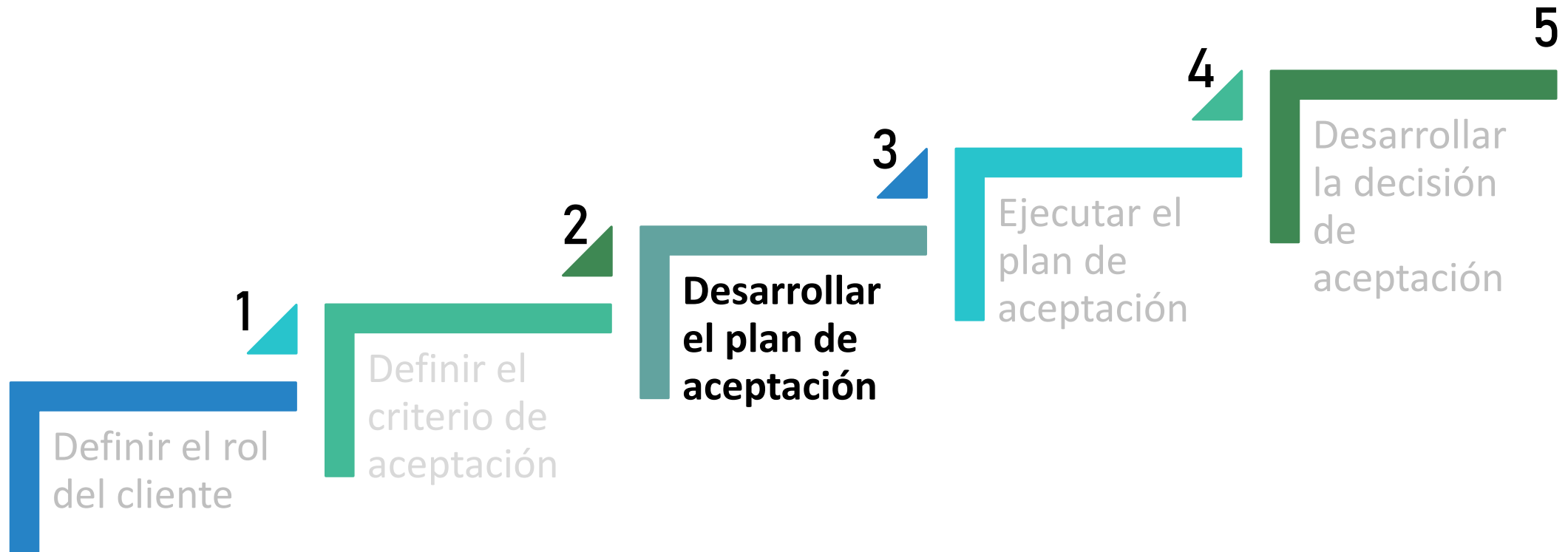
# Validación

## 2. DEFINIR EL CRITERIO DE ACEPTACIÓN

- Por ejemplo, si el número de requerimientos inconsistentes excede el criterio de aceptación, el documento de requerimientos debe ser rechazado.
- En este momento, los procedimientos establecidos para iteraciones y control de cambios entran en acción.

# Validación

## ■ Proceso de aceptación del software



# Validación

## 3. DESARROLLAR EL PLAN DE ACEPTACIÓN

- El primer paso para una aceptación de software efectiva es el desarrollo simultáneo de un **plan de aceptación** de software, **planes del proyecto en general** y **requerimientos contractuales**; para asegurar que las necesidades del usuario están representadas de forma correcta y completa en los requerimientos contractuales.

# Validación

## 3. DESARROLLAR EL PLAN DE ACEPTACIÓN

### ■ Plan:

1. Definición de objetivos de las actividades de aceptación. Realizado por el administrador de la aceptación y los usuarios.
  - Las bases de los objetivos son los **requerimientos contractuales**; cómo se espera que el software sea usado en el ambiente operacional. Asimismo, los riesgos asociados con el ciclo de vida del proyecto.

# Validación

## 3. DESARROLLAR EL PLAN DE ACEPTACIÓN

### ■ Plan:

1. En base a los objetivos, el plan inicial de aceptación de software es preparado, revisado y aprobado. En la siguiente tabla se muestran las secciones que un plan de aceptación puede contener.
2. El administrador de la aceptación es el responsable de implementarlo y asegurar que los objetivos de la aceptación se cumplan.



# Validación

## 3. DESARROLLAR EL PLAN DE ACEPTACIÓN

### Descripción del proyecto

Tipo de sistema; metodología de ciclo de vida; comunidad de usuarios para la liberación del sistema; tareas mayores que el sistema debe satisfacer; interfaces externas del sistema; uso esperado del sistema; mal uso potencial; riesgos; restricciones; estándares y prácticas.

