



APPROACHES

Material producido por la Universidad Peruana de Ciencias Aplicadas
UPC, 2021.



CONTENIDO

Objetivo de aprendizaje	3
Introducción	3
Agile.....	5
Lean	8
Design Thinking	9
IBM Garage Method	10
Conclusiones.....	12
Referencias	12



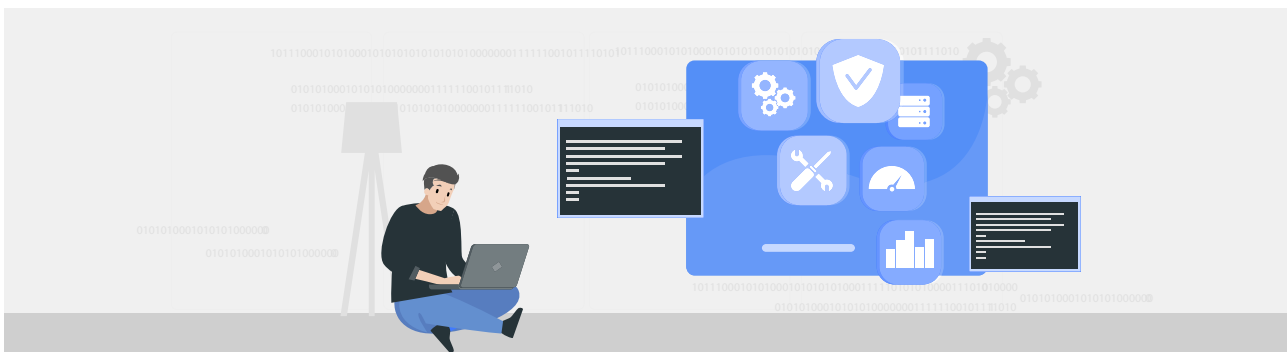
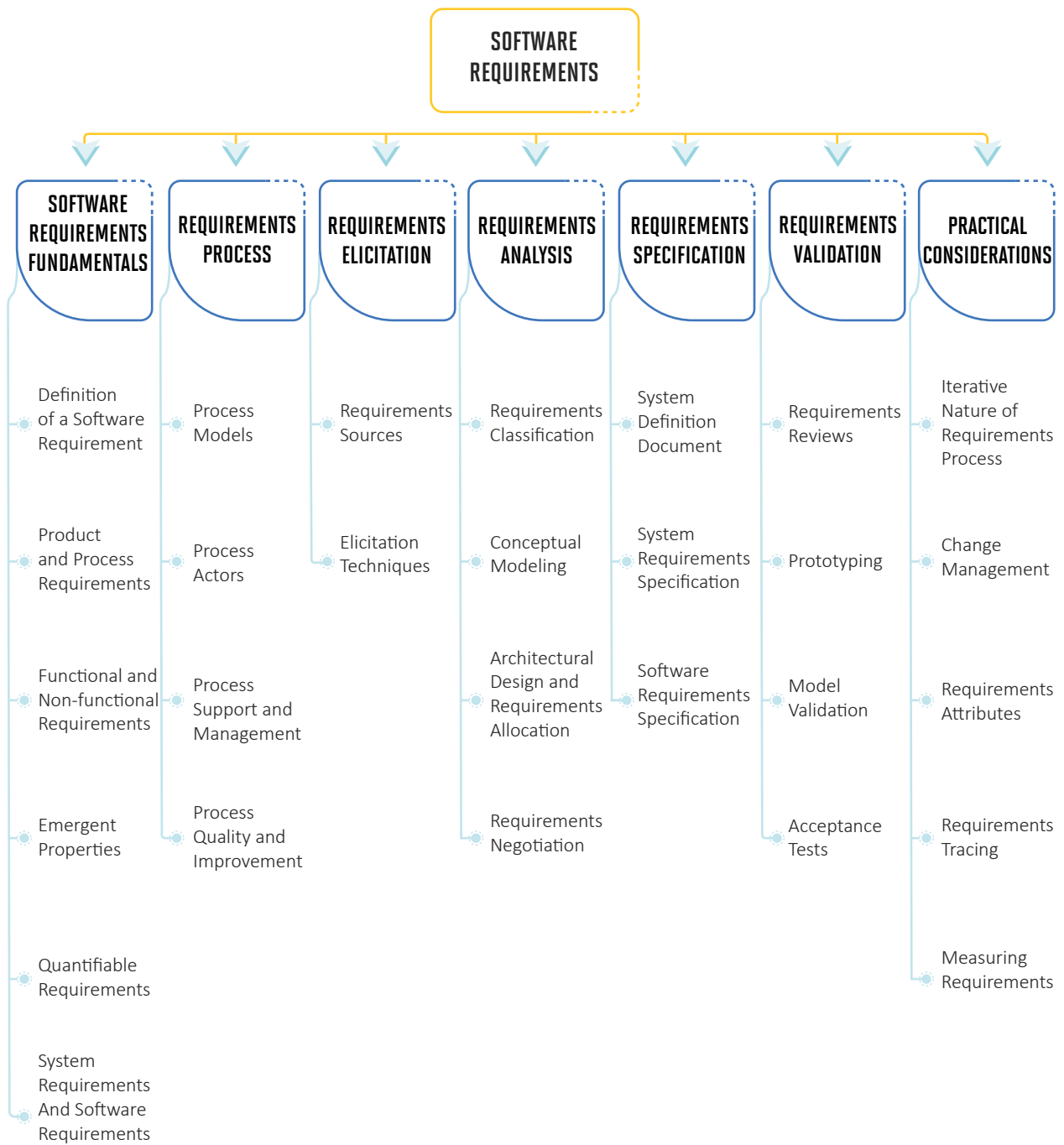
1. OBJETIVO DE APRENDIZAJE



