

Verificación y validación de software

UNIDAD II

Planeación de la verificación y validación del software

Planeación de la verificación y validación del software

Unidad II.

Planeación de la verificación y validación del software

- Roles de la V&V

Planeación de la verificación y validación del software

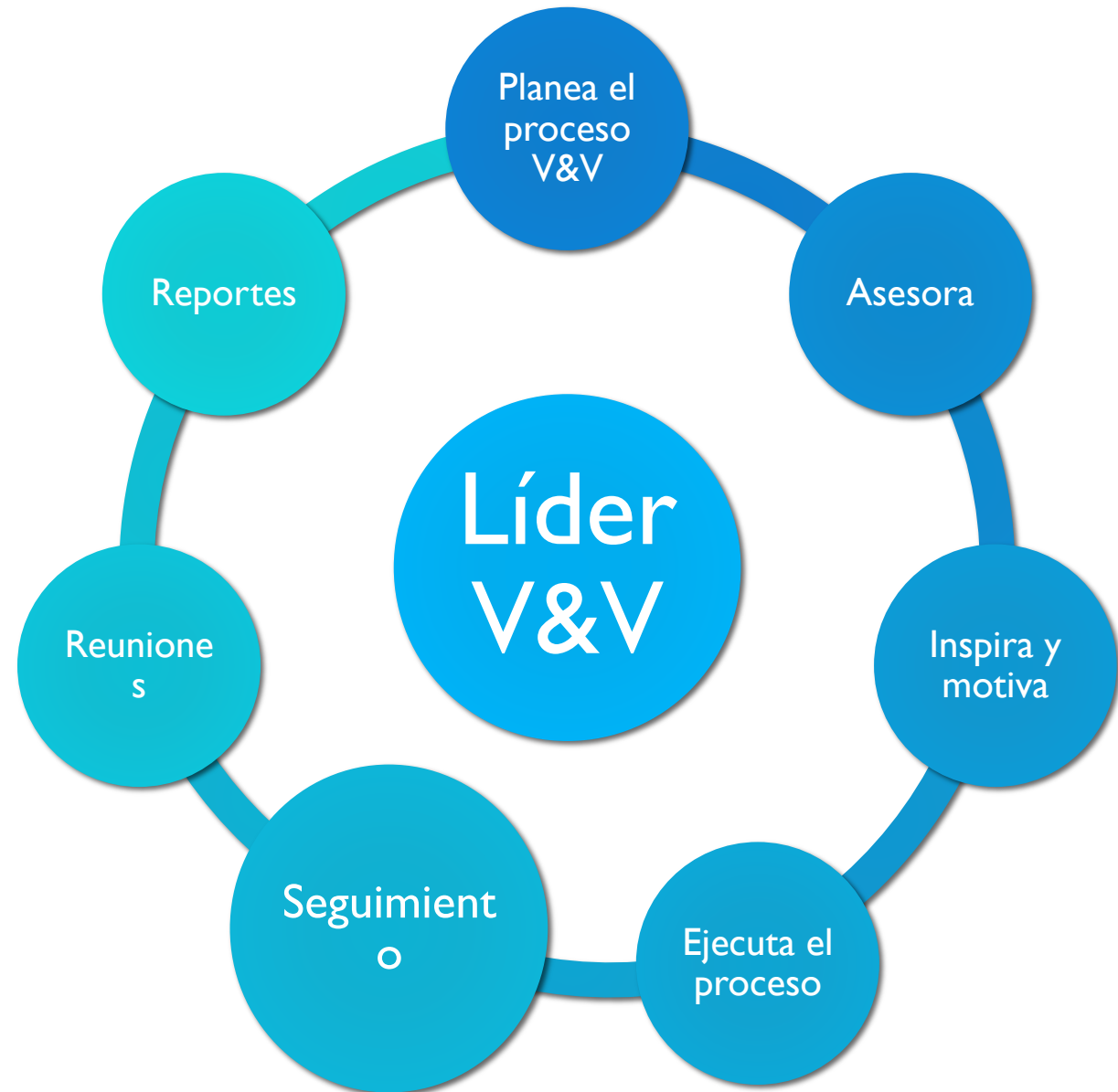
Rol.

Responsabilidad asignada a un individuo verificador cuando se encuentra ejecutando revisiones individuales.

El objetivo de asignar roles es incrementar la productividad del equipo.

