# Planteamiento de la construcción del modulo

Objetivos: Establecer la metodología a seguir para el desarrollo del módulo de consultas federadas en Apache Marmotta.

Resultados esperados: Documento que explique qué metodología se seguirá para el desarrollo del modulo

El desarrollo del módulo de consultas federadas para datos geoespaciales debe de basarse en alguna metodología que establezca un ciclo de vida del software. Alfonso Fuggetta, basado en el libro *The cathedral and the Bazaar* [1] de Eric Raymon, propone en su *paper: Open source software – an evaluation* [2] que las metodologías de desarrollo de software para tecnologías *open source* deben de ser de rápido prototipado, con desarrollo evolutivo e incremental. La propuestas de este tipo de metodologías de ciclo de vida de software por Fuggetta son

* Espiral
* Metodologías ágiles

Si bien las metodologías ágiles engloban a diversas metodologías, se discutirán 3.

A continuación, se describirán cada metodología y al final se dará un veredicto acerca de cuál metodología se usará con base al contexto del proyecto.

Método en espiral

Es un método iterativo representado por una espiral el cual permite llevar a cabo lanzamientos del software de manera incremental o refinación progresiva mediante la iteración de ciclos de la espiral. La espiral consta de 4 fases las cuales se deben de cursar para completar el ciclo y volver a empezar [3] (Fig. 1)

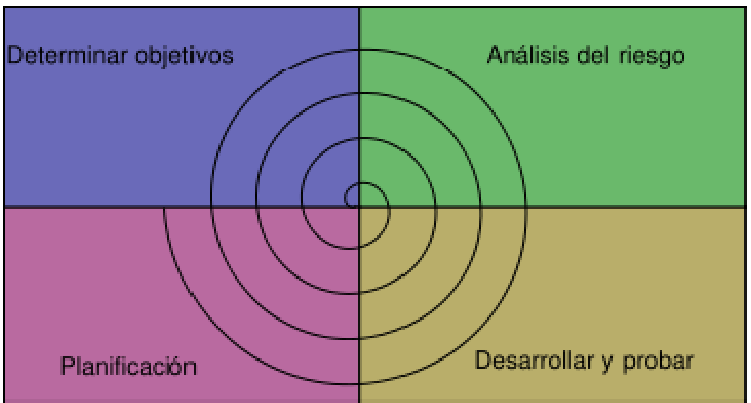


Fig. 1 Diagrama método espiral

Las fases son

* Determinar objetivos:
  + Establecer, en conjunto con el cliente, los objetivos para la fase actual.
  + Determinar restricciones.
  + Identificar riesgos presentes en el proyecto a su vez de contemplar métodos para sortearlos.
  + Acordar qué productos se deben obtener tales como documentación, especificaciones o requerimientos del sistema.
* Análisis de riesgo:
  + Identificar y evaluar riesgos para con el fin de llevar a cabo acciones que disminuyan riesgos significativos. Las acciones pueden ser simulaciones o prototipos del sistema.
  + Analizar riesgos potenciales además de propuesta y selección de alternativas para eliminar o disminuirlos.
* Desarrollo y prueba:
  + Desarrollar el software basándose en los objetivos acordados para la actual fase y los riesgos previamente identificados.
  + Se prueba el software en un entorno de producción para validar su funcionamiento.
* Planificación:
  + Evaluar los productos hechos para organizar como se llevará a cabo la siguiente fase.
  + Detallar qué problemas surgieron en la actual fase para darles solución en la siguiente.
  + Comprobar que los problemas que se presentaron en la anterior fase se solucionaron para corregirlos o considerar si una alternativa es mejor solución al implementarse en la siguiente fase.

Ventajas y desventajas

* Ventajas
  + Los cambios que requiera el cliente pueden ser implementados a temprana etapa del desarrollo del software.
  + Disminución de dificultades al contemplar y tomar acciones contra los riesgos que se presentan.
  + Reutilización de software existente e implementación de funcionalidades de forma progresiva.
  + Se puede incorporar objetivos de calidad de software dentro de su desarrollo.
  + Los requerimientos del sistema pueden ser atendidos de manera precisa.
  + El cliente puede ver resultados del software desde temprana etapa del proyecto.
* Desventajas
  + El tiempo del desarrollo es ambiguo.
  + Si no se realiza un análisis de riesgo adecuado podría afectar negativamente al proyecto por completo.
  + La administración y el proceso del proyecto es más complejo.
  + No es apto para proyectos que involucren riesgos pequeños puesto que podría ser costoso.