### Responsabilidades de los usuarios

Organización y responsabilidades para las actividades de aceptación; requerimientos de recursos y calendario; requerimientos de instalaciones; requerimientos para soporte automático, datos especiales, entrenamiento; estándares, prácticas y convenciones; actualizaciones y revisiones a los planes de aceptación y productos relacionados

Contenido del plan de aceptación

# Validación

## 3. DESARROLLAR EL PLAN DE ACEPTACIÓN

### Procedimientos administrativos

Reportes de anomalías; control de cambios; mantenimiento de registros; comunicación entre desarrollador y administrador.

### Descripción de la aceptación

Objetivos de todo el proyecto; resumen del criterio de aceptación; actividades de aceptación mayores y revisiones; requerimientos de información; tipos de decisiones de aceptación; responsabilidad para decisiones de aceptación.

Contenido del plan de aceptación

# Validación

## 3. DESARROLLAR EL PLAN DE ACEPTACIÓN

### Revisión de cada aceptación

Productos para aceptación; objetivos de cada revisión; criterio de aceptación; fuente de información adicional para cada producto; requerimientos de aceptación; enfoque general; técnicas de prueba y examinación y soporte automático requerido.

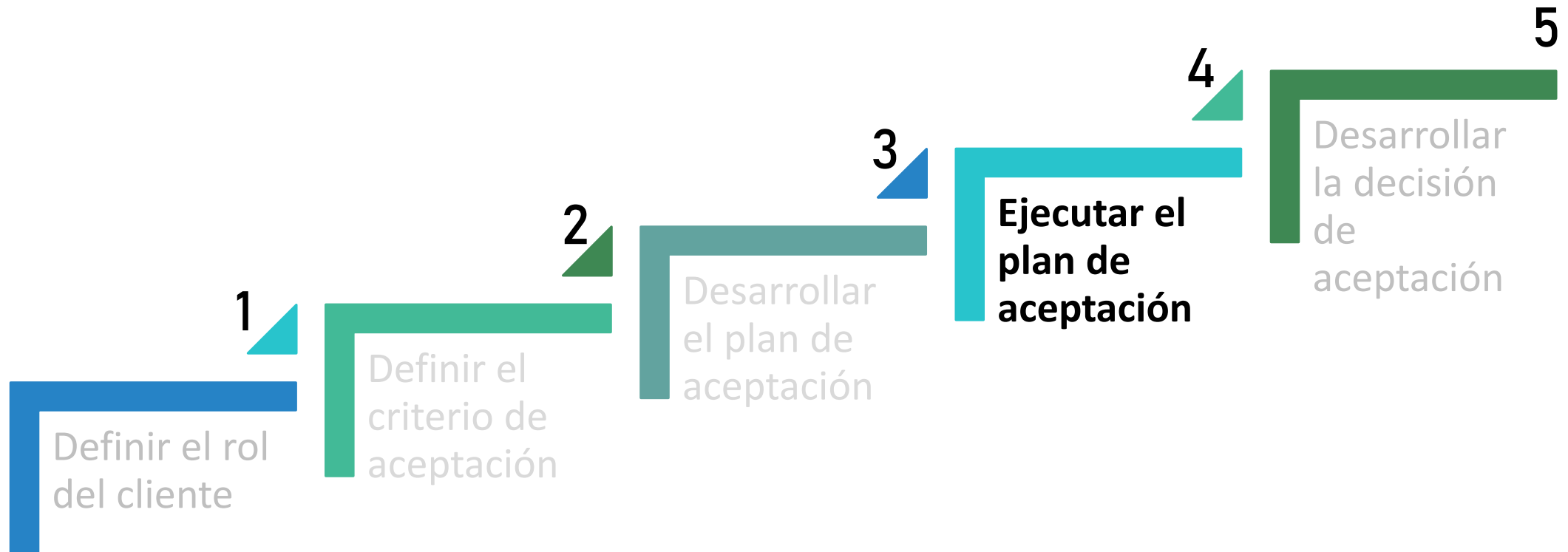
### Prueba de aceptación final

Plan de prueba y criterio de aceptación; casos de prueba y procedimientos; resultados de pruebas y análisis; instalaciones; staff.

Contenido del plan de aceptación

# Validación

## ■ Proceso de aceptación del software



# Validación

## 4. EJECUTAR EL PLAN DE ACEPTACIÓN (Conducción del plan de aceptación y revisiones)

- El objetivo de esta fase es determinar si el criterio de aceptación ha sido cumplido en el producto liberado.
- Puede ser completado a través de revisiones, lo que implica observar en productos provisionales y entregables parciales en varios puntos a través del proceso de desarrollo.
- Asimismo puede implicar testear el sistema de software ejecutable.