Planeación de la verificación y validación del software

Rol del Líder de V&V



Planeación de la verificación y validación del software

Cada individuo, miembro del equipo de verificación, requiere concentrarse en aquello en lo que está interesado y es competente. Mientras sus compañeros de equipo se hacen cargo de otras tareas al mismo tiempo.

Es tarea del líder de V&V identificar y asignar roles de forma efectiva.

Planeación de la verificación y validación del software

- Más de una persona pueden tener el mismo rol y un verificador puede tener varios roles.
- Asignar roles considerando los talentos e intereses de cada individuo y con su consentimiento

Planeación de la verificación y validación del software

Tipos de roles

Roles por documento.

Roles definidos por listas de verificación y/o por procedimientos.

Planeación de la verificación y validación del software

Tipos de roles

Roles por documento.

Se le otorga una responsabilidad especial a un individuo para poner particular atención en uno o más documentos o artefactos utilizados en el proceso de verificación.

Planeación de la verificación y validación del software

Tipos de roles

Roles definidos por listas de verificación y/o por procedimientos.

- Una forma de definir roles es escribir cuatro preguntar en formas de Checklist que definan las tareas a realizar.
- Esto es mejor que otorgar un nombre “tester”, “risk analyst” y asumir que el verificador sabe qué hacer.

Planeación de la verificación y validación del software

Por ejemplo,

El checklist para estimaciones puede tener preguntas acerca del rendimiento o estimaciones de entregas en el proyecto.

Roles definidos por procedimientos.

Contiene un checking de procedimiento.

Planeación de la verificación y validación del software

Ejemplos:

- Concentrate on finding issues in source documents. **Document role**
- Check for ambiguity, unintelligibility. **Checklist role.**
- Check that there is a test planned for each requirement stated. **Procedure role.**

Planeación de la verificación y validación del software

Ejemplo de los diferentes perfiles que forman parte del equipo V&V

Planeación de la verificación y validación del software

Tipo	Contract	Reqts.	Archit. Design	Detail Design	Test Plan	Test Design	Source Code	Tech. Doc.	User Manual
Task author	x	x	x	x	x	x	x	x	x
Req. analyst	x	x	x	x	x	x			x
Architectural designer	x	x	x	x	x	x		x	
Detailed designer			x	x	x	x	x	x	
Programmer				x	x	x	x	x	
Tester	x	x	x	x	x	x	x	x	x
Maintainer		x	x	x	x	x	x	x	x
User	x	x	x		x	x			x
Manager	x	x	x		x				x
Marketing	x	x	x						x
Legal Department	x	x						x	x

Planeación de la verificación y validación del software

User – Se concentra en el punto de vista del cliente o el usuario.

Tester – Se concentra en las consideraciones para las pruebas (testability, requerimientos para pruebas, orden de las pruebas, desarrollo para pruebas en paralelo, etc.)

System – Se concentra en las implicaciones más amplias del sistema (HW, documentación, ventas, tiempos de entrega)

Planeación de la verificación y validación del software

Financial – Se concentra en costos e ingresos, estimaciones, incertidumbre, fechas, cantidades.

Quality – Se concentra en todos los aspectos de atributos de calidad.

Service – Se concentra en lo relativo al campo de servicio, mantenimiento, suministro, instalación.

Rules – Pone especial atención en las reglas para el producto específico.

