

# Qué es un programador ágil

Pepe Doval

SetPay — Agile Galicia

28 de marzo de 2017



Esta obra está suxeita á licencia Recoñecemento-Compartirlgual 4.0 Internacional de Creative Commons. Para ver unha copia desta licencia, visite <http://creativecommons.org/licenses/by-sa/4.0/>.

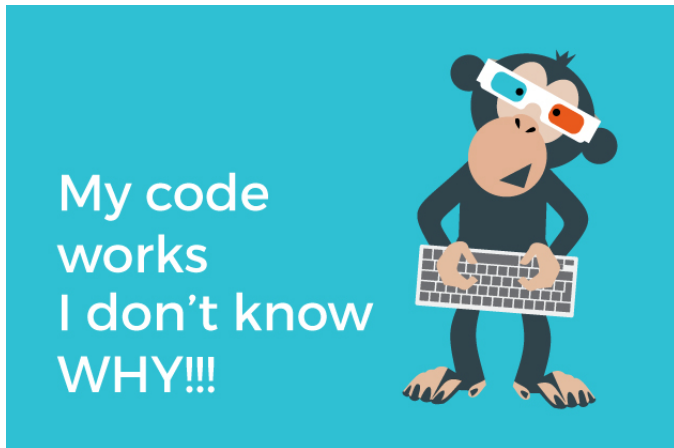
# Agile

# El programador ágil



**Figura:** <https://www.apico.net/blog/how-bad-software-developer-are-you.html>

# El programador frágil



**Figura:** <https://www.apico.net/blog/how-bad-software-developer-are-you.html>

# El plan



**Figura:** <https://pixabay.com/p-618371/>

# Cuando los planes salen bien



# Metáfora del cañón



**Figura:** <http://maxpixel.freegreatpicture.com/Cannonballs-Weapon-Cannon-Cannonball-History-War-314498>

# Asunciones del programador frágil

- El cliente **sabe** lo que quiere
- El programador(equipo) **sabe** cómo construirlo
- **No** ocurren cambios durante el desarrollo



# Asunciones del programador ágil

- El cliente ~~sabe~~ **descubre** lo que quiere
- El programador(equipo) ~~sabe~~ **descubre** cómo construirlo
- ~~No~~ ocurren cambios durante el desarrollo

# Manifiesto



**Manifiesto por el Desarrollo Ágil de Software**

Estamos descubriendo formas mejores de desarrollar software tanto por nuestra propia experiencia como ayudando a terceros. A través de este trabajo hemos aprendido a valorar:

- Individuos e interacciones sobre procesos y herramientas
- Software funcionando sobre documentación extensiva
- Colaboración con el cliente sobre negociación contractual
- Respuesta ante el cambio sobre seguir un plan

Esto es, aunque valoramos los elementos de la derecha, valoramos más los de la izquierda.

Kent Beck	James Grenning	Robert C. Martin
Mike Beedle	Jim Highsmith	Steve Mellor
Arie van Bennekum	Andrew Hunt	Ken Schwaber
Alistair Cockburn	Ron Jeffries	Jeff Sutherland
Ward Cunningham	Jon Kern	Dave Thomas
Martin Fowler	Brian Marick	

**Figura:** <http://agilemanifesto.org/>

# Prácticas

# Al programar (1/2)

- Pair programming
- TDD
- Refactorización
- Integración continua

# Al programar (2/2)

- Diseño simple
- Propiedad colectiva del código
- Estándares y convenciones

# Al planificar (1/2)

- Iteraciones
- Time boxing
- Planning game
- Standup meetings

# Al planificar (2/2)

- Automatización
- Radiadores de información
- Definitions of done

# Al aprender (1/2)

- Katas
- Coding Dojos
- Mob programming
- Desksurfing



# Al aprender (2/2)

- Patrones de aprendizaje
- Mentoring
- Artesanía del software

# El ingrediente secreto



**Figura:** <http://kungfupanda.wikia.com/>

# Agile

