

Los proyectos de software pueden salir bien...

**pregúnteme como**

[www.techbizdesign.com](http://www.techbizdesign.com)

# CIFRAS Y HECHOS

A black smartphone is positioned diagonally across the frame. The screen is lit up and shows the 'The Economist' logo, which consists of a red rectangle with the words 'The Economist' in white serif font. The phone is resting on a newspaper with blurred text in the background. The overall scene is set against a dark, semi-transparent vertical bar on the left side of the image.

The  
Economist

WYCIEK  
#PANAMA  
W LICZB



# 142.000 M€

**pierde cada año la economía Europea por proyectos de IT  
que no salen bien. EL 5% de su PIB...**

# 75%

**de los ejecutivos de negocio (y IT) que creen  
que los proyectos nacen condenados desde  
el principio**

Fuente: Geneca - [Up to 75% of Business and IT Executives Anticipate Their Software Projects Will Fail](#), 2011



# 1 de cada 6

proyectos de IT tiene un  
sobrecoste del 200% y un  
retraso del 70%



# 17%

de los proyectos van tan mal que ponen en  
riesgo la existencia de la compañía



**Pero entonces...**

**¿Cuántos proyectos salen mal?**

# Salen mal cuando:

Se entregan "evidentemente" tarde, por encima de presupuesto o con menos de las funciones requeridas



Se cancelan antes de acabarse o no se usa nunca una vez entregado

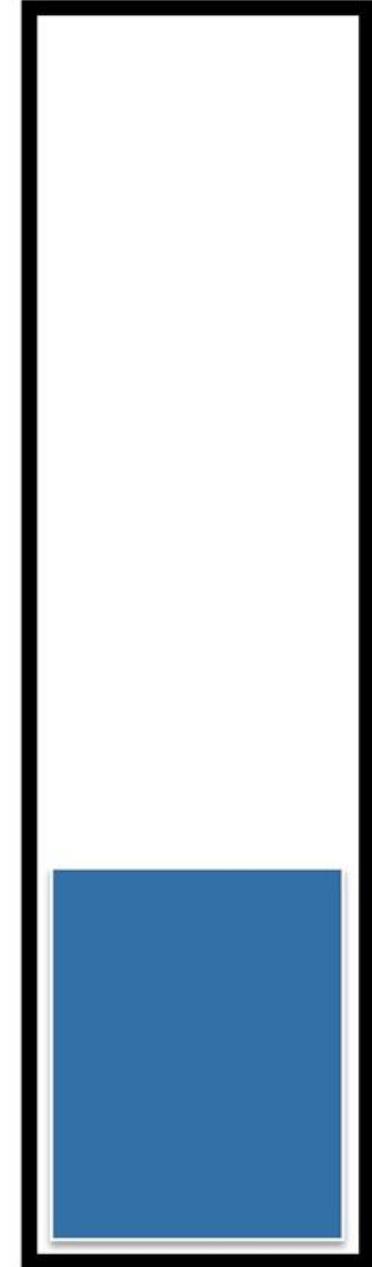


**Tomemos como ejemplo los proyectos  
de sistemas de atención al cliente o  
CRM...**



Advanced Market Research

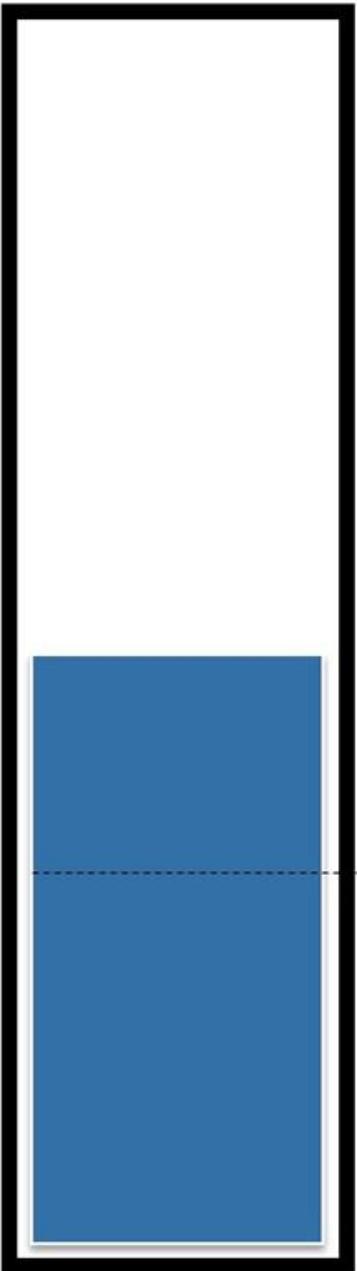
29%



Fuente: Virginia University – Curso Coursera “Agile meets Design Thinking”, 2016



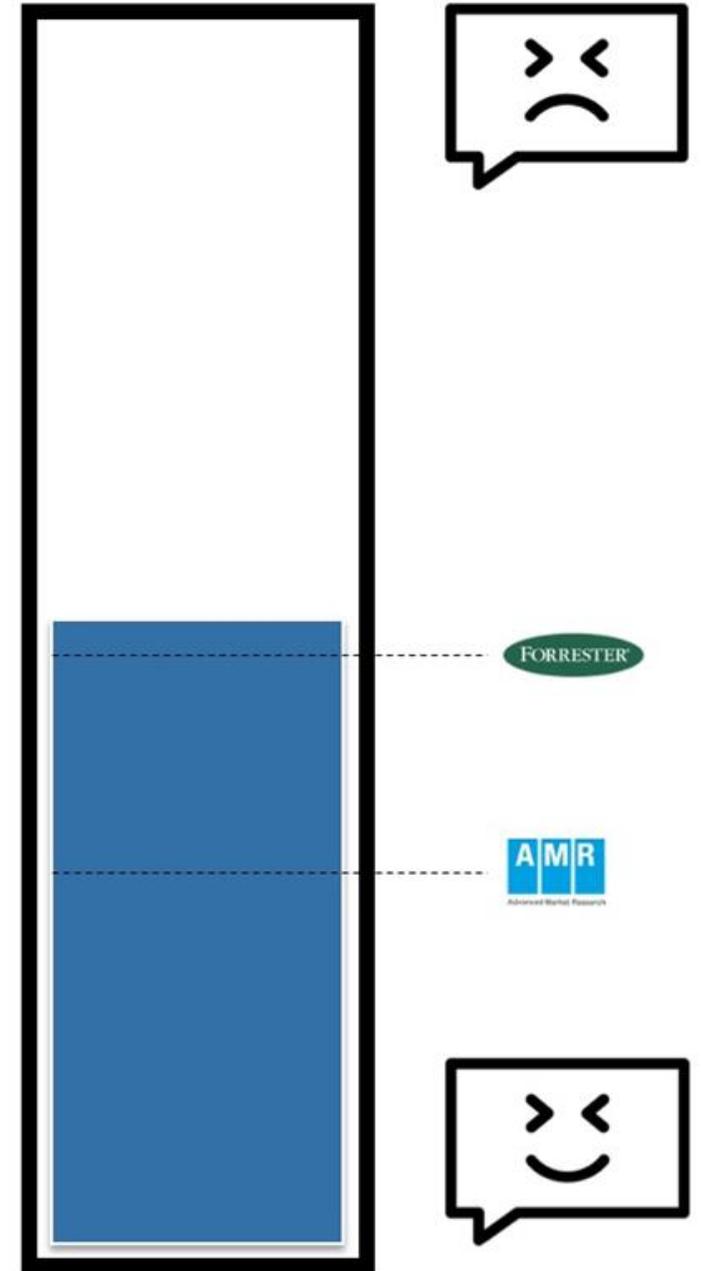
47%



Fuente: Virginia University – Curso Coursera “Agile meets Design Thinking”, 2016

Gartner®

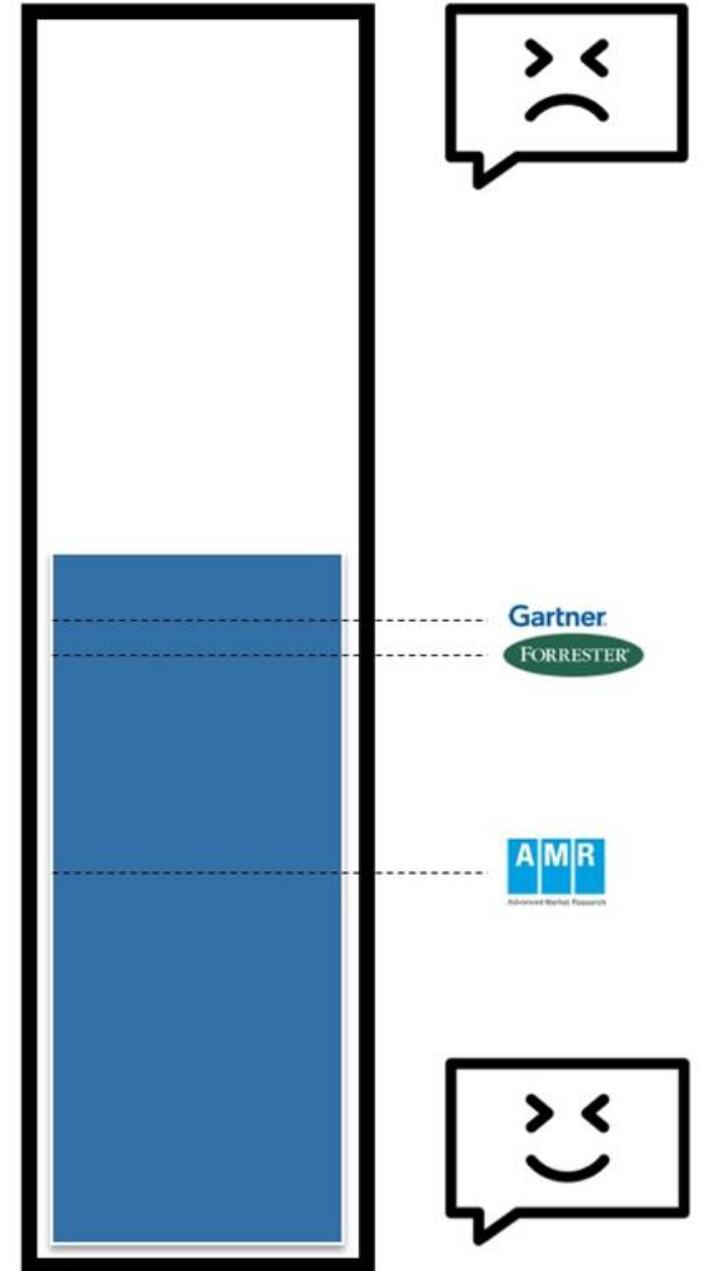
Más del **50%**



Fuente: Virginia University – Curso Coursera “Agile meets Design Thinking”, 2016

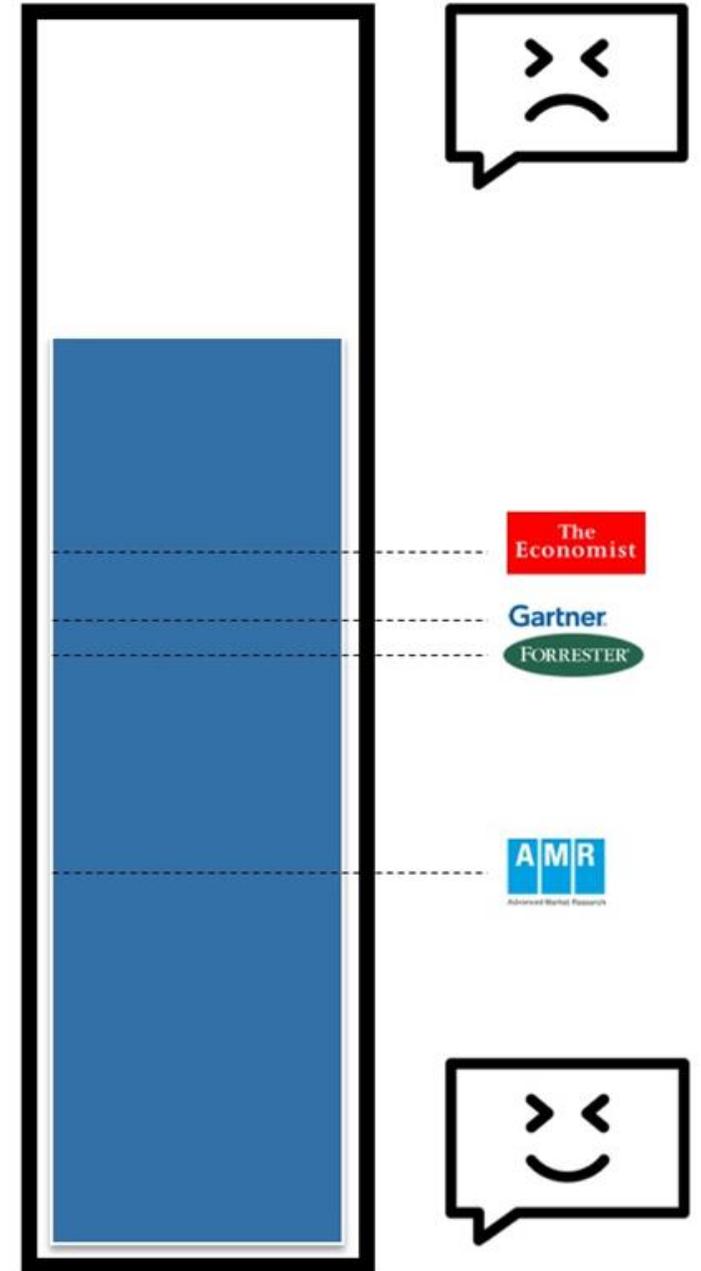
The  
Economist

56%



Fuente: Virginia University – Curso Coursera “Agile meets Design Thinking”, 2016

70%



**A nivel global...**

**66%**

**de los proyectos de software van por encima  
de presupuesto**



**33%**

**de los proyectos de software además van  
por encima de tiempo**



# PROBLEMAS Y PECADOS

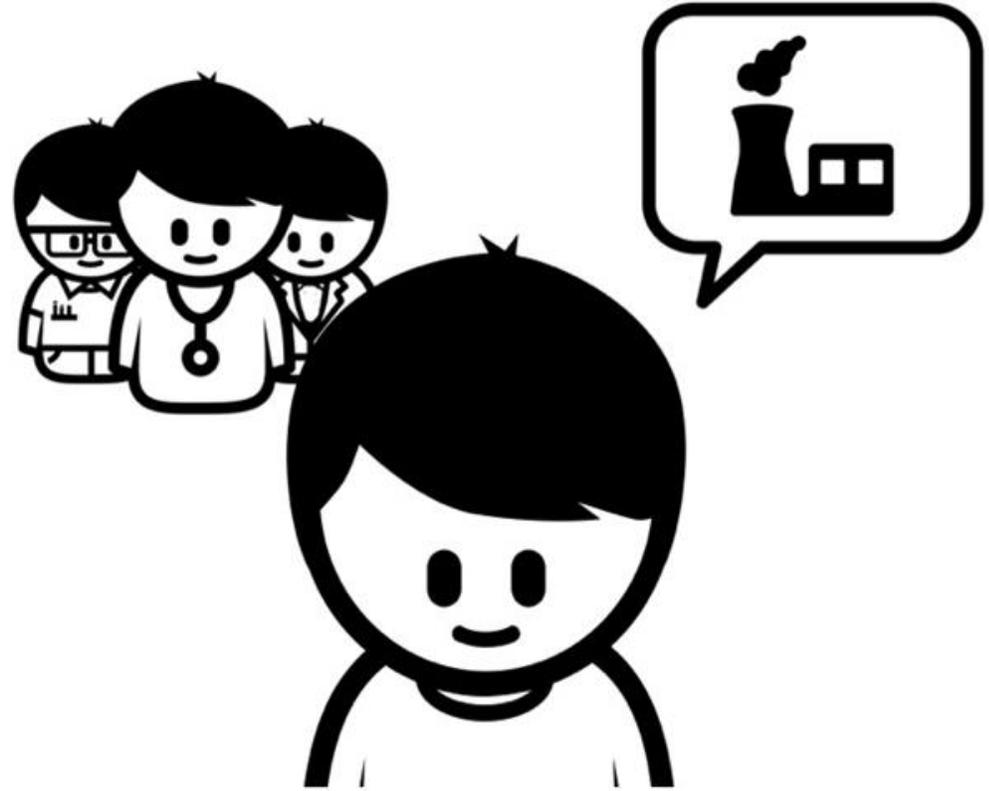


# Déjenme que les cuente una historia...



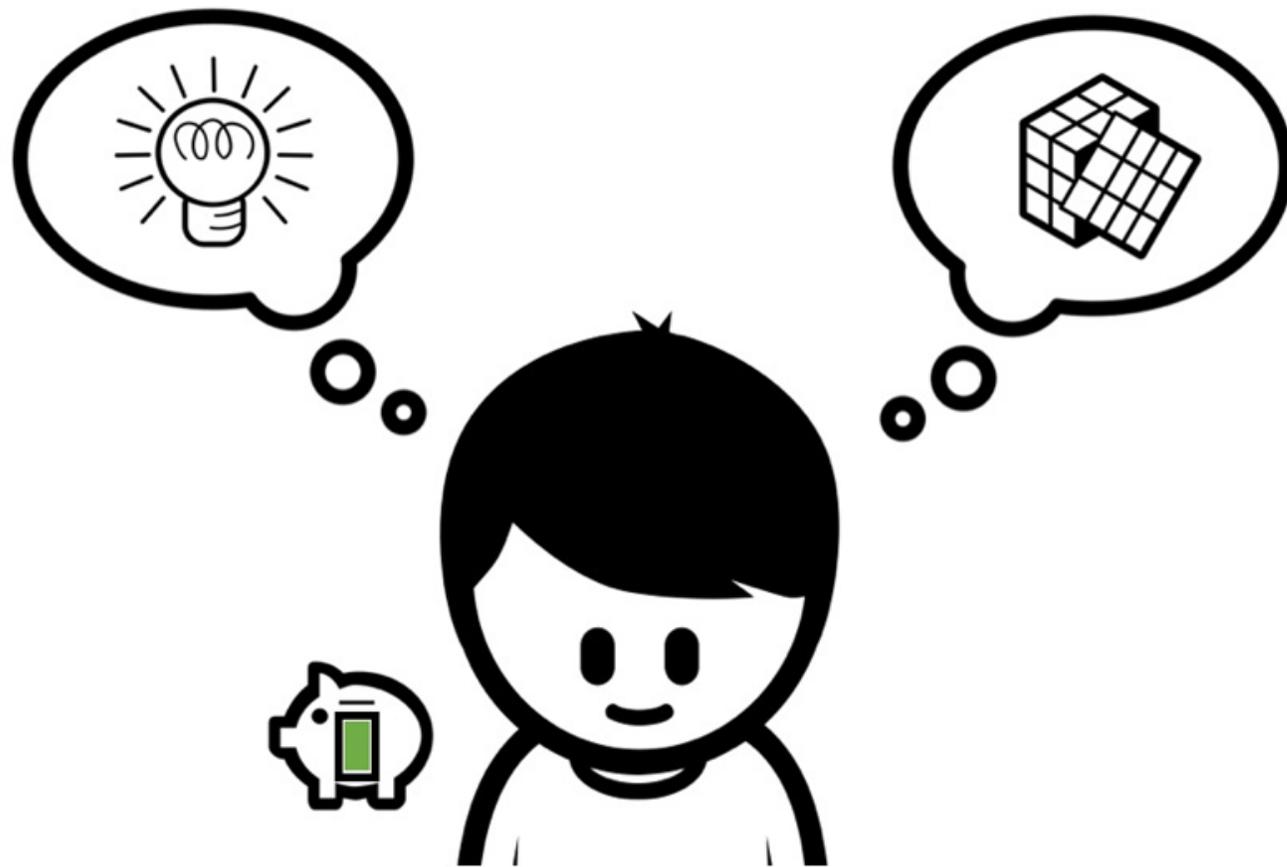
PEPE ES EL JEFE DE UN DEPARTAMENTO  
IMPORTANTE DENTRO DE UNA GRAN EMPRESA

DE ÉL DEPENDEN MUCHAS PERSONAS QUE LE  
REPORTAN DIARIAMENTE QUEJAS Y MEJORAS



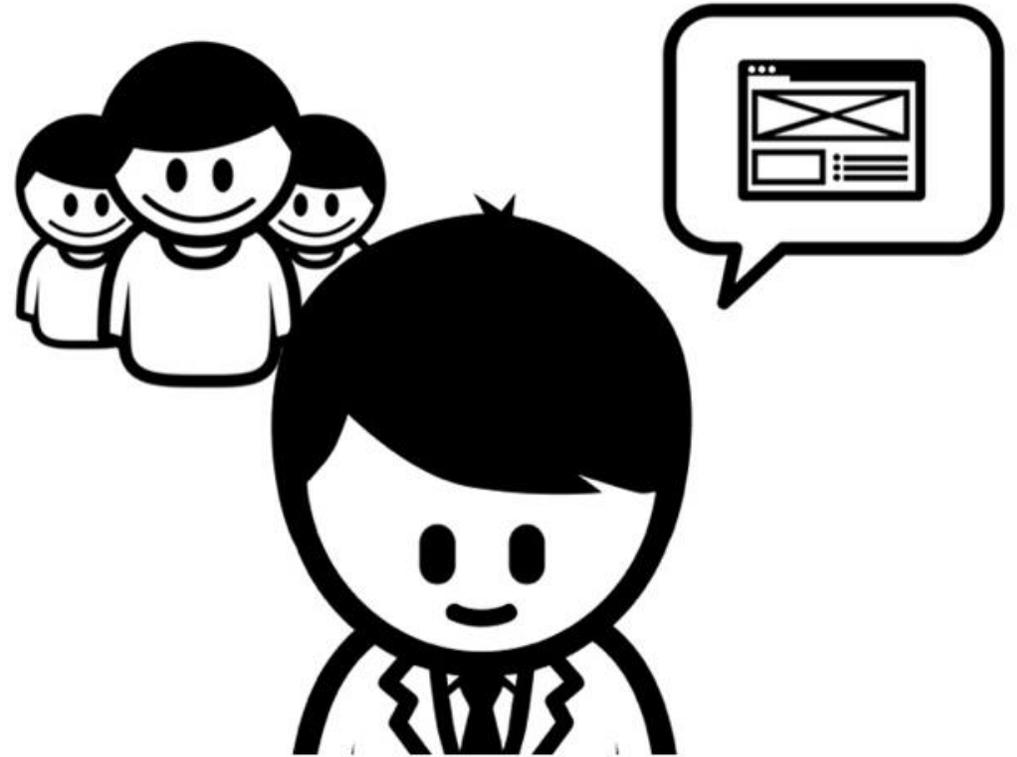
TIENE UNA IDEA QUE QUIERE LLEVAR A CABO  
O BIEN UN PROBLEMA QUE QUIERE  
SOLVENTAR

CREE QUE EL SOFTWARE PUEDE AYUDARLE Y  
PARA ELLO DESTINA UN DINERO...



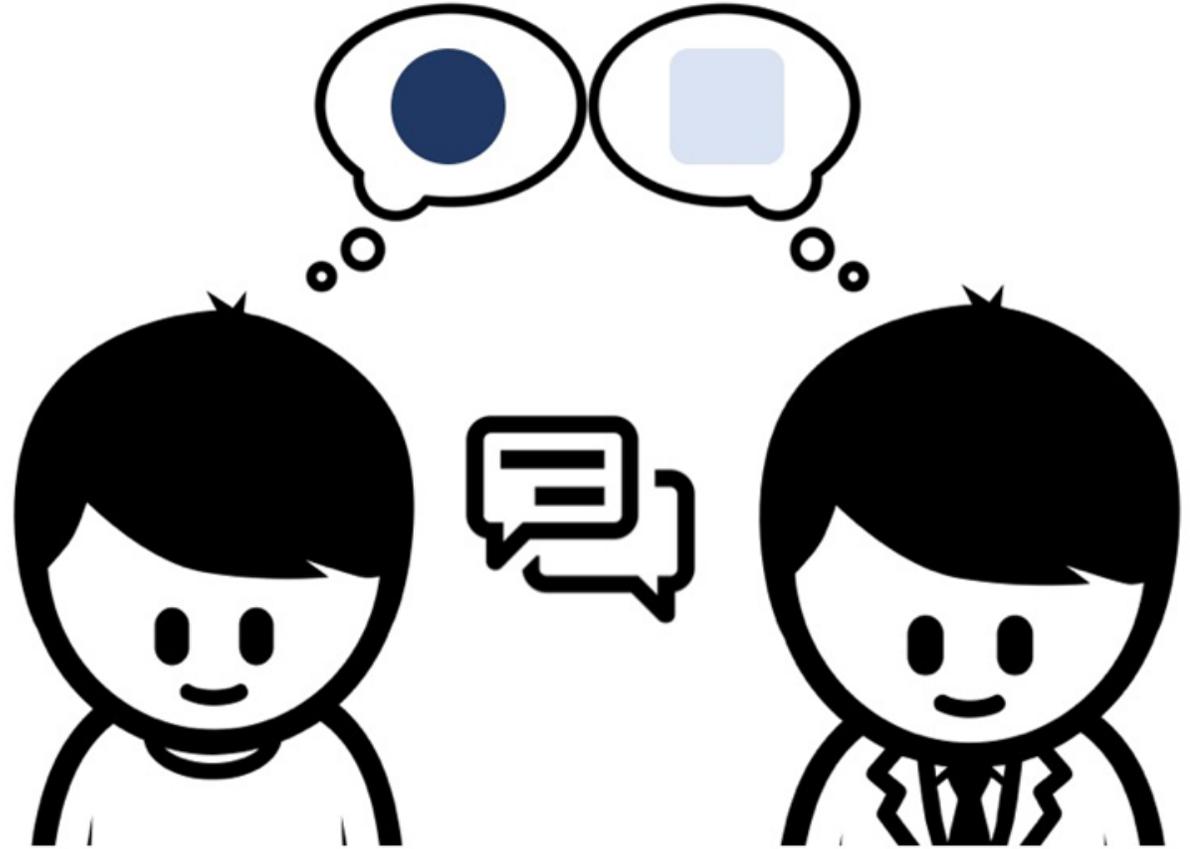
JUAN TIENE UNA EMPRESA CON MUCHOS  
EXPERTOS QUE SE DEDICA A HACER  
PROYECTOS DE SOFTWARE

ÉL CREE QUE PUEDE AYUDAR A PEPE.



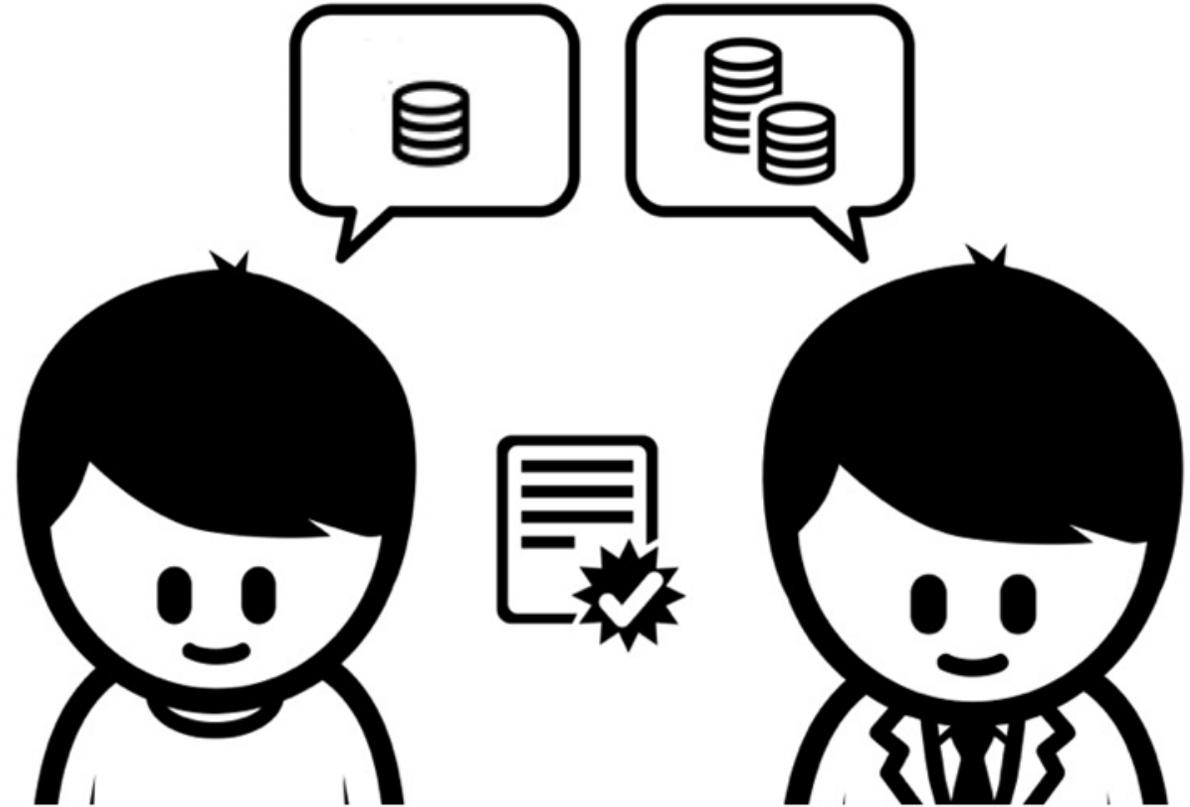
PEPE Y JUAN SE REUNEN...

PEPE EXPLICA LO QUE QUIERE Y JUAN  
LEVANTA LOS REQUISITOS DE LO QUE DEBE  
CONSTRUIR.



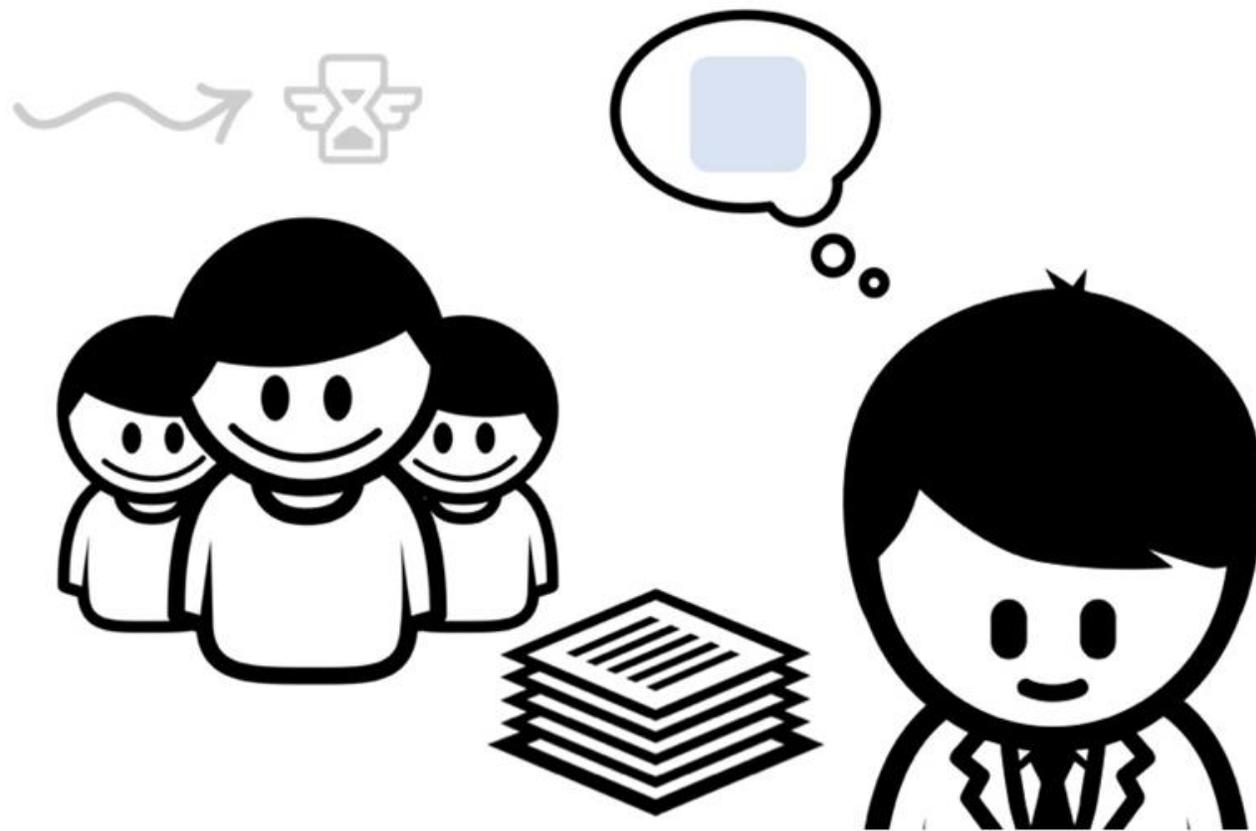
DESPUÉS DE MUCHO NEGOCIAR, SE  
PONEN DE ACUERDO.

JUAN PREPARA UNA PROPUESTA TÉCNICA  
Y ECONÓMICA DE LO QUE HA ENTENDIDO  
QUE NECESITA PEPE



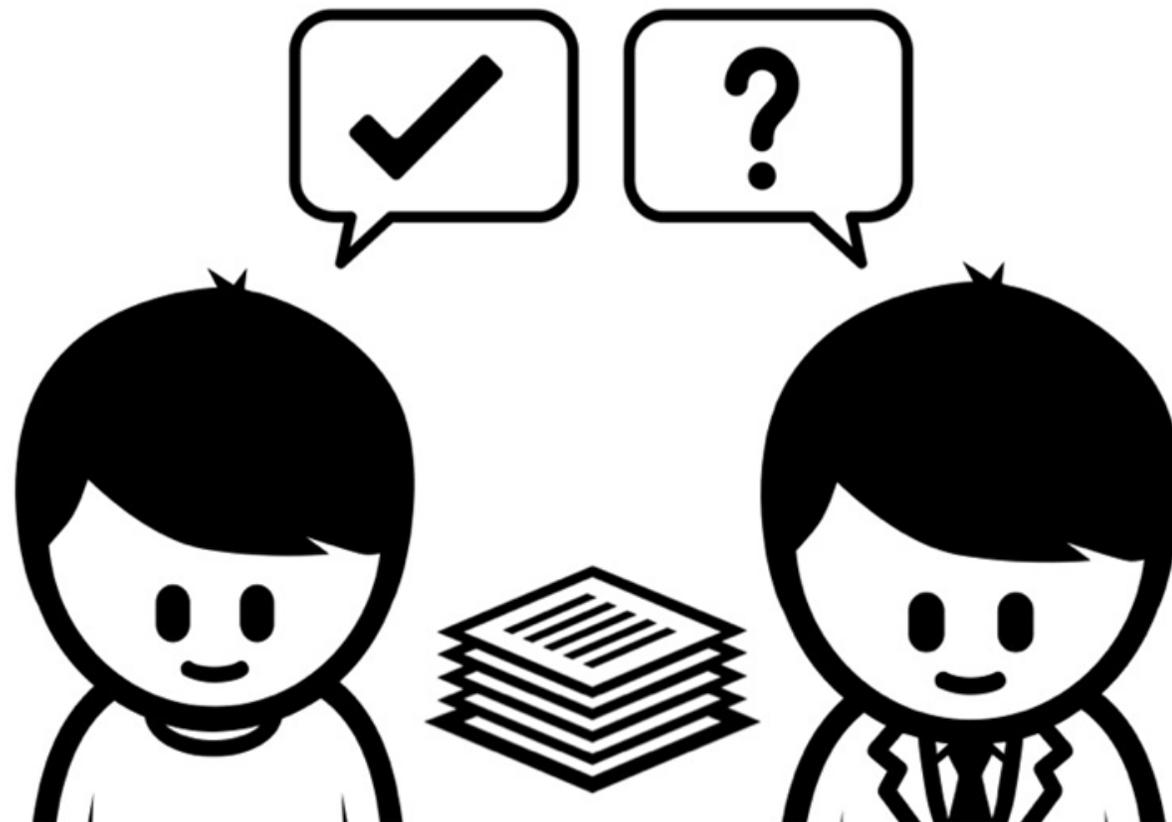
ANTES DE COMENZAR A CONSTRUIR NADA,  
EL EQUIPO DE JUAN ESCRIBE COMO VA A  
SER LO QUE VAN A CONSTRUIR

PASAN UN BUEN TIEMPO PARA HACERLO...



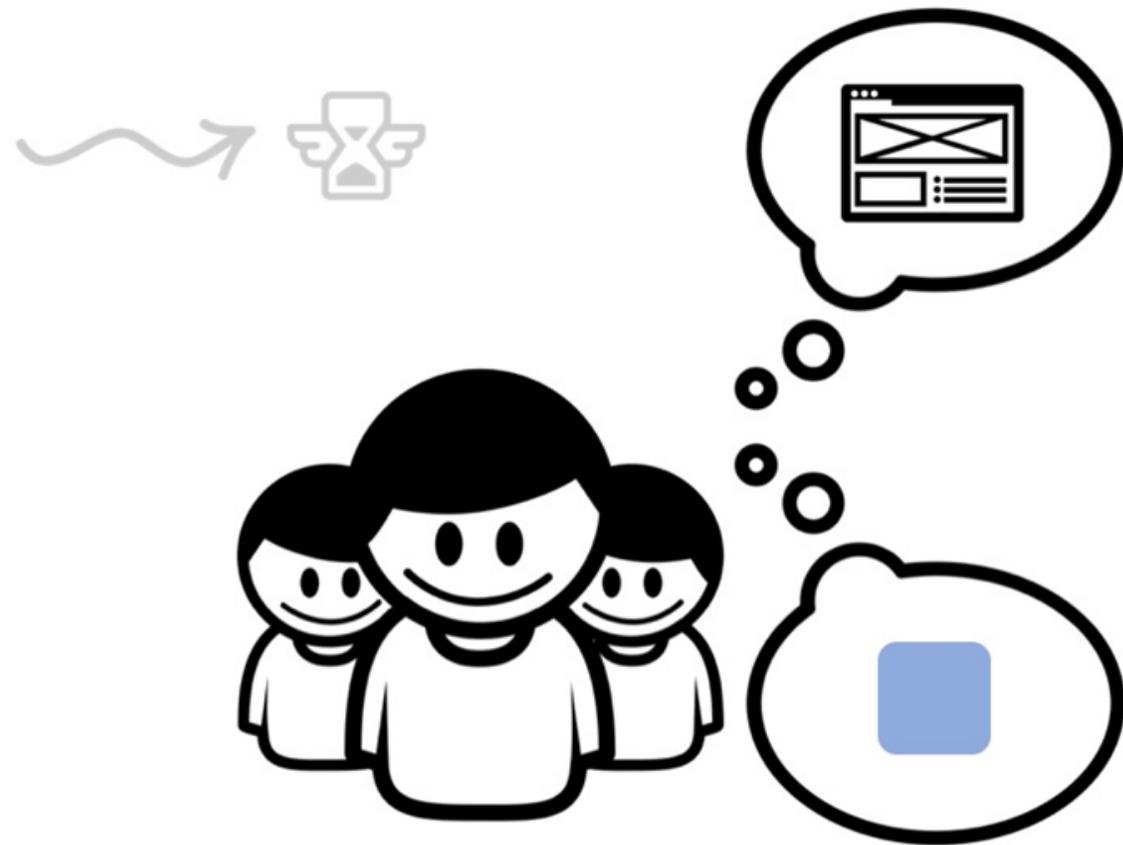
JUAN PIDE A PEPE QUE VALIDE LOS DOCUMENTOS QUE EXPLICAN COMO VA A SER LO QUE VA A CONSTRUIR

PEPE NO ENTIENDE LOS DOCUMENTOS Y SE FÍA DE JUAN QUE LE ASEGURA QUE ES LO QUE HA PEDIDO



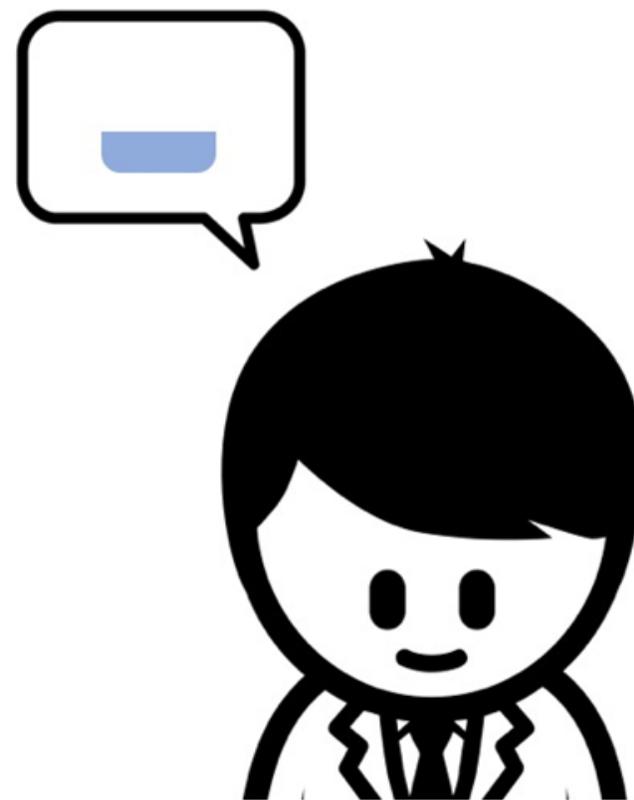
EL EQUIPO DE JUAN COMIENZA A  
CONSTRUIR

Y VAN PASANDO LOS MESES...



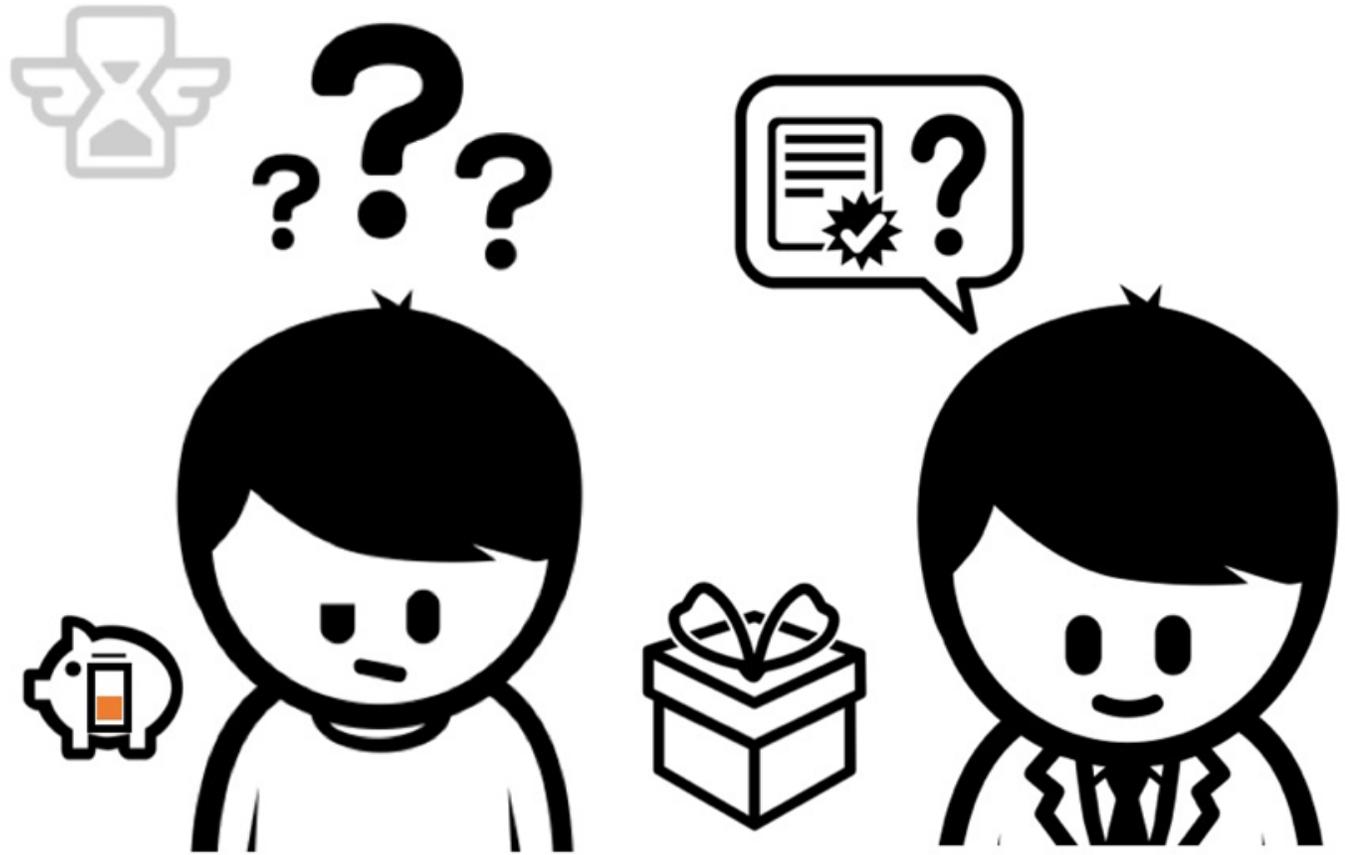
DESPUÉS DE MUCHO INSISTIR, PEPE  
CONSIGUE QUE LE ENSEÑEN COMO VA LA  
CONSTRUCCIÓN DE LO QUE QUIERE

NO ENTIENDE BIEN QUE LE MUESTRAN Y  
SIGUE CONFIANDO EN QUE LE  
ENTREGARAN LO QUE QUIERE POR EL  
DINERO QUE YA HA PAGADO

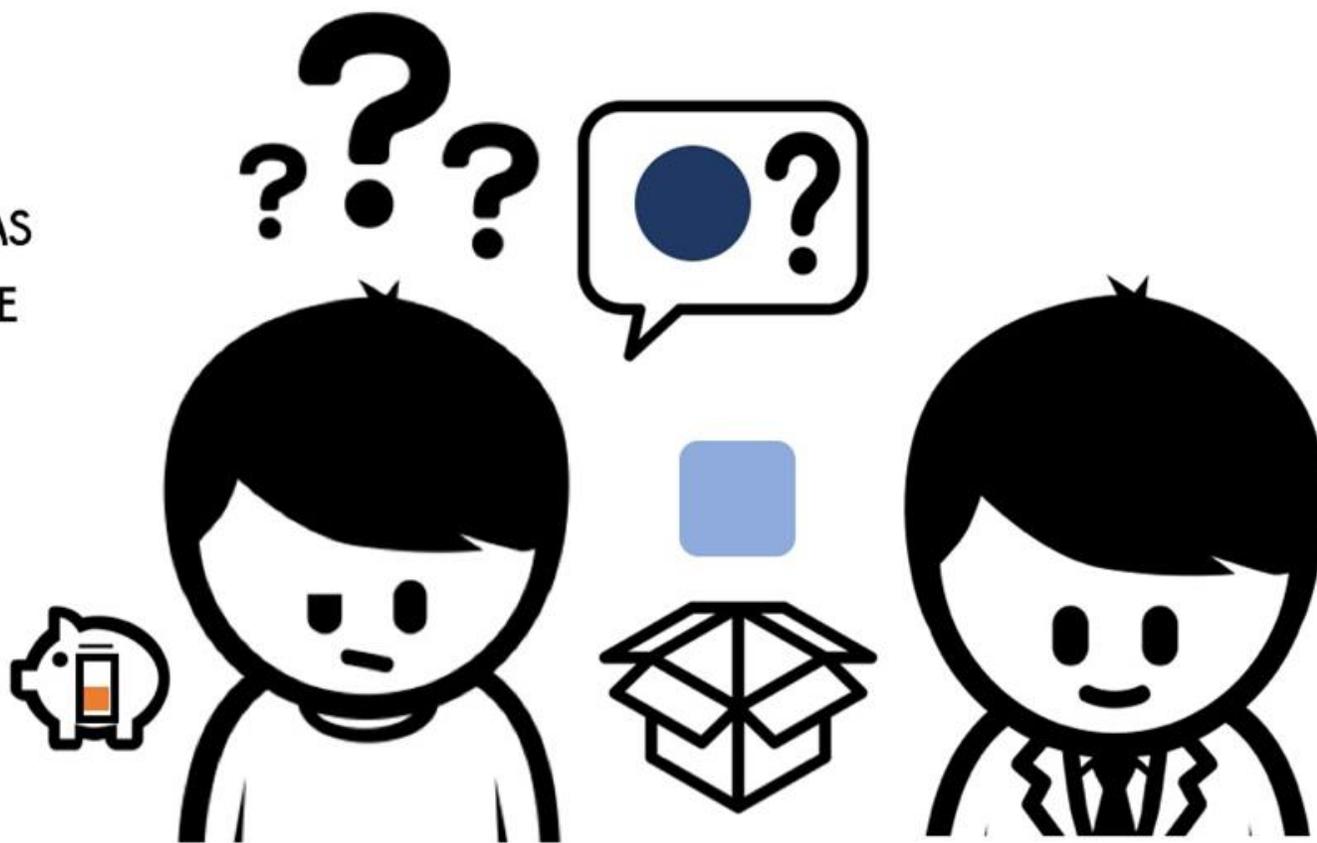


DESPUÉS DE MUCHO TIEMPO Y ALGÚN QUE OTRO RETRASO, JUAN ENTREGA A PEPE LO QUE HA CONSTRUIDO

ES HORA DE VALIDAR LO QUE HAN HECHO...

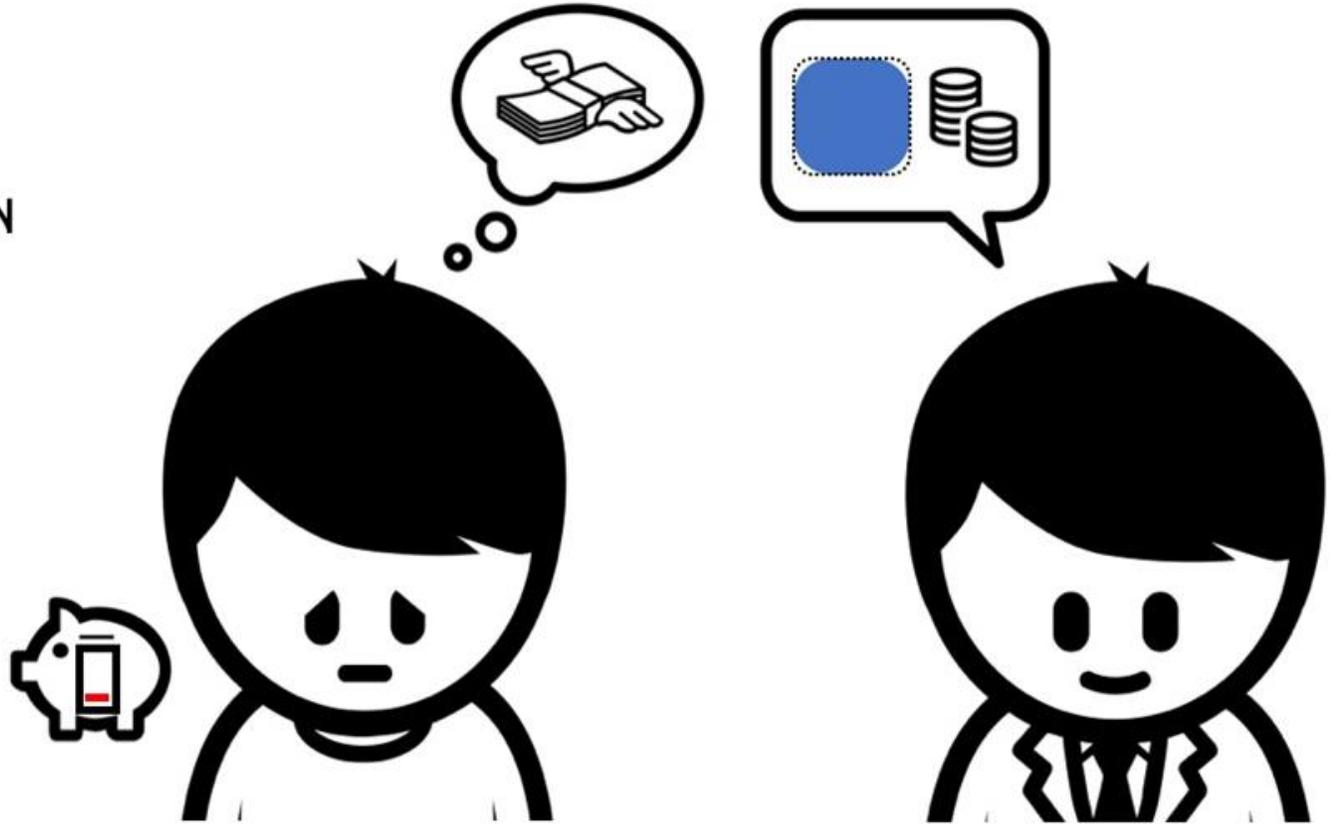


PEPE ENCUENTRA QUE HAY BASTANTES COSAS  
QUE NO SON COMO LAS HABÍA PENSADO Y LE  
PIDE A JUAN QUE LAS CAMBIE

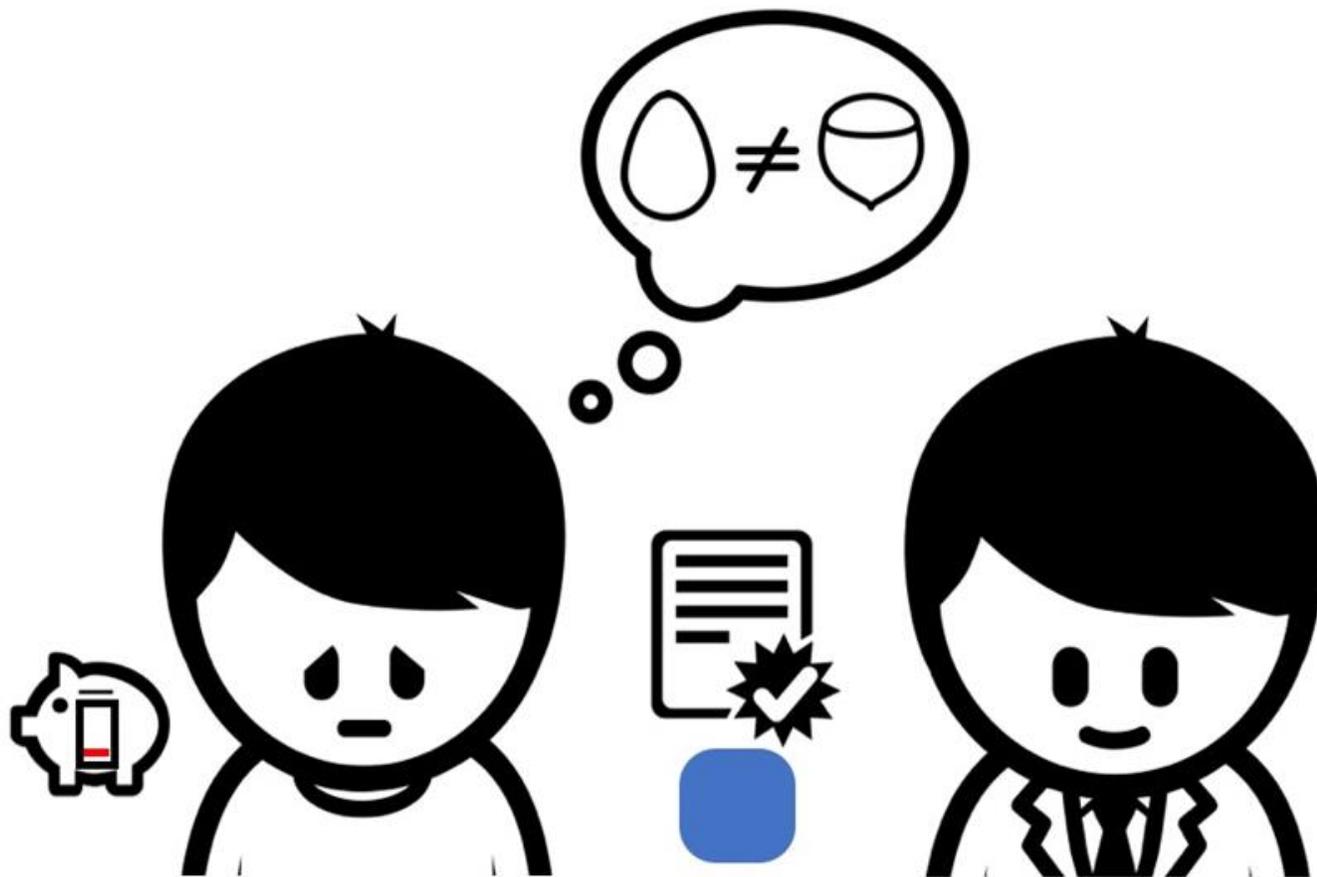


DESPUÉS DE CAMBIAR CUATRO COSAS, JUAN  
LE PIDE MÁS DINERO A PEPE PARA PODER  
HACER LO QUE EL PIDE

LE RECUERDA QUE EL VALIDÓ LAS  
"ESPECIFICACIONES"...

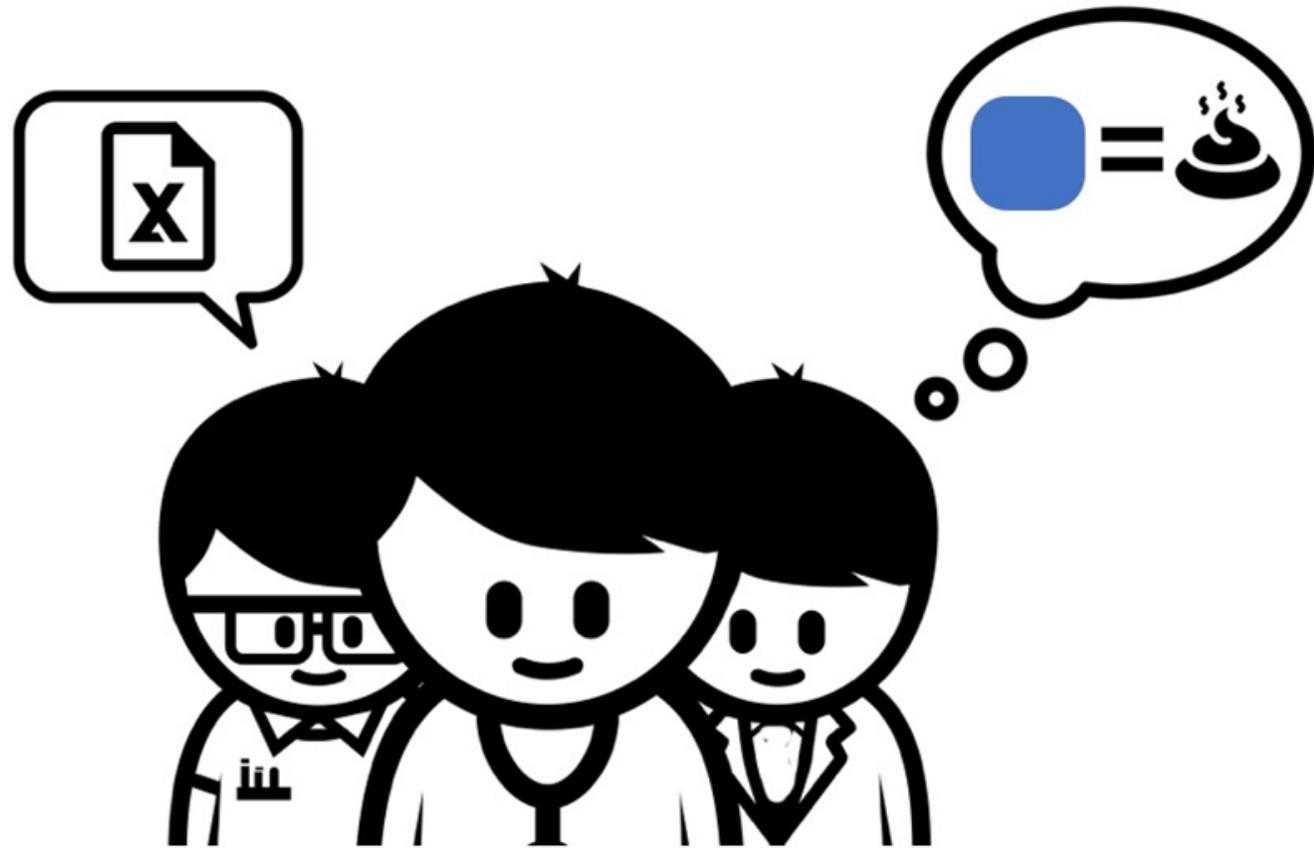


DESPUÉS DE MUCHO TIRA Y AFLOJA AMBOS  
ACUERDAN CERRAR EL DESARROLLO EN EL  
PUNTO EN EL QUE ESTÁ.



CUANDO PEPE SE LO MUESTRA A SUS USUARIOS ÉSTOS, ADEMÁS DE NO VER REFLEJADOS SUS PROBLEMAS, LO VEN COMPLICADO, INCOMPLETO Y POCO USABLE.

EVIDENTEMENTE DECIDEN SEGUIR CON SUS VIEJOS EXCEL...



**Moraleja**

# Pepe tiene algo que...

Le ha resultado más caro de lo que creía



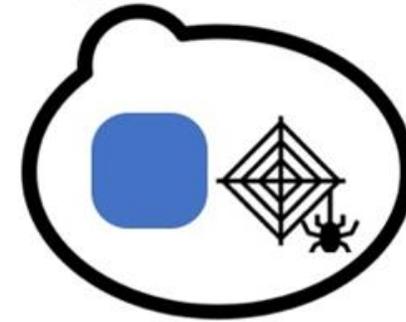
Por lo que ha tenido que esperar un montón



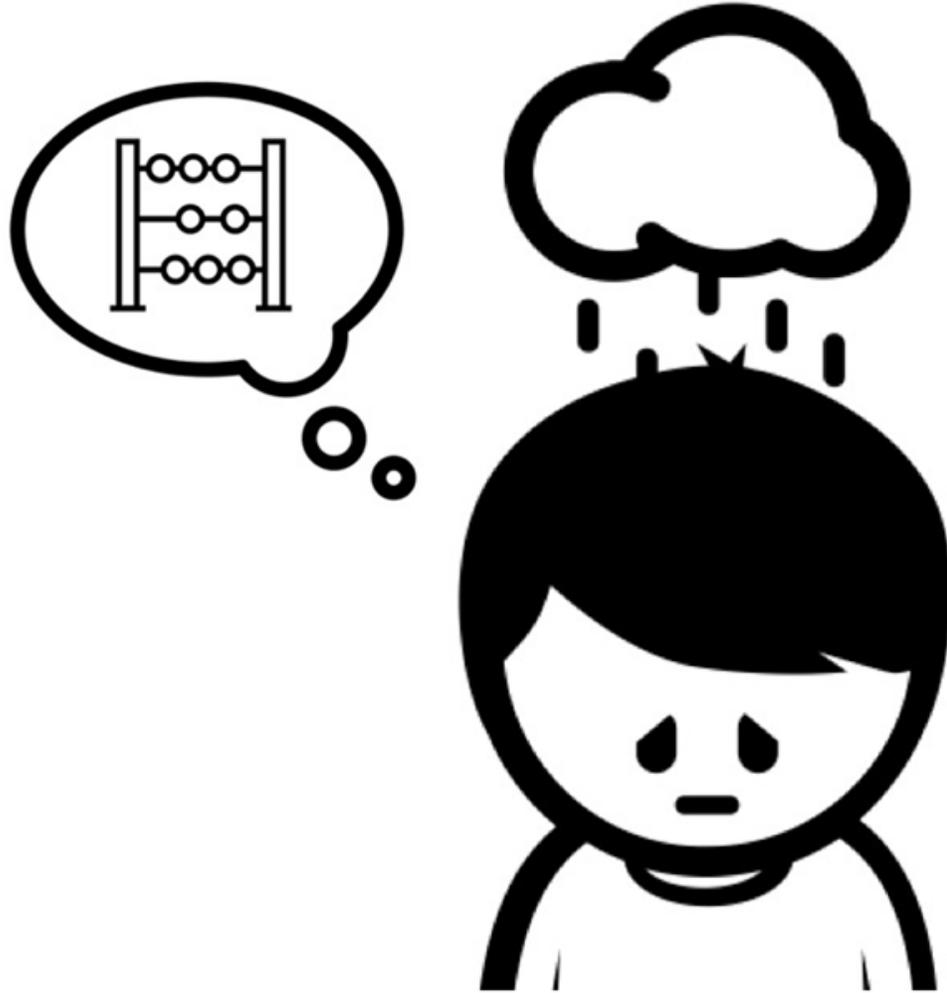
No es como el quería



Nadie va a usar



Pepe lo pensará dos veces la próxima vez



**Ahora, asumamos por un momento que...**

Hay un business case que sustenta el proyecto

Hay un sponsor ejecutivo que lo apoya

Quien desarrolla tiene los recursos adecuados

Quien desarrolla aplica las metodologías adecuadas

Quien desarrolla usa las herramientas e infraestructura adecuada

Quien desarrolla sabe gestionar el proyecto adecuadamente

**Y que los unicornios existen...**



**Pero entonces...**

**¿Qué ha podido ir mal?**

# ¿Se acuerdan del chiste?



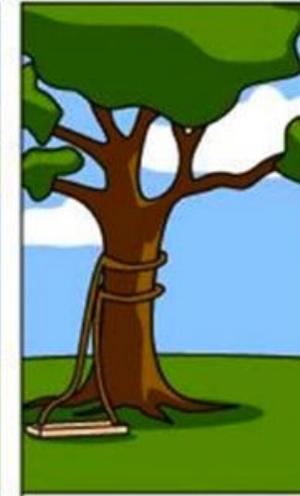
Cómo lo explica el cliente



Cómo lo entiende el Jefe de Proyecto del proveedor



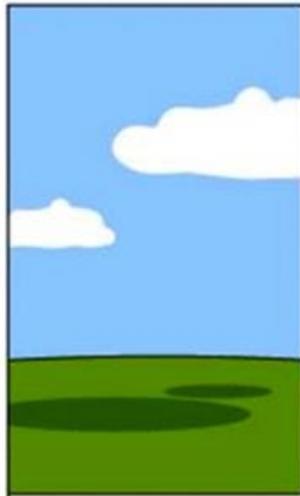
Cómo lo diseña el Analista Funcional del proveedor



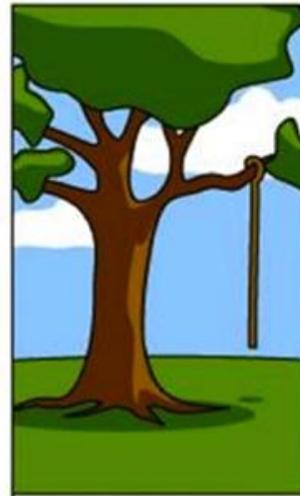
Cómo lo programa el programador del proveedor



Cómo lo describe el consultor de negocio



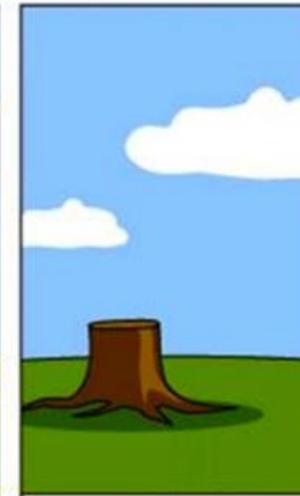
Cómo el proyecto es documentado



Lo que instaló el Departamento de IT



Cómo fueron facturando el proyecto



Cómo fueron dando soporte al proyecto



Lo que el cliente realmente necesitaba



**Todos cometemos pecados...**

---

# Soberbia

---



# Pensamos por los demás y creemos que tenemos la solución

---

No preguntamos a los usuarios reales, los que tienen los problemas y usan los programas

No validamos las propuestas con los usuarios

Los requisitos de lo que se "tiene" que hacer no reflejan necesidades reales

Caemos siempre en el decir que debemos hacer para curarnos en vez de decir que nos duele

Los "expertos" de los que crean el software creen que tienen la solución para todo.  
Ellos tienen la razón...



---

# Gula

---



# Abarcamos más de lo que podemos analizar, gestionar, tratar y digerir

---

Alargamos innecesariamente el tiempo en que disponemos de cosas tangibles y con valor

Ampliamos el riesgo de cometer errores

Nuestra ansia de controlarlo todo nos lleva a eternizamos en los detalles lo que aumenta exponencialmente los tiempos

Los que tienen que desarrollar el software aprovechan para aprender



---

# Pereza

---



# No sabemos lo que queremos, ni sabemos lo que no queremos

---

El software es algo intangible. Es difícil ponerle forma, color o medida...

Preferimos decidir en base a la comparación y eso es imposible con el software

Preferimos que nos muestren para decir si o no

Antes de entrar en detalle en algún tema, lo que nos consumirá mucho tiempo, preferimos dar las cosas por sentadas

Y eso pasa sobre todo en aquellos que tienen que idear el software. Cuando no entienden al cliente, suponen...



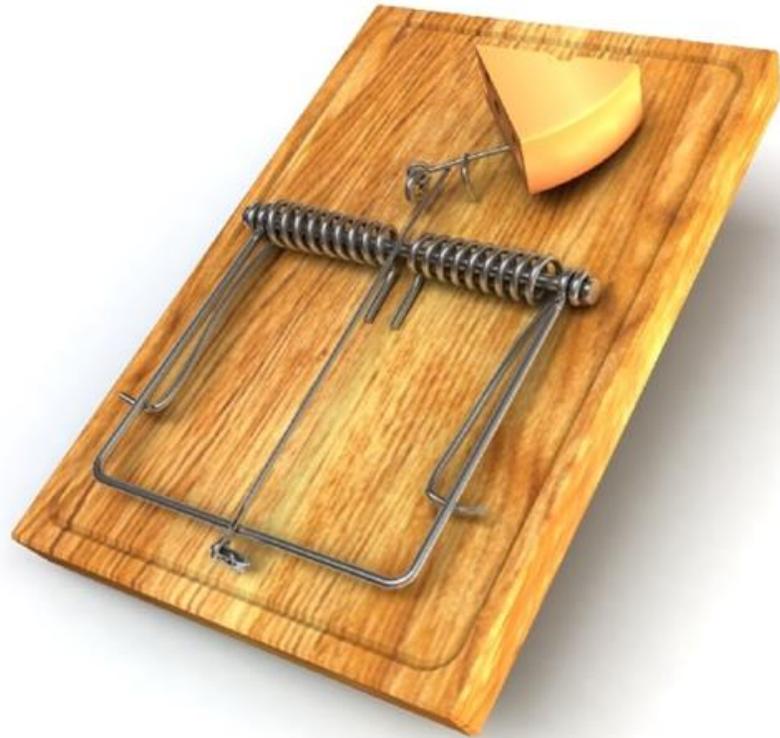
# Tal que así...



---

# Envidia

---



# Queremos lo que vemos en otras partes, aunque no entendamos bien que es...

---

Todos queremos salir perfectos en la foto y ser los más innovadores

Pero la tecnología avanza rápido y las cosas quedan obsoletas rápidamente

La gente cuando habla de software no miente, pero tiende a no decir toda la verdad. Son los pescadores modernos...

Queremos siempre lo más nuevo, lo que otros nos han vendido como perfecto

Pedimos que se hagan cosas por la excitación propia de la tecnología, no por necesidades de negocio



---

# Avaricia

---



# TODOS queremos TODO por el justo valor que tienen las cosas

---

Desconocemos lo que cuesta hacer las cosas y lo que cuesta adoptarlas

No renunciamos a la máxima calidad, aunque sea algo inasumible, al mismo precio

Escribimos, por si acaso, para que pueda decir "ya te lo dije"

Y al final tenemos lo que pagamos





**Y lo que debería ser todo Lujuria...**



**... se convierte en pura Ira**

# SOLUCIONES Y PENITENCIAS



**¿Las claves para encontrar  
la solución?**



**Son éstas...**

# TENER UNA RAZÓN

## Y UN SPONSOR

Debe haber un problema que solventar

Debe haber una necesidad que cubrir

Debe haber alguien dispuesto a liderar y pelearse por el proyecto

Debe haber un beneficio esperado para realizarlo (tangible o intangible)

Debe haber una motivación clara y alguien que tire...

...si no, mejor no comiences



# 2 PONER AL USUARIO EN EL CENTRO

## Y HACERLE CASO

El usuario sabe lo que le gusta

El usuario sabe lo que no le gusta

Es clave empatizar con él y conocer cual es su entorno

Es clave contar con su criterio y su validación

Debemos evitar las asunciones e ir a la fuente

Miremos y comprobemos en vez de hablar de oídas

Debemos contar con él continuamente durante el proceso de definición y creación



# 3 DISEÑAR Y TOCAR LAS COSAS

ANTES DE LANZARNOS

Evitemos charlas y discusiones intangibles

Pongámonos manos a la obra y trabajemos

Hagamos tangibles las ideas con prototipos

Opinemos sobre esos prototipos de "alta resolución" y avancemos

Prueba y error.

Mejor equivocarse antes e identificarlo que esperar al final...

El fallo es necesario



# 4 ENTENDER BIEN LOS PROBLEMAS

NO TAN SOLO REGISTRARLOS

La base de todo es la observación y el dialogo

No nos debe dar miedo preguntar

No nos debe dar miedo asumir nuestras miserias internas

El papel lo aguanta todo

Contrastemos lo que vamos identificando con acciones específicas

Pensemos globalmente y vayamos acotando para acabar focalizándonos en lo realmente importante



# 5 TESTEAR, VALIDAR, TESTEAR, VALIDAR...

Y ASÍ UNA Y OTRA VEZ

El camino al éxito normalmente no es una línea recta

El proceso de entender > crear > aprender debe ser lo más rápido posible

Antes validamos y avanzamos

Antes detectamos los errores

Antes decidimos que es lo que nos gusta o lo que no

La alternativa son tiempos interminables y poca información



# 6 MAXIMIZAR LA CREATIVIDAD

## Y APROVECHAR EL CONOCIMIENTO COLECTIVO

Salgamos de nuestra zona de confort

Juguemos y permitámoslo todo sin juzgar las ideas

Solo así aparecerán nuevas soluciones a los problemas tradicionales

El pensamiento integrativo es la clave. Todas las ideas de todos valen...

Cien mentes piensan mejor que una sola

Aprovechemos entonces el pensamiento colectivo y vayamos probando

Pero no confundamos... hay que tomar decisiones y avanzar al ritmo adecuado



# 7 CREAR VALOR DE MANERA ÁGIL

## MIENTRAS SOLUCIONAMOS PROBLEMAS

Construyamos también de manera iterativa implicando al usuario

Entreguemos cosas que puedan ser valiosas para los usuarios lo antes posible

No esperemos al final para puedan disfrutar

No esperemos al final para cambiar algo

Asegurémonos que lo que hemos diseñado juntos se materializa convenientemente

Es un ejercicio brutal de transparencia...

...aprovechémoslo

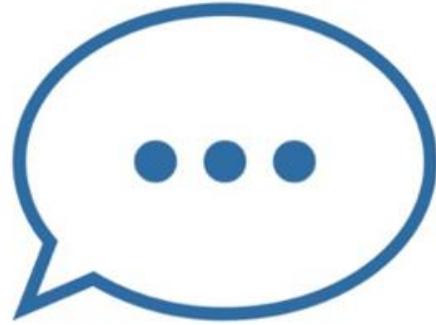


**¿La manera de conseguir  
vencer los obstáculos?**



**Es ésta...**

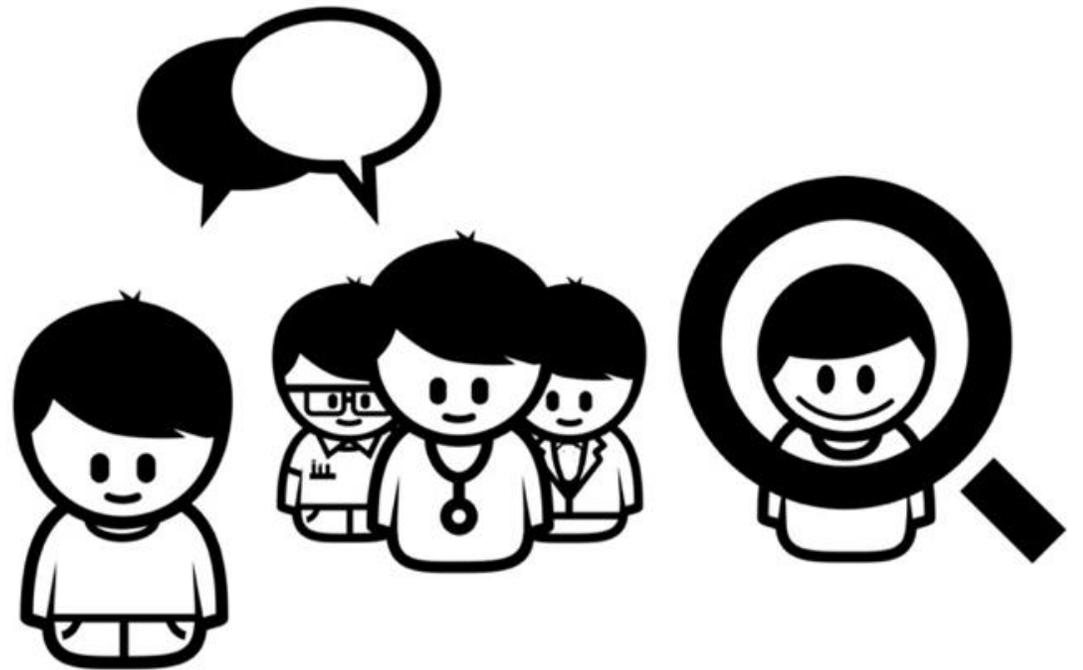
# Paso 1 – Descubrir y empatizar



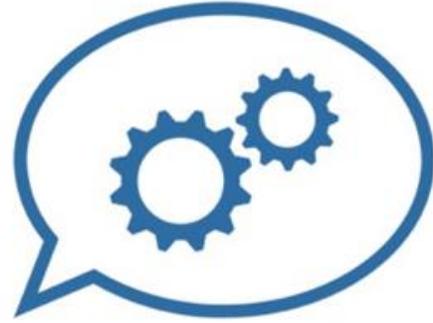
**¿Quién es realmente mi usuario?**

**¿Qué le pasa?**

**¿Cómo se comporta?**

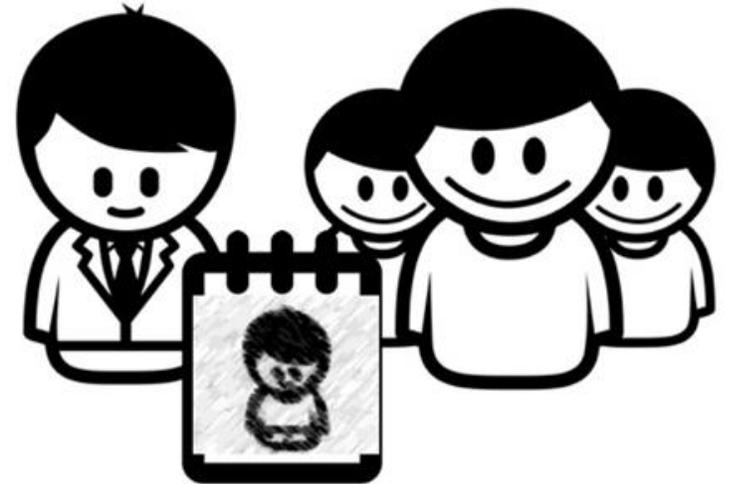


## Paso 2 – Diseñar



**¿Cuáles son las necesidades de mi usuario?**

**¿Cuáles son sus percepciones?**