Planeación de la verificación y validación del software

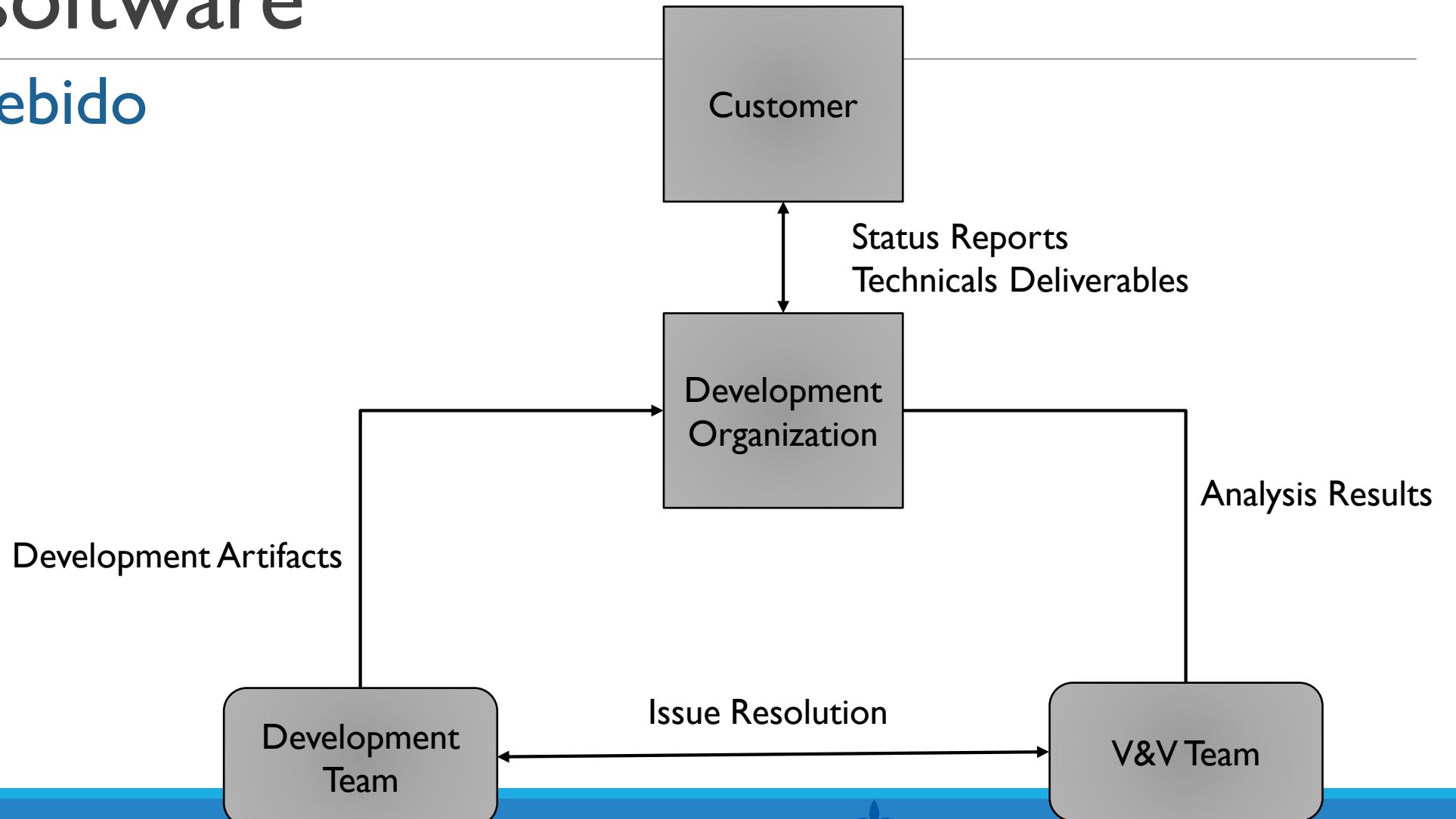
Participantes y Roles

ANALYST	Document Roles	Checklist-defined roles
John	TEST PLAN	ESTIMATES
Mary	CONTRACT	TESTING
Stephen	REQT, SYSTEM TEST PLAN	QUALITY
Ruth	DESIGN - RULES	LEGAL