2. INTRODUCCIÓN

2.1 SWEBOK

El Software Engineering Body of Knowledge (SWEBOK) nos presenta las áreas de conocimiento asociadas a Software Requirements (Requisitos de Software). Sin embargo, estas áreas no están circunscritas a un approach o enfoque en particular. En las siguientes secciones revisaremos diversos approaches que influyen en la manera como se lleva a cabo el ciclo de vida de software.





3. AGILE

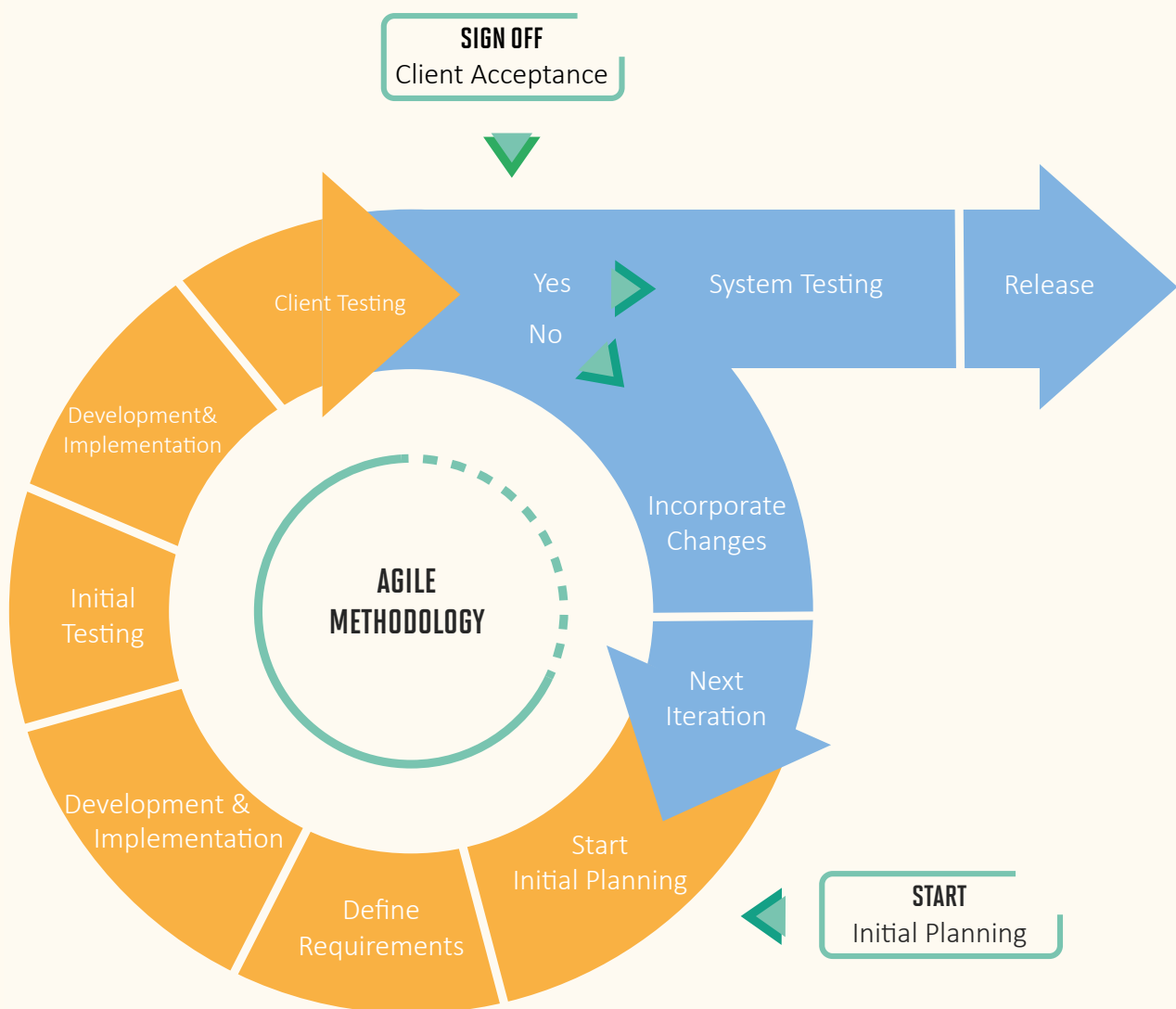
En marzo de 2001, Kent Beck junto con un grupo de críticos de sistemas de producción se reunieron en Salt Lake City para discutir sobre el desarrollo de software. En dicha reunión se acuñó el término *Métodos ágiles*, como alternativa para los métodos y metodologías formales de la época. El Manifiesto ágil es producto de ese grupo. En base a una serie de postulados en los que se fundamenta, el manifiesto incluye además 12 principios, que presentamos a continuación.