# Validación

## 4. EJECUTAR EL PLAN DE ACEPTACIÓN (Conducción del plan de aceptación y revisiones)

- La determinación de cuál de las dos técnicas (revisiones, ejecuciones) emplear, o ambas, depende del nivel de criticidad del software, el tamaño del software, los recursos implicados y el periodo de tiempo de desarrollo.
- La aceptación del software está especificada en un plan formal que identifica los productos a ser testeados, el criterio éxito/falla, las revisiones y los tipos de testing que ocurrirán a través de todo el ciclo de vida.

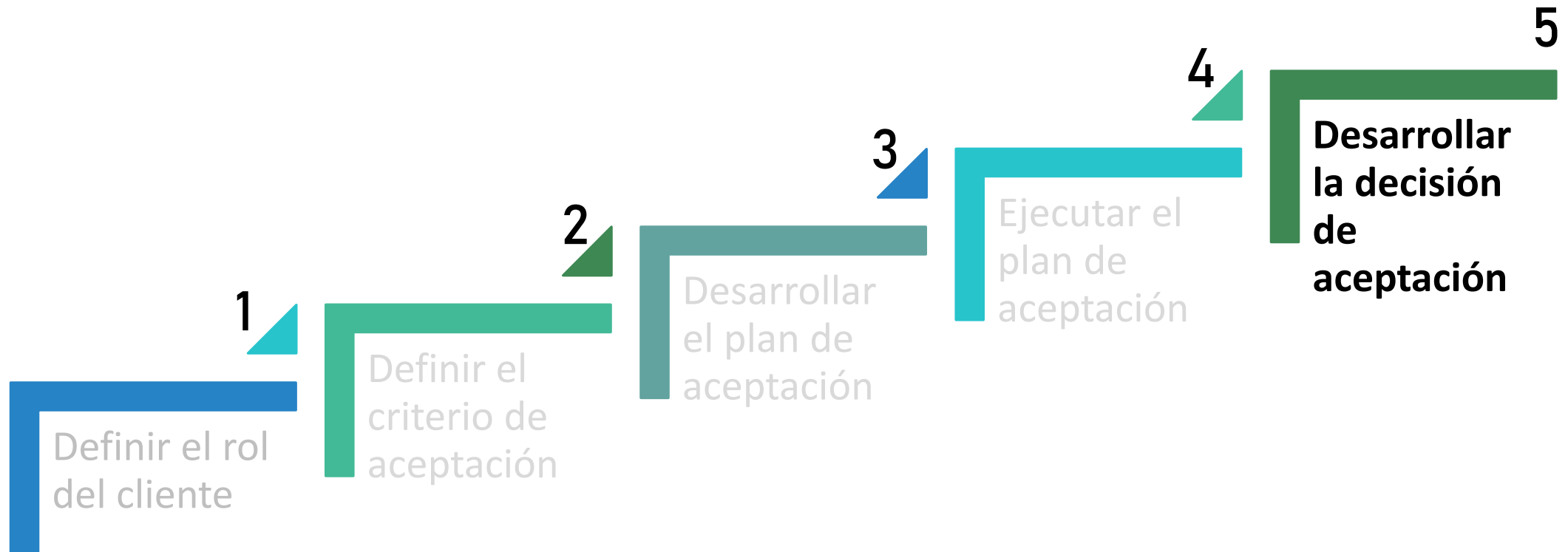
# Validación

## 4. EJECUTAR EL PLAN DE ACEPTACIÓN (Conducción del plan de aceptación y revisiones)

- La prueba de aceptación final es la última oportunidad para el cliente de revisar el software en el aspecto funcional, de interfaz, rendimiento, seguridad, seguridad de software ty características de calidad.
- En este punto, el software debe incluir: el software liberado y toda la documentación de usuario.

# Validación

## ■ Proceso de aceptación del software





# Validación

## 5. DESARROLLO DE LA DECISIÓN DE ACEPTACIÓN

- La aceptación final del software basado en pruebas de aceptación de software significa que el contrato y el proyecto han sido completados, con excepción de algunas advertencias o contingencias de aceptación.
- Se realiza el pago del software y el desarrollador no tiene más obligaciones (el mantenimiento es otra fase).
- La decisión de aceptación puede incluir:

# Validación

## 5. DESARROLLO DE LA DECISIÓN DE ACEPTACIÓN

1. Se requieren cambios y se acepta antes de continuar a la siguiente fase.
2. Se requieren algunos cambios y se acepta en la siguiente revisión mayor.
3. El progreso continua y los cambios son aceptados a la siguiente revisión.
4. No se requieren cambios y la aceptación continua.

# Validación

## 5. DESARROLLO DE LA DECISIÓN DE ACEPTACIÓN

- ❑ Aunque la meta es lograr y aceptar solo software perfecto, muy probablemente algunos criterios no serán completamente satisfechos por cada producto.
- ❑ El cliente puede elegir aceptar el producto con criterios insatisfechos.

# GRACIAS POR SU ATENCIÓN

---