Estructura de comunicación

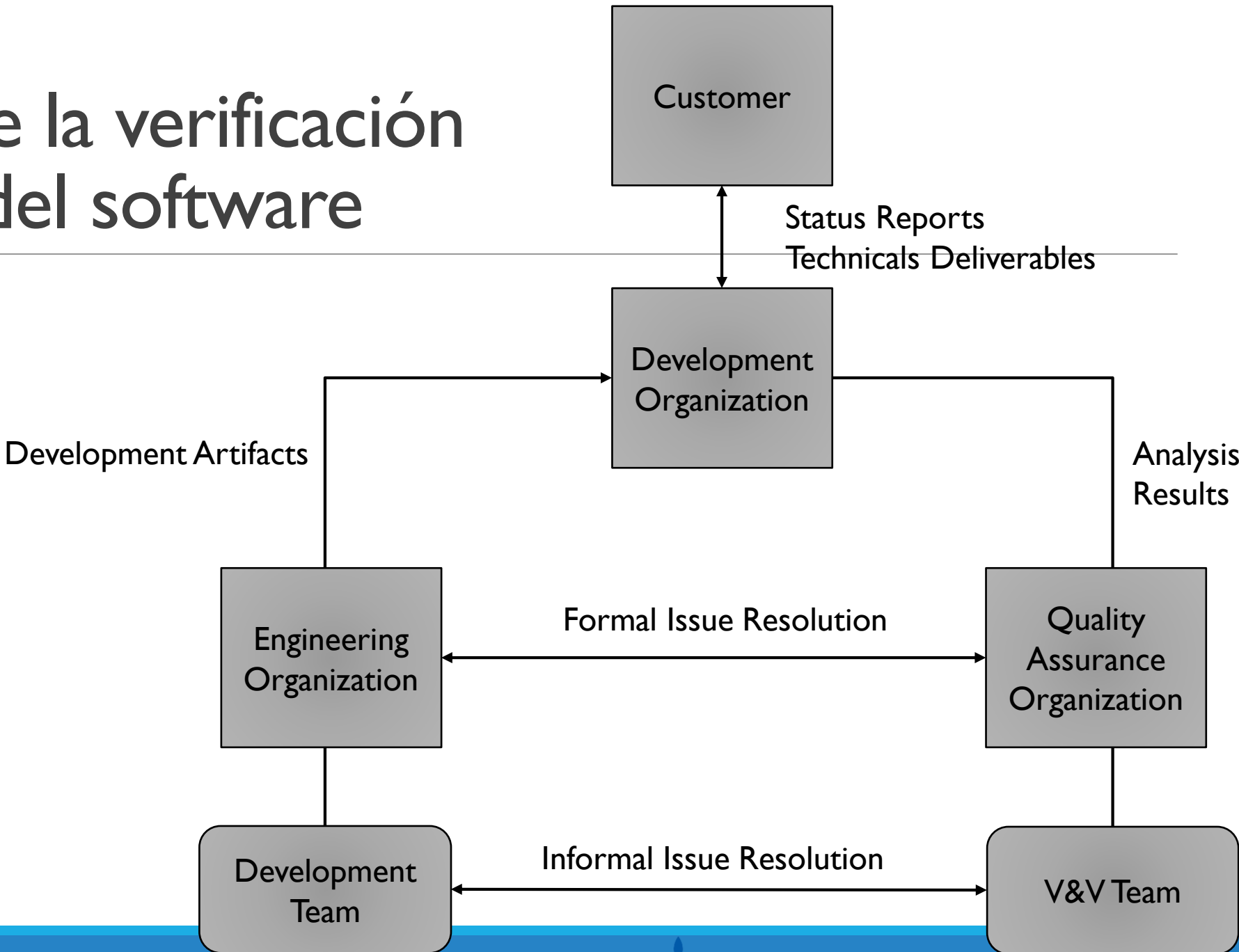
Planeación de la verificación y validación del software