3.1 12 AGILE PRINCIPLES



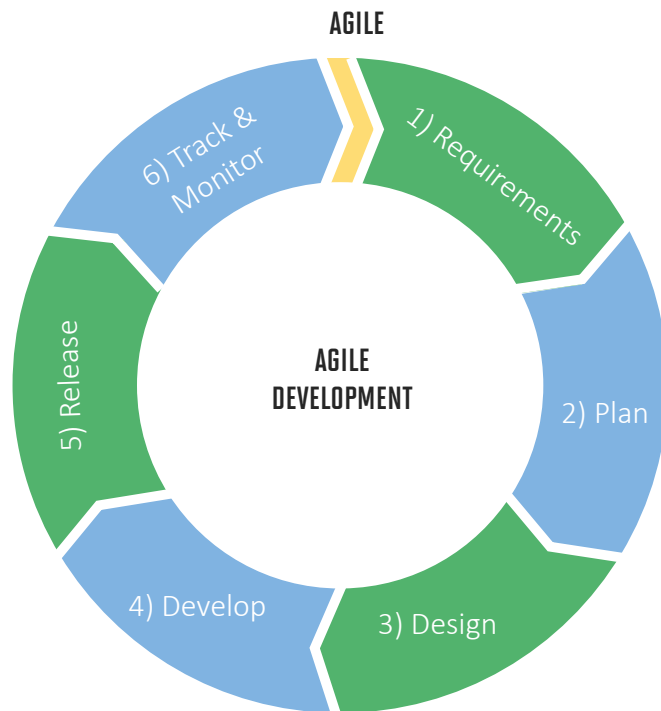


El proceso agile sigue un enfoque iterativo e incremental; es decir, que las actividades del ciclo de vida de software se repiten en cada ciclo o iteración. Esto incluye la definición de requisitos, la cual se va refinando y expandiendo si es necesario.

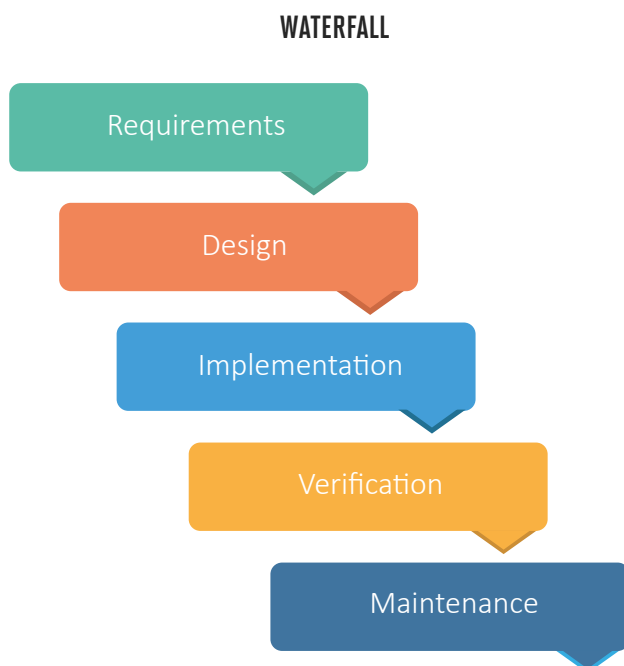




Aquí apreciamos claramente la diferencia entre el proceso ágil y el proceso Waterfall (en cascada). Observamos que el primero está más orientado a abrazar el cambio como un hecho durante el proceso, dentro del alcance del proyecto; mientras que, la esencia del proceso Waterfall parte de la premisa que un entorno donde, luego del contrato inicial, los cambios se tratan de controlar o postergar.



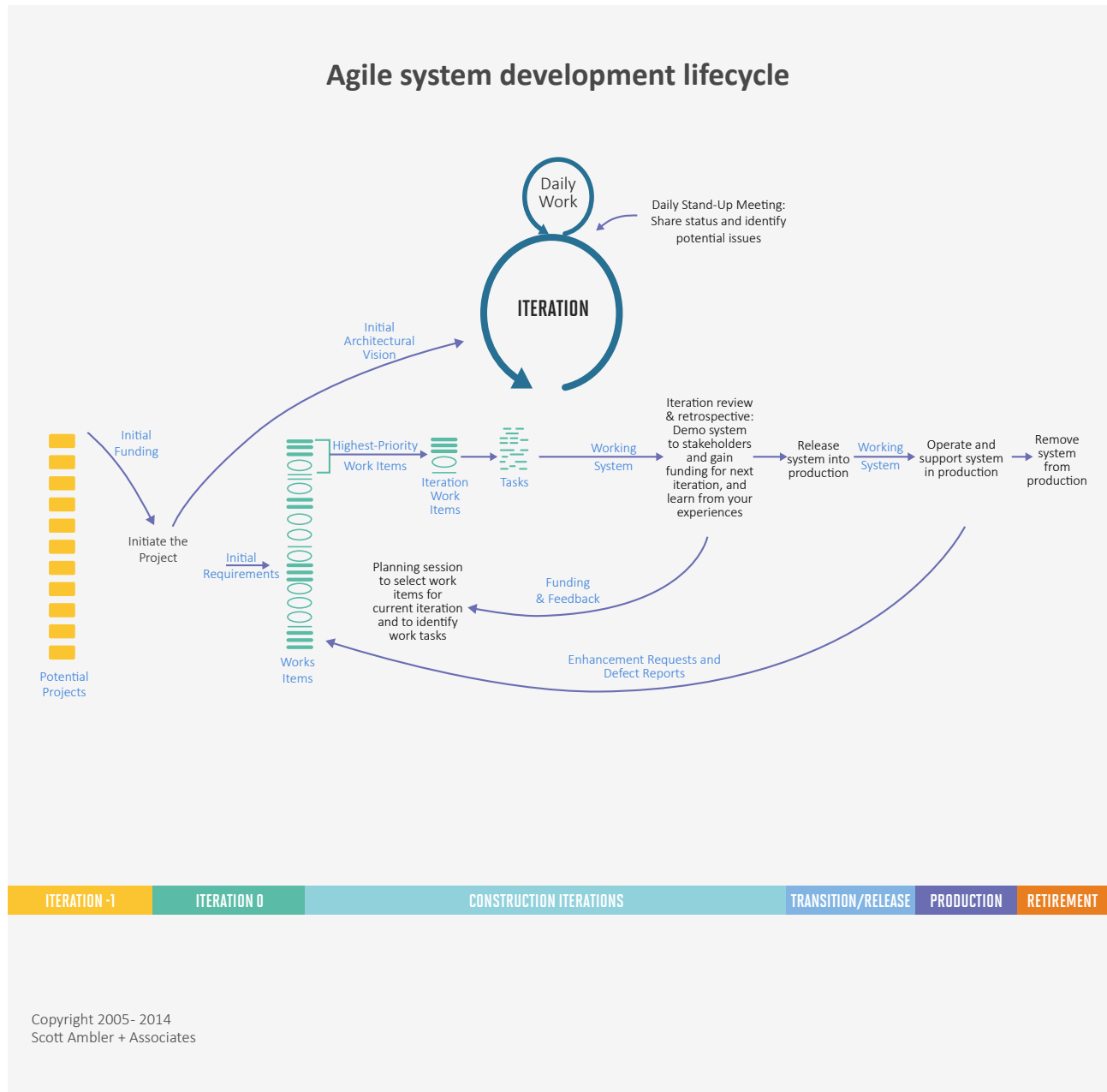
- Continuous cycles
- Small, high-functioning, collaborative teams
- Multiple methodologies
- Flexible/continuous evolution
- Customer involvement



- Sequential/linear stages
- Contract negotiation
- Best for simple, unchanging projects
- Close project manager involvement
- Upfront planning and in-depth documentation



Este gráfico resume bastante bien las características del ciclo de vida ágil para el desarrollo de software. Luego del establecimiento de requisitos iniciales, cada iteración brinda la posibilidad de refinarlos, incorporar nuevos requisitos o modificar la prioridad de requisitos.

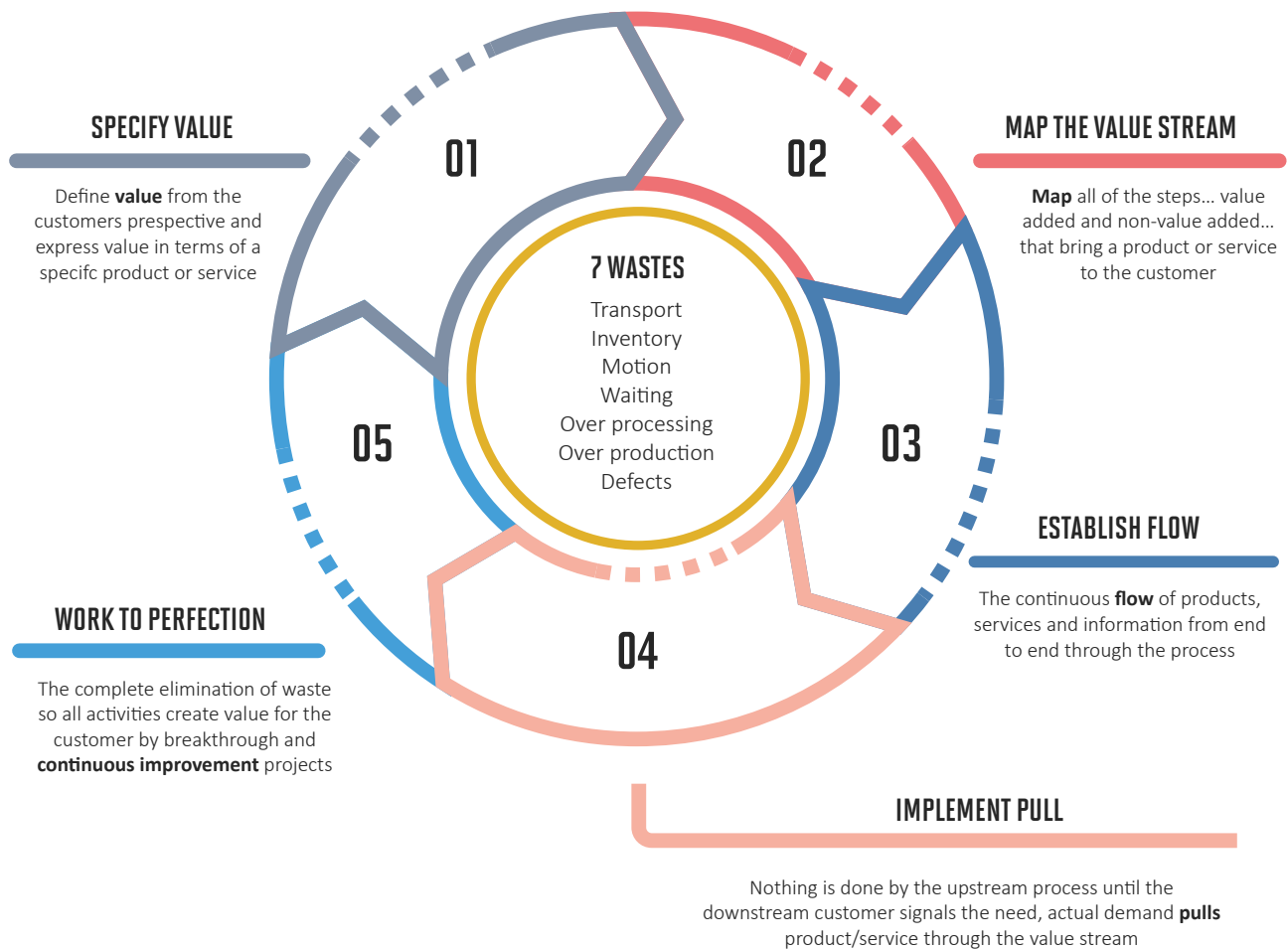


4. LEAN

El approach de Lean se basa en una idea central: maximizar el valor de cara al cliente y minimizar el desperdicio. Dicho de otra manera, hablamos de crear más valor con menos recursos.



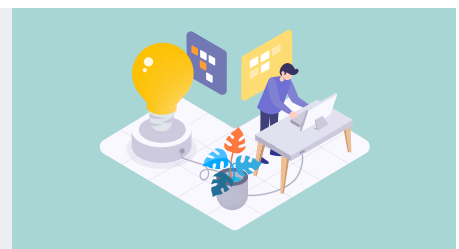
4.1 LEAN PRINCIPLES



Ver más en <https://www.lean.org/WhatsLean/Principles.cfm>

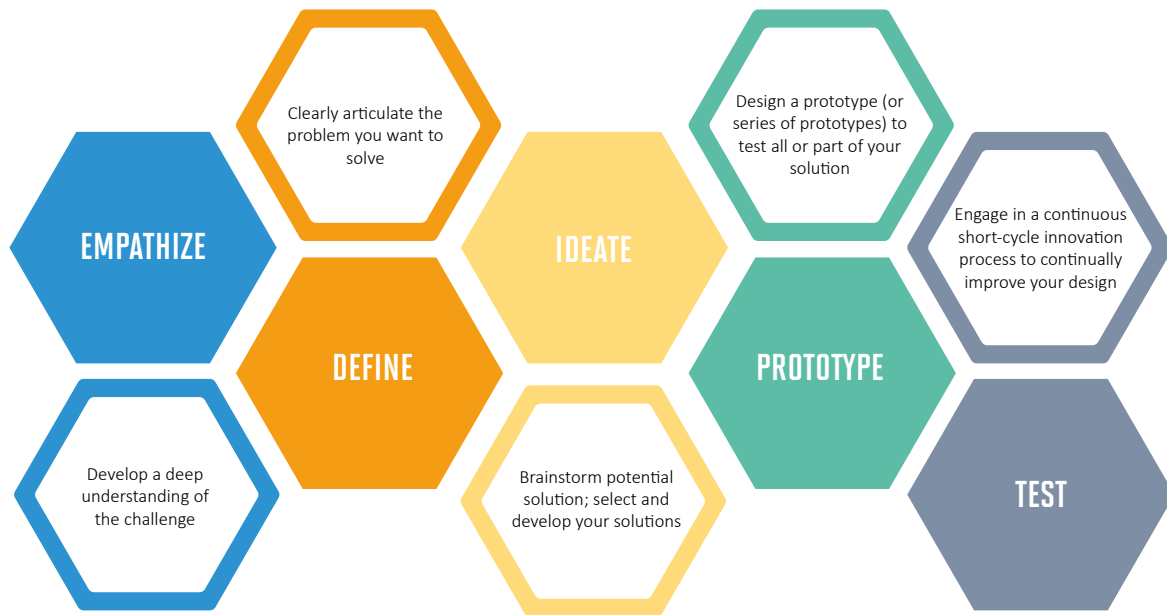
5. DESIGN THINKING

Esta filosofía considera tan importante el análisis y el raciocinio como la intuición para resolución de problemas. Ésta gira en torno a la experiencia de usuario. Para este análisis se integran equipos multidisciplinares, que generen tantas ideas como sea posible.



Finalmente, el objetivo del proceso es aprender a partir de las reacciones del usuario al interactuar con el producto.

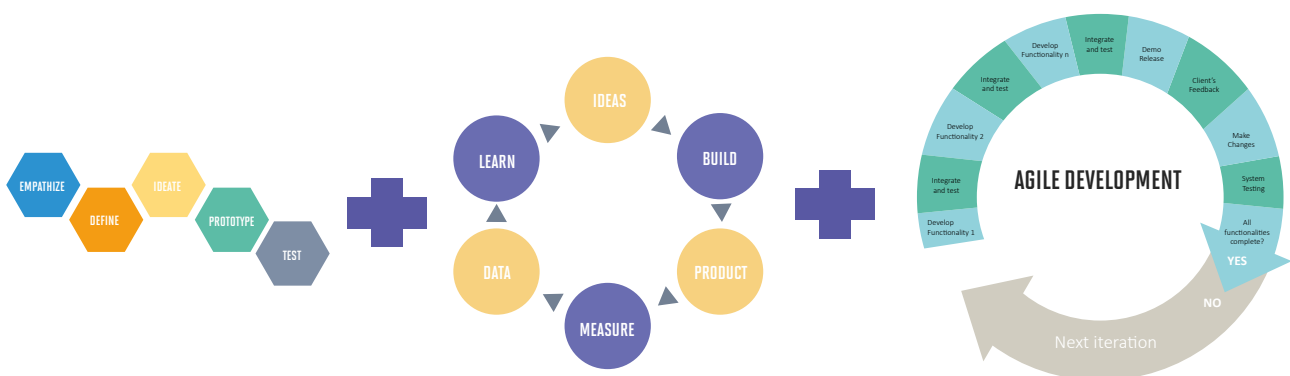
Ver más en <https://designthinking.ideo.com/>



5.1 DESIGN THINKING & LEAN STARTUP & AGILE

Muchos entornos buscan establecer un approach personalizado, resultado de la combinación de Design Thinking, Lean y Agile.

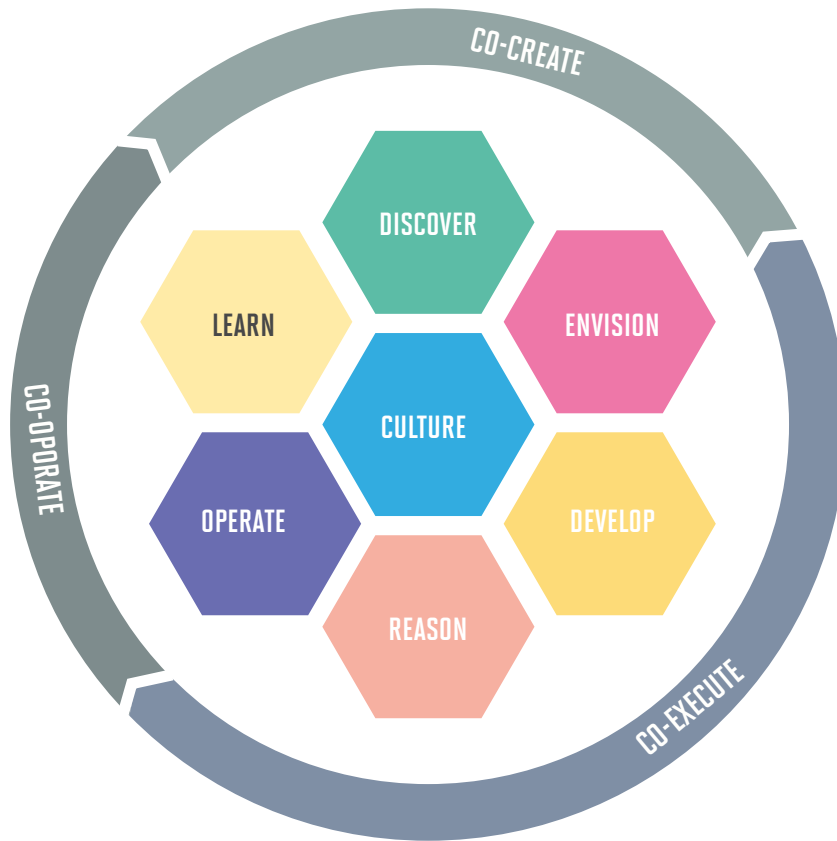
- Design Thinking tiene que ver con cómo exploramos y resolvemos problemas.
- Lean es el framework para poner a prueba nuestras creencias y aprender el camino hacia los logros que requerimos.
- Agile se enfoca en cómo nos adaptamos a las condiciones cambiantes del software.



6. IBM GARAGE METHOD

Es un modelo end-to-end para la transformación digital. Busca ayudar a generar ideas innovadoras y proporciona prácticas, tecnologías y experiencia que haga falta para convertirlas en valor empresarial.

Ver más en <https://www.ibm.com/garage>

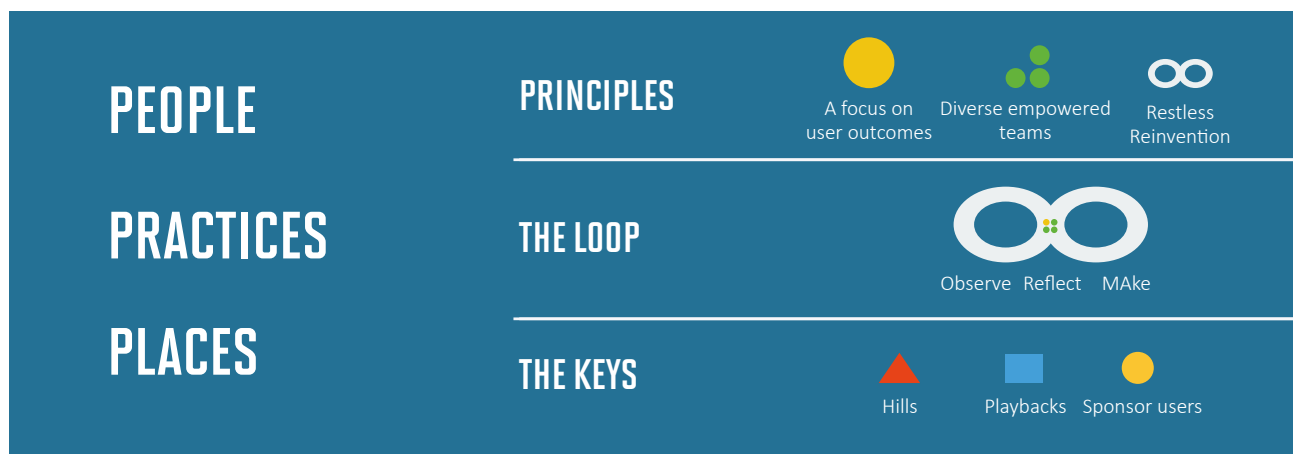


6.1 ENTERPRISE DESIGN THINKING

Es el pensamiento de diseño reinventado para la empresa moderna, constituyéndose en un marco para resolver problemas de los usuarios. El marco se basa en tres grandes ejes: los principios, the loop y las llaves. En lugar de definir un proyecto por sus componentes estructurales o requisitos técnicos, Design Thinking se centra en averiguar cómo una idea puede ayudar a un usuario final a lograr objetivos específicos.



Ver más en <https://www.ibm.com/design/thinking/page/framework>





7. CONCLUSIONES



- El proceso Agile tiene un enfoque iterativo e incremental que concibe el cambio como un hecho durante cada ciclo, lo cual da la posibilidad de modificar los requisitos.
- El approach Lean busca crear el mayor valor posible con la menor cantidad de recursos, minimizando el desperdicio.
- El Design Thinking se centra en la elaboración de ideas y en la experiencia del usuario para la resolución de problemas y lograr los objetivos específicos.

REFERENCIAS

Para profundizar:

- <https://www.computer.org/education/bodies-of-knowledge/software-engineering>
- <https://www.iiba.org/standards-and-resources/babok/>
- <https://melsatar.blog/2012/03/15/software-development-life-cycle-models-and-methodologies/>



© UPC. Todos los derechos reservados

Autor: Ángel Augusto Vasquez Nuñez