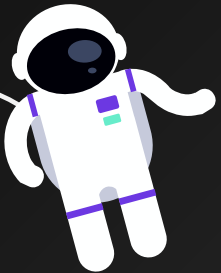


Administración de proyectos Informáticos

Gestión de Proyectos

Agenda

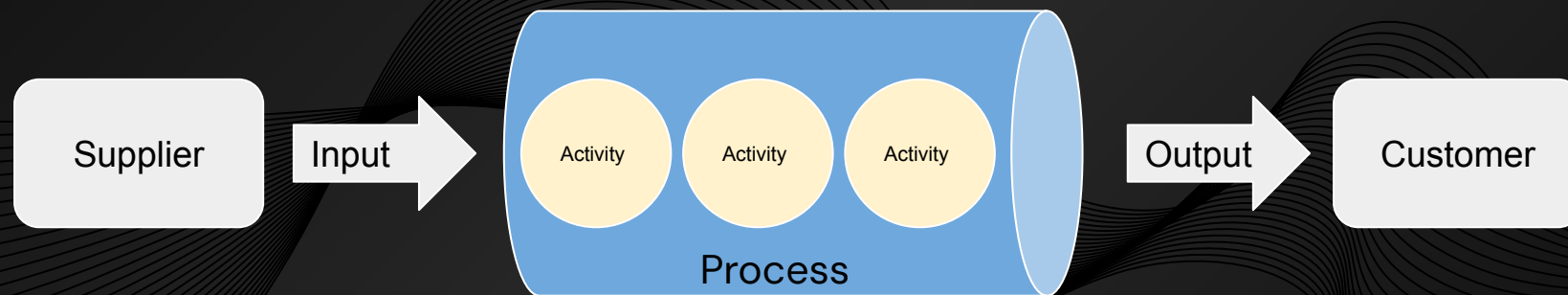
- Modelos, procesos, técnicas y herramientas
- Ciclos de vida del proceso, producto y gestión
- Ciclo de vida del software



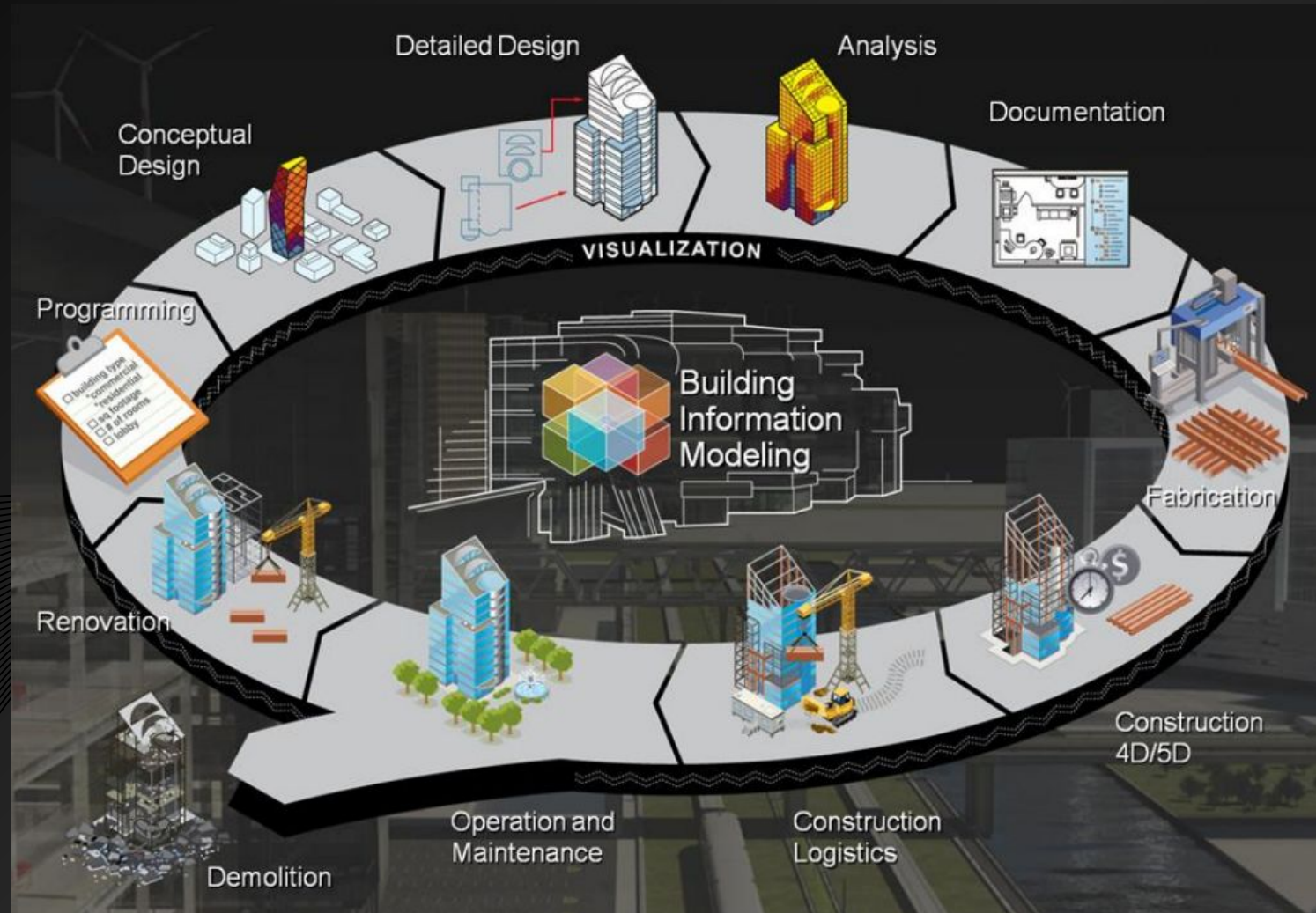
Que es un Proceso??

Definición de proceso

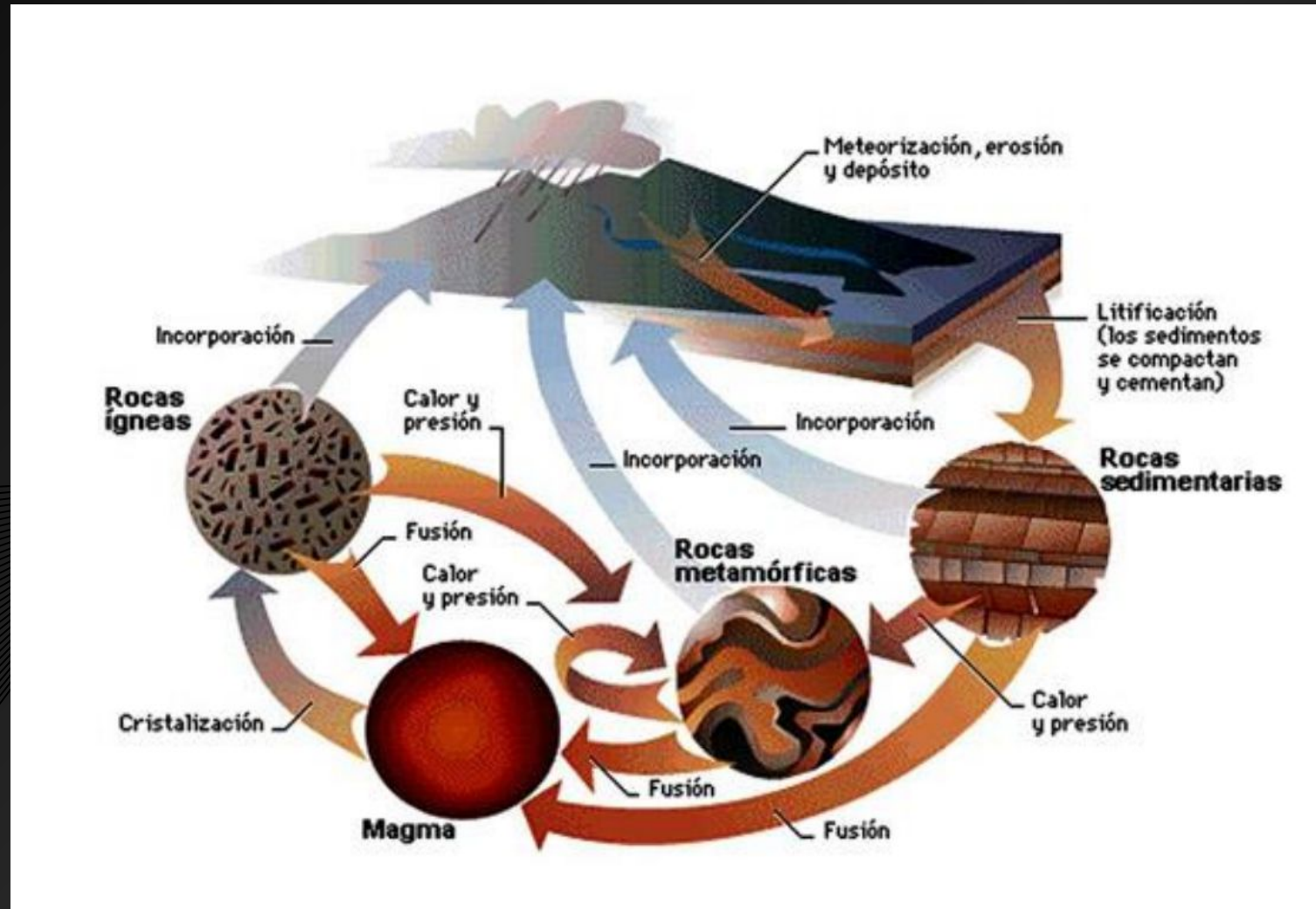
Un proceso es una serie interrelacionada de actividades que convierten las entradas del negocio en los resultados del negocio.



Ciclo de vida



Ciclo de vida



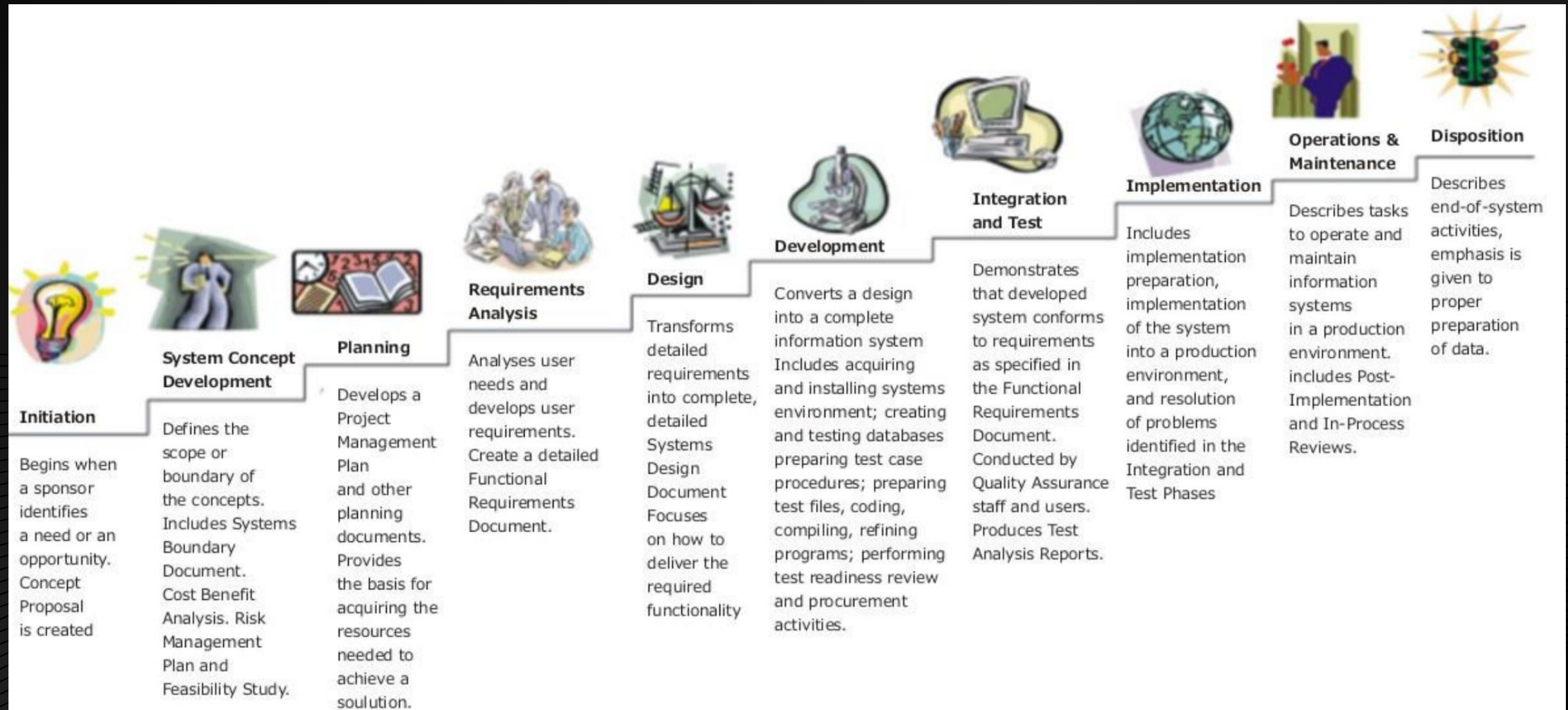
Ciclo de vida

Modelo en Cascada



Ciclo de vida

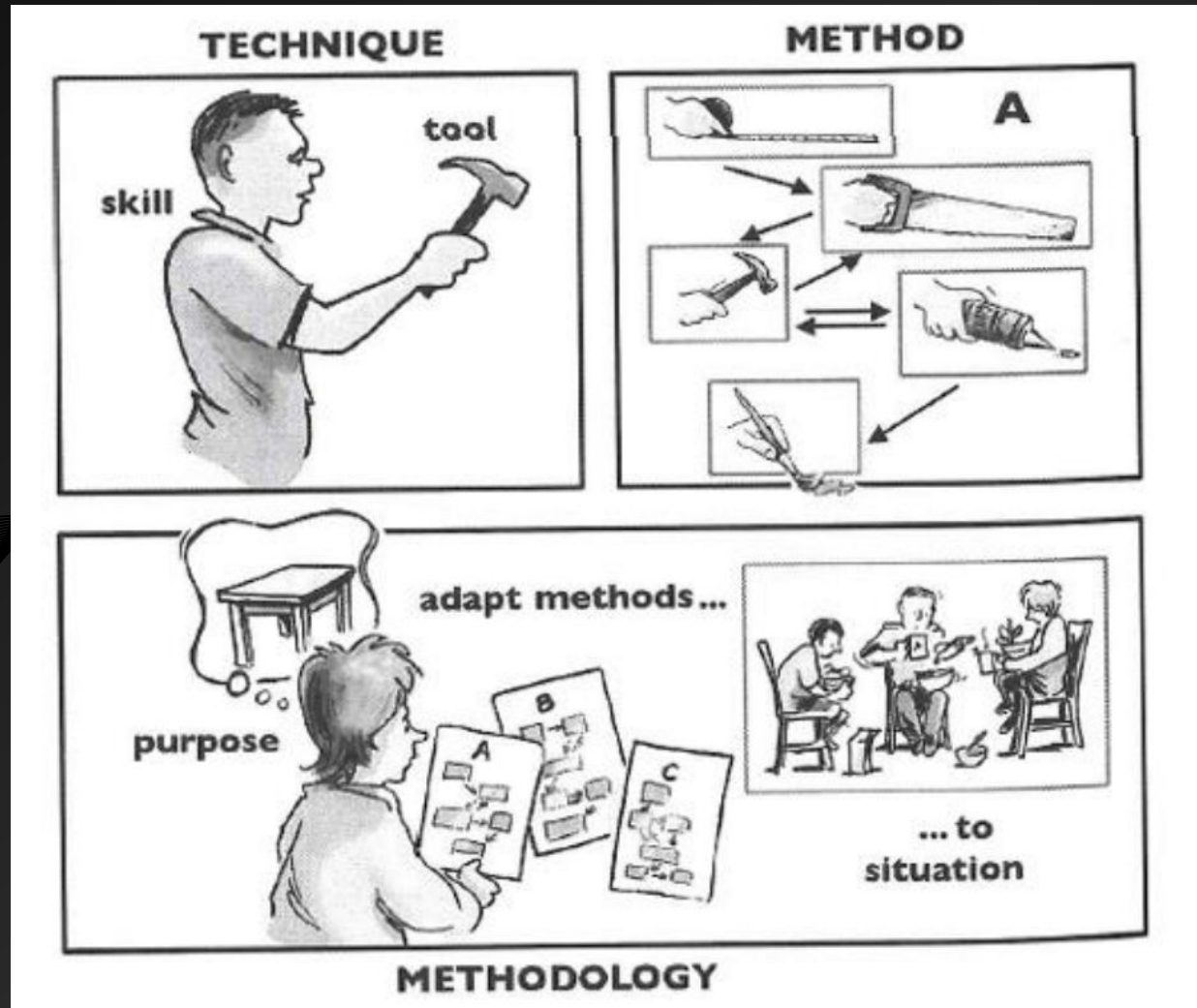
Fases del ciclo de vida



Metodo

- Palabra que proviene del término griego *methodos* ("camino" o "vía").
- Se refiere al medio utilizado para llegar a un fin.
- Modo ordenado y sistemático de proceder para llegar a un resultado o fin determinado.

Technique, Method, Methodology



Tecnica

Mecánica que se practica en una situación controlada para adquirir la habilidad

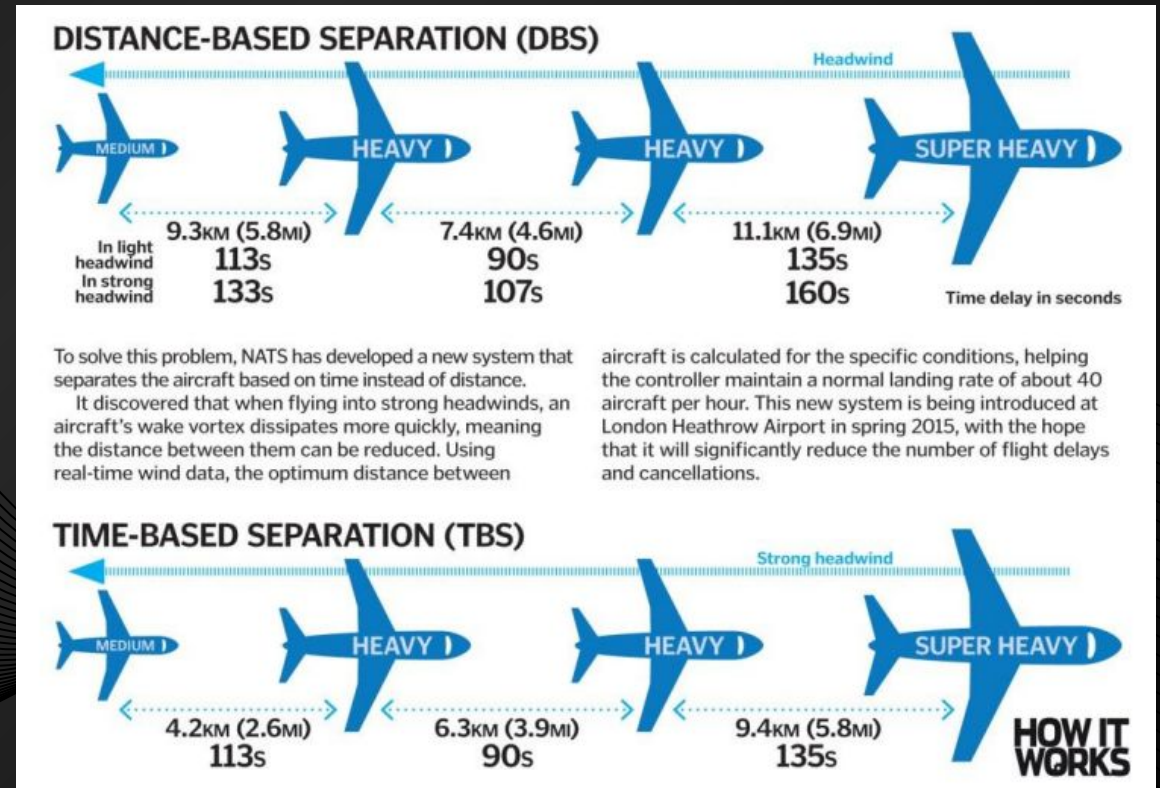


Técnica (La realidad)

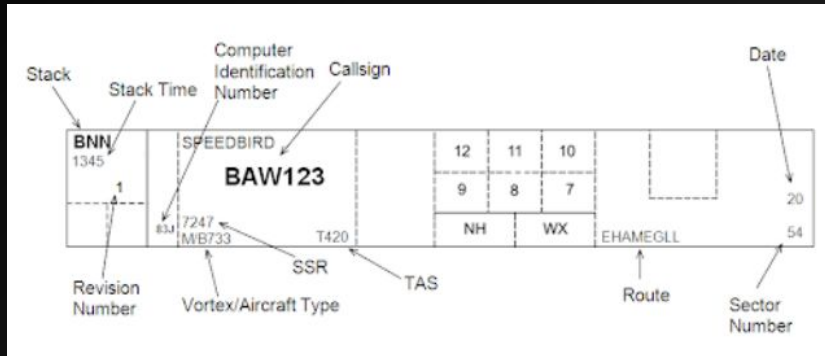
Necesidad de observar, adaptarse e improvisar



Método vs Herramienta



Herramientas



- Flight progress strip
- Digital progress strip
- Status board

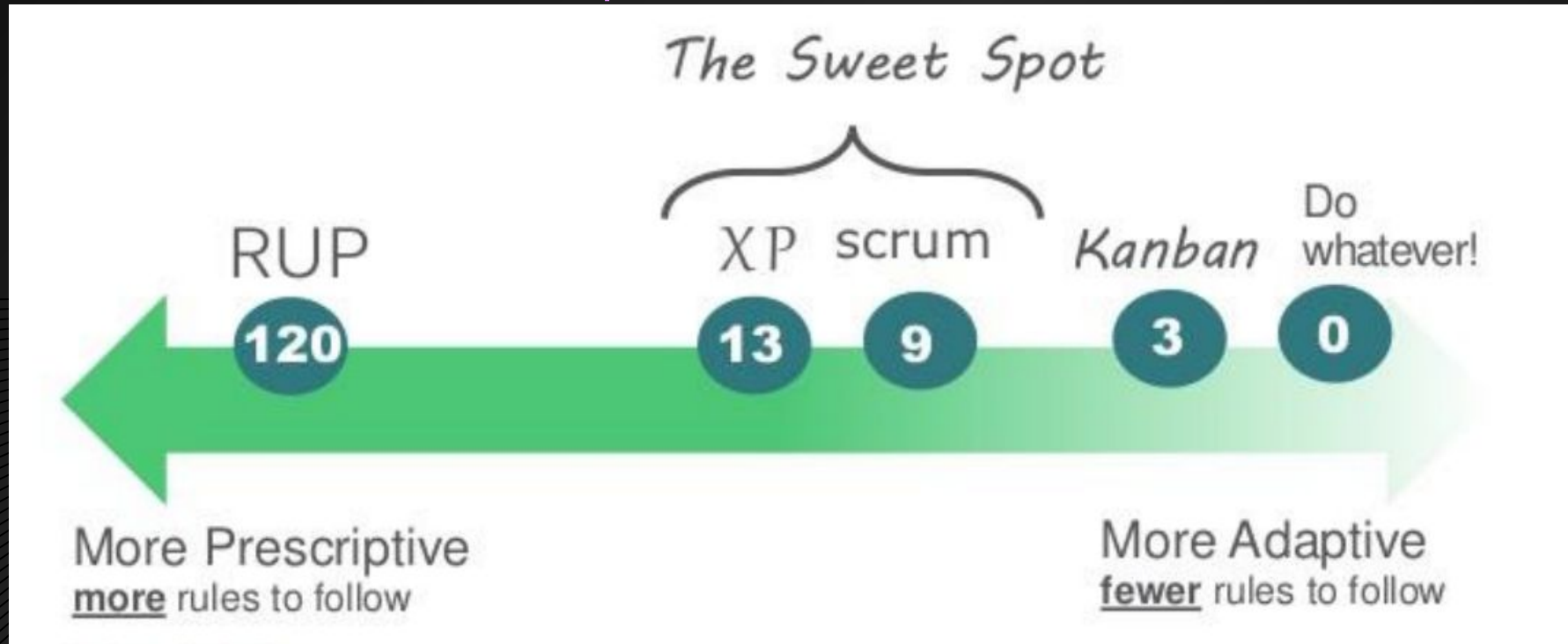
LOCAL	RUNWAY	16/34	1	2	3	4	5	TAXI	STRIP STORAGE
ARRIVALS	ETA								STANDING BY
AUA920A	A380	F04	34	CTC					
AUA910A	A380	F04	34	CTC					
BER882A	A320	H42	34	CTC					
AUA832	A321	F26	34	CTC					
AUA862A	B777	F42	34	CTC					
BT1208	A320	D25	34	CTC					
AUA9508	A321	H46	34	CTC					
AUA798	A320	H43	34	CTC					
AFR123	F70	C31	34	CTC					
AUA842	A321	F23	450	3253	34	LAND			
BAW704A	A43	4543	34	LAND					
A320 F08	104								
AUA830A	A43	1003	34	TDN					
A380 F13	104								
LOT226	K0X3C	4573	29	LUP-BAR					
A319 F37	SPW								
AUA900A	A43	5433	34	TAXI					
A380 F04	104								
AUA117	S013C	4630	29	TAXI					
F100 H49	EDW								
NLY674	OSP/L	7142	TCAS	TOCL					
A319 B75	LNL		29	BAR					
BER882D	A321	H42	25						
AUA862D	B777	F42	25						
BAR704B	A321	H46	25						
AUA831	A321	H46	25						
AUA830D	A321	H46	25						
AUA900D	A321	H46	25						
AUA913	A321	H46	25						



RCVY	APP	FB	DW	PROFILE
WATD	SIDE	TM	ST	T
C 1830	112	184	32	/
I 1841	113	1	32	/
W 1833	204	26	75	32
I 1837	210	22	112	32
W 1831	300	24	144	30
I 1839	314	21	114	30
I 1835	304	23	98	32
I 1813	301	23	98	32
I 1835	404	24	98	32
I 1832	400	25	98	32
W 1845	502	23	150	35
I 1832	503			35
W 1826	703	26	151	22
W 1829	601	23	75	12
W 706				22
W 701				22

SDLC(Systems Development Life Cycle)

Prescriptive vs
Adaptive



Modelos de proceso de software

¿Para qué modelos de proceso?

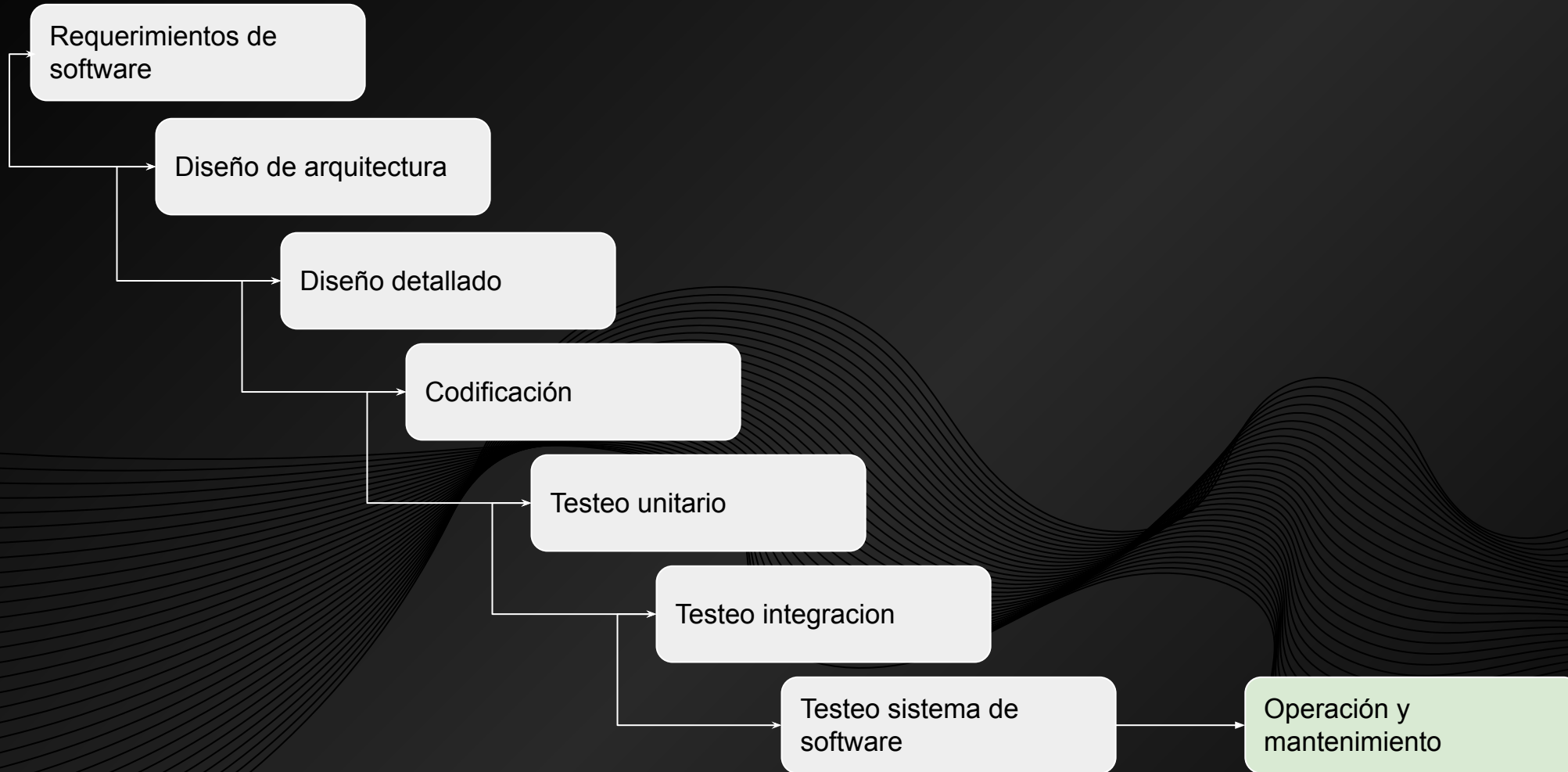
- Cuando un grupo de desarrollo describe su proceso de desarrollo, forma una comprensión común de las actividades, recursos y restricciones involucradas en el desarrollo de software
- Ayuda encontrar inconsistencias, redundancias y omisiones
- El modelo debería reflejar los objetivos de desarrollo (tales como construir software de alta calidad, encontrar fallas tempranas en el desarrollo y satisfacer requerimientos de presupuesto y restricciones de cronograma).
- Permite evaluar actividades candidatas para atacar estos objetivos.

Modelos de proceso de software

Modelos y Métricas

- Las mediciones capturan información sobre el proceso que se está ejecutando
- Permiten:
 - Evaluar el proceso
 - Profundizar el conocimiento del producto
 - Detectar las debilidades del ambiente
 - Proveer conocimiento sobre la mejora del proceso
 - Ayudar en la evolución del proceso

MCV Cascada Completo



Modelo de cascada clásico

- Requiere definir qué hará el sistema antes de construirlo (REQUERIMIENTOS/DISEÑO)
- Requiere planear la interacción de las partes antes de construirlas (DISEÑO/CODIFICACIÓN)
- Una etapa debe concluir antes de comenzar la subsiguiente
- Recorre las etapas secuencialmente
- Cuando se requiere, vuelve atrás y rehace el material: en la práctica es típicamente iterativo
- Los gerentes deben seguir muy de cerca el proyecto y corregir muy tempranamente
- Históricamente fue el primero de todos los modelos
- Fue utilizado en una variedad de contextos y aún continúa siendo el modelo de referencia que ha dado lugar a múltiples modelos
- Indudablemente ayuda a los desarrolladores a delinear qué tienen qué hacer
- Es más eficiente cuando el problema y la solución son bien conocidos, esto es cuando no hay iteraciones en el ciclo de documentos.
- Requiere producir una serie de documentos que luego se usan en el testeo y mantenimiento
- Reduce los costos de desarrollo y mantenimiento como consecuencia de las razones anteriores
- Permite a la organización de desarrollo ser más estructurada y organizada

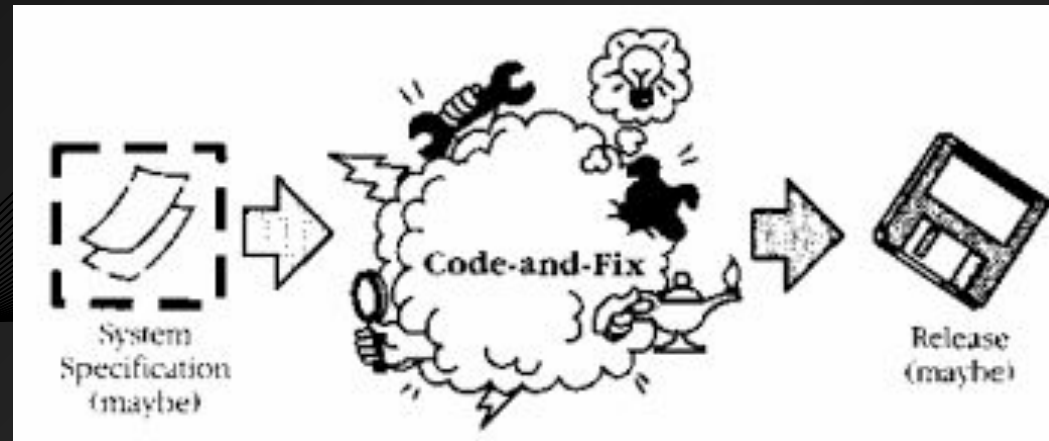
Ventajas del cascada

- Buen desempeño con definición estable del producto y con metodologías bien comprendidas
- Como todo el planeamiento se hace al principio, minimiza el overhead del planeamiento
- No provee productos tangibles de software hasta el final, pero la documentación que generan provee indicaciones de progreso
- Funciona bien en proyectos complejos y bien comprendidos
- Funciona bien cuando los requerimientos de calidad predominan sobre los de costo y cronograma
- Trabaja bien con gente técnicamente débil o inexperta, porque provee una estructura que minimiza el desperdicio de esfuerzo

Desventajas del cascada

- Las desventajas provienen de la dificultad de especificar los requerimientos al comienzo del proyecto antes de contar con un diseño y con código
- El desarrollo de software es iterativo en su esencia
- No trata al desarrollo como una actividad de resolución de problemas, sino como de “satisfacer una lista de pedidos”
- Reparar un olvido en el desarrollo puede ser muy costoso
- Ciertos errores de requerimientos sólo se pueden apreciar en momentos de construcción avanzada del sistema
- En los hechos de una etapa se pasa a cualquiera de las otras
- El mayor problema es la falta de flexibilidad (en el desarrollo real en muchos casos no se trata de construir todo, sino lo máximo posible)
- Se puede volver atrás, pero con un gran esfuerzo
- Para un desarrollo rápido genera demasiada documentación. Hasta el final genera pocos signos visibles de progreso, con lo que crea la percepción de avance lento
- Se ha convertido en un “modelo ideal típico”

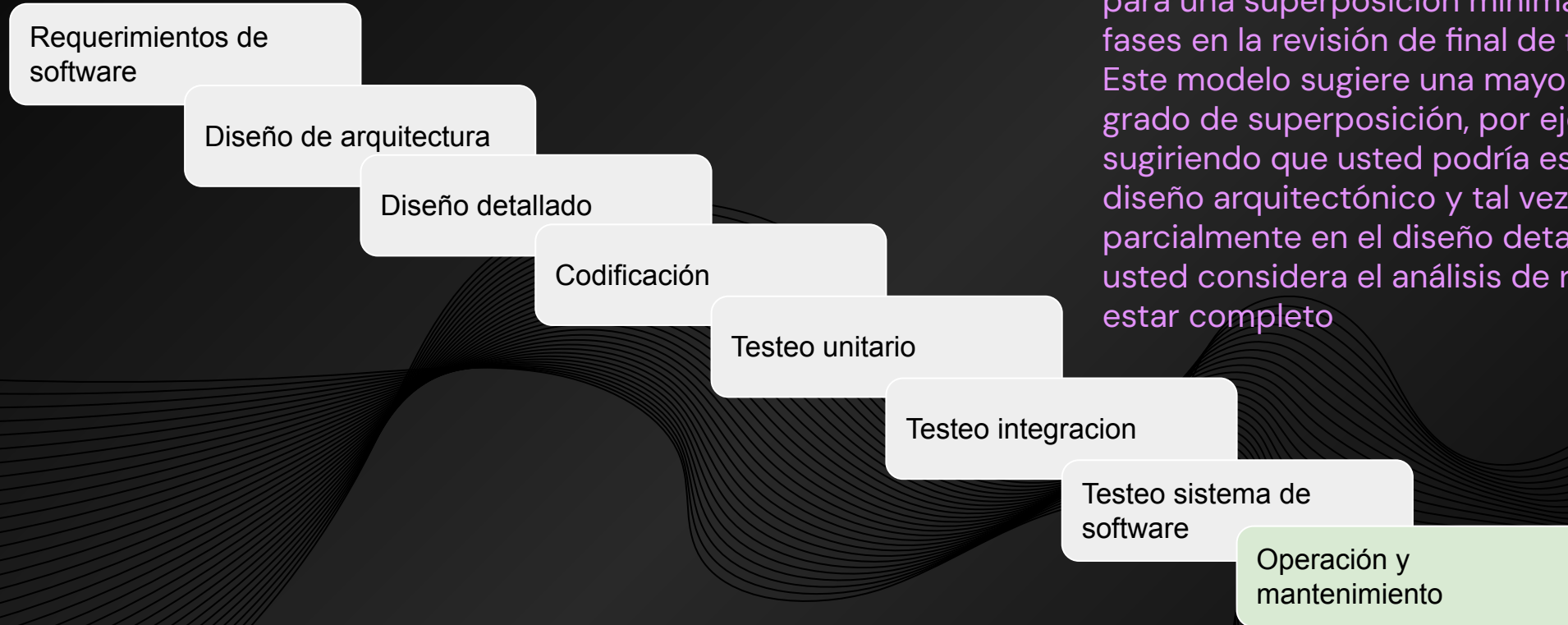
Code and Fix



Code and Fix (Características)

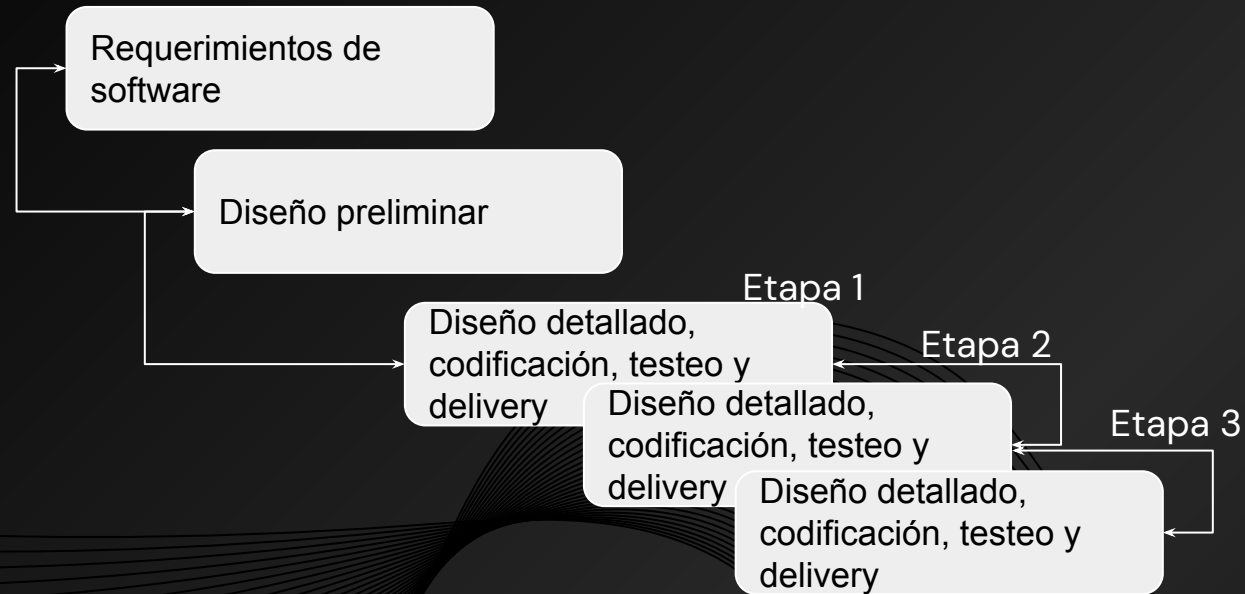
- Características generales
 - Habitualmente es el que se usa cuando no se usa ningún ciclo de vida
 - Si no hay mucho planeamiento seguro se usa este ciclo
 - Se parte de una idea general del producto (con o sin especificación)
 - Luego se usa una combinación de diseño informal, metodologías varias hasta que se tiene un producto para entregar
- Ventajas:
 - No hay overhead, no se pierde tiempo en ninguna actividad más que en codificar
 - No requiere entrenamiento: cualquiera que sepa programar está familiarizado con este ciclo
- Desventajas:
 - Salvo proyectos pequeños es un modelo peligroso: no se puede evaluar el progreso hasta que se hizo el código, no hay forma de evaluar calidad ni de identificar riesgos
 - Si se descubre que el enfoque de diseño es equivocado, no hay otra que volver a empezar

Cascada con superposición de fases



El modelo de cascada tradicional permite para una superposición mínima entre fases en la revisión de final de fase. Este modelo sugiere una mayor grado de superposición, por ejemplo, sugiriendo que usted podría estar bien en diseño arquitectónico y tal vez parcialmente en el diseño detallado antes usted considera el análisis de requisitos para estar completo

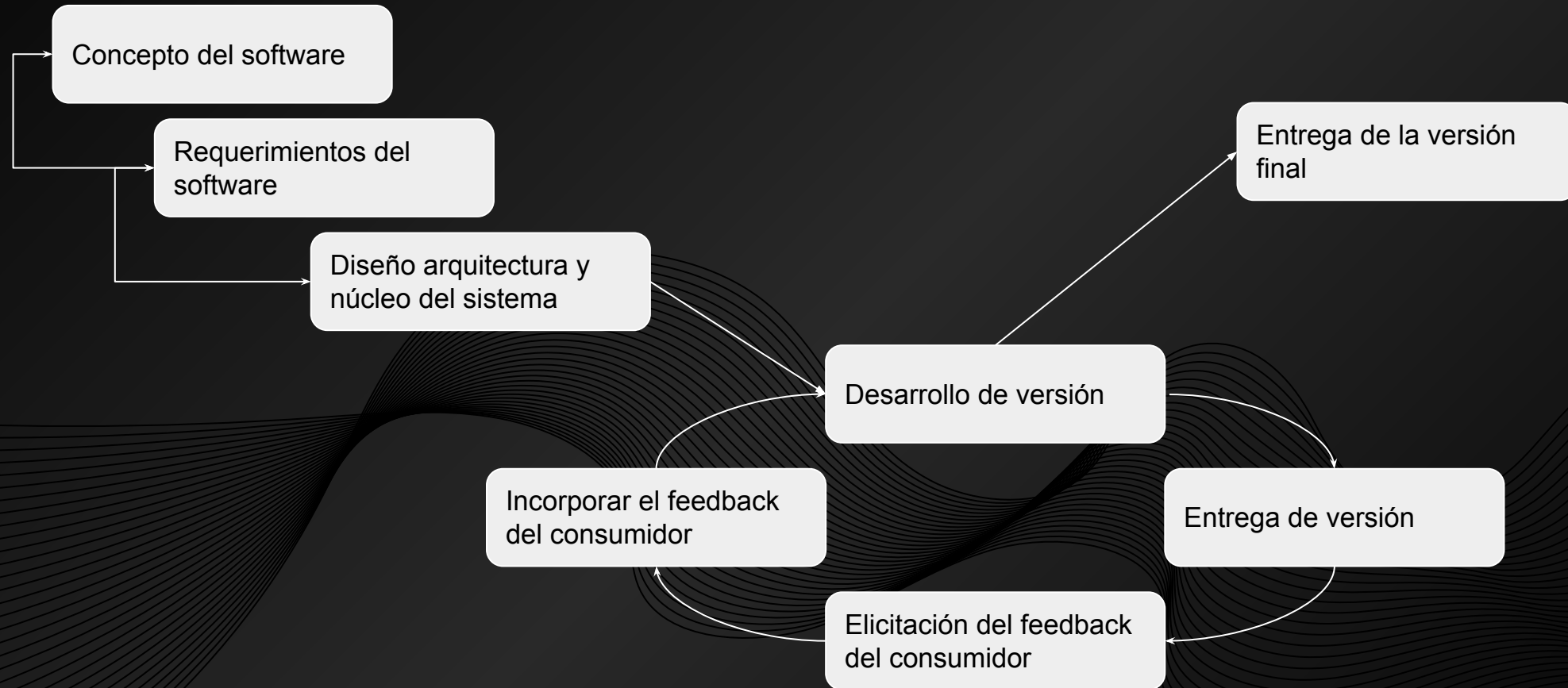
Entrega en etapas (Incremental)



The primary advantage of staged delivery is that it allows you to put useful functionality into the hands of your customers earlier than if you delivered 100 percent of the project at the end of the project. If you plan your stages carefully, you may be able to deliver the most important functionality the earliest, and your customers can start using the software at that point. Staged delivery also provides tangible signs of progress earlier in the project.

The main disadvantage of staged delivery is that it won't work without careful planning at both the management and technical levels. At a management level, be sure that the stages you plan are meaningful to the customer and that you distribute work among project personnel in such a way that they can complete their work in time for the stage deadline. At a technical level, be sure that you have accounted for all technical dependencies between different components of the product.

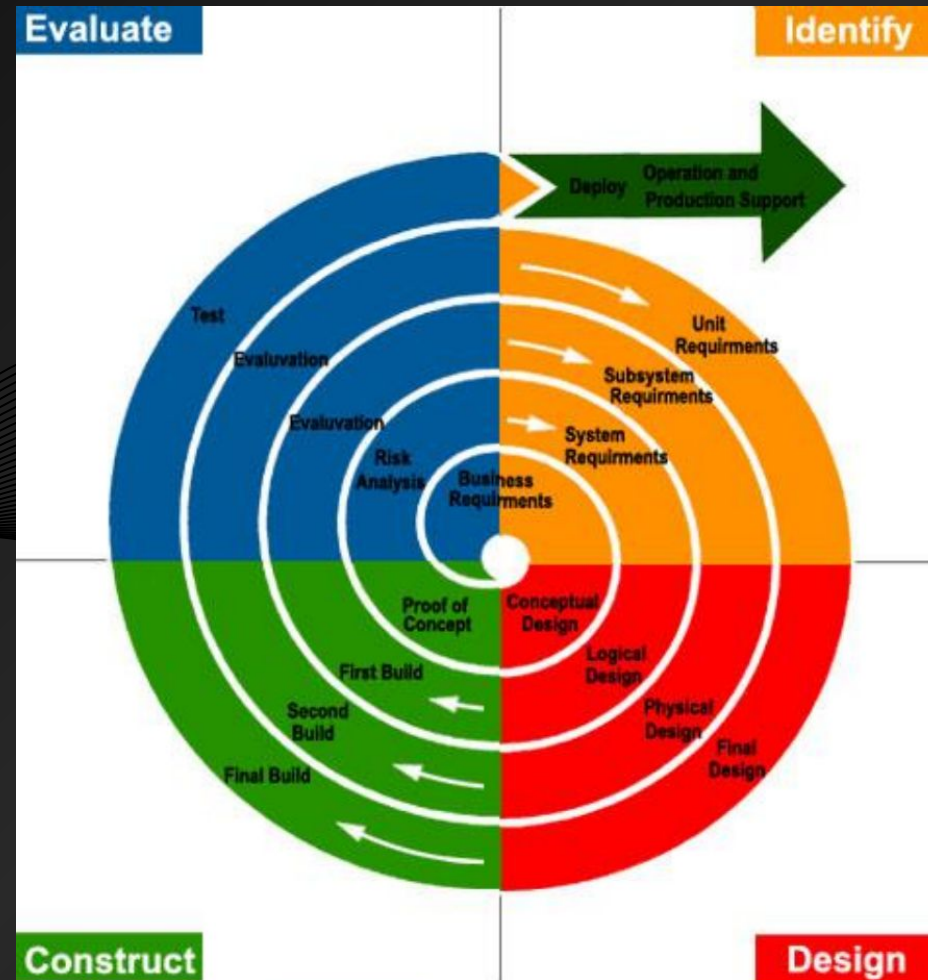
Entrega evolutiva



Prototipos

- **Rápidos desechables**, construcción “quick and dirty” que atacan un aspecto particular para que el usuario potencial lo utilice durante un tiempo y provea información al equipo.
- **Evolutivos**, atacan una funcionalidad acotada del sistema que resuelve algunos de los requerimientos del usuario.

Modelo en espiral



Modelo en espiral

- **Objetivos.** Objetivo general a alcanzar, luego se irá desglosando en más niveles. **Restricciones.**
- **Restricciones** que se imponen al proyecto.
- **Alternativas.** Se detallan las alternativas de solución.
- **Riesgos.** Se detallan los riesgos que podría generar cada alternativa.
- **Resolución de Riesgos.** Propuesta de solución a cada uno de los riesgos.
- **Plan de la etapa siguiente.** Plan de la etapa siguiente.
- **Compromiso.** Detalles de compromisos para llevarla a cabo.

Características del Modelo en espiral

- Es el ciclo de vida más sofisticado
- Es un ciclo de vida orientado al riesgo que parte el proyecto en pequeños mini proyectos, cada uno de los cuáles ataca algún riesgo
- El “riesgo” se refiere a no entender los requerimientos o la arquitectura, problemas de performance, etc
- Una vez que se atacaron los principales riesgos concluye como lo haría un cascada
- Se parte en pequeña escala, explora los riesgos, hace un plan para manejarlos y plantea un enfoque para la próxima iteración
- Cada iteración abarca los seis pasos del espiral
- Las iteraciones iniciales son las más baratas: se gasta menos en las iniciales
- No hay que tomar el diagrama muy rígidamente: pueden no ser los seis pasos o más/menos de los cuatro loops
- Se puede combinar con otros modelos de ciclo de vida de dos maneras:
 - Una vez que se redujeron los riesgos con iteraciones se puede seguir con otro ciclo
 - Dentro de un ciclo de vida, se puede incluir una iteración para resolver algún tipo de riesgo

Modelo en espiral

- **Ventajas**
 - Provee tanto control gerencial como el modelo tradicional de cascada
 - Cómo está orientado por los riesgos, provee información temprana sobre éstos
 - Salir del proyecto no tiene alto costo de salida en las etapas tempranas
- **Desventajas**
 - La única dificultad es que es complicado: requiere un gerenciamiento cuidadoso
 - Puede ser dificultoso definir hitos para establecer el siguiente loop
 - Si el proyecto es modesto, y no se requiere la flexibilidad y gestión de riesgo de este modelo, conviene no aplicarlo

¡Muchas Gracias!