Embebido



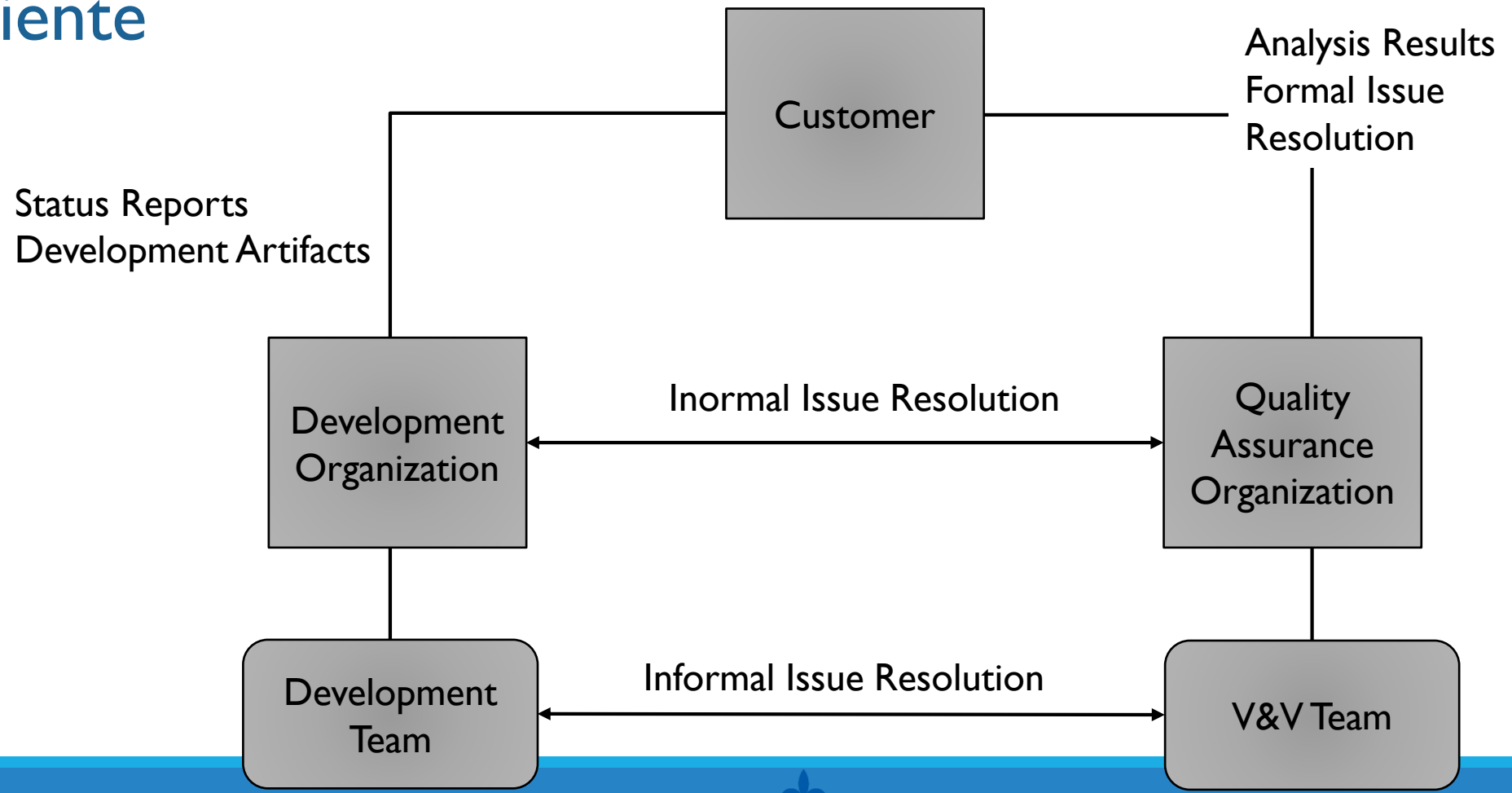
Planeación de la verificación y validación del software

Interno



Planeación de la verificación y validación del software

Independiente



Plan del Proyecto V&V

Planeación de la verificación y validación del software

Project Plan Topic	Description
Overview	Describe el sistema sobre el cual se establecerá la V&V. Incluye: Meta V&V Stakeholders y sus requerimientos. Objetivos V&V
V&V Concept	Roles y responsabilidades
Resources	Recursos requeridos para ejecutar el plan.
Technical Scope	V&V requeriments V&V Scope
Tools and Methods	Herramientas y Métodos a emplear.
Schedule	Calendario general

Planeación de la verificación y validación del software

Overview – Deja claro lo que la V&V logrará y qué se considera exitoso.

Concept – Estructura de comunicación V&V, roles y responsabilidades.

- La autoridad para resolver problemas.
- La autoridad para aprobar productos V&V.
- Autoridad para hacer cambios al plan.
- Responsabilidades para cada elemento.

Planeación de la verificación y validación del software

Resources – Recursos requeridos para realizar las tareas V&V. Staff, instalaciones, herramientas, financiamiento, requerimientos especiales (accesos de seguridad, documentación, etc.)

Planeación de la verificación y validación del software

Technical Scope – Requerimientos V&V del sistema, objetivos específicos, requerimientos asociados a cada objetivo.

Tools and Methods – Técnicas y métodos empleados para el proyecto V&V. (Análisis manual, Análisis estático, Análisis dinámico, Análisis formal)

- Descripción del método.
- Riesgo asociado al método.

Planeación de la verificación y validación del software

Schedule – Resume las tareas V&V, entregables y fechas.
Diagrama de Gantt.

V&V Task	V&V Requirement Fulfilled	V&V Approach	Duration Days (Min, Avg, Max)
Identify a potential faults	None	Static Analysis {Use cases and Scenarios}	(1,2,3)
Identify the interfaces required to identify and handle these faults	None	Static Analysis {Use cases and Scenarios}	(2,4,6)
Identify the data items that should be passed between interfaces	None	Static Analysis {Use cases and Scenarios}	(3,7,10)
Compare and contrast the results with the system and software interface requirements	3.2.1	Manual Analysis	(5,7,12)

Gracias por su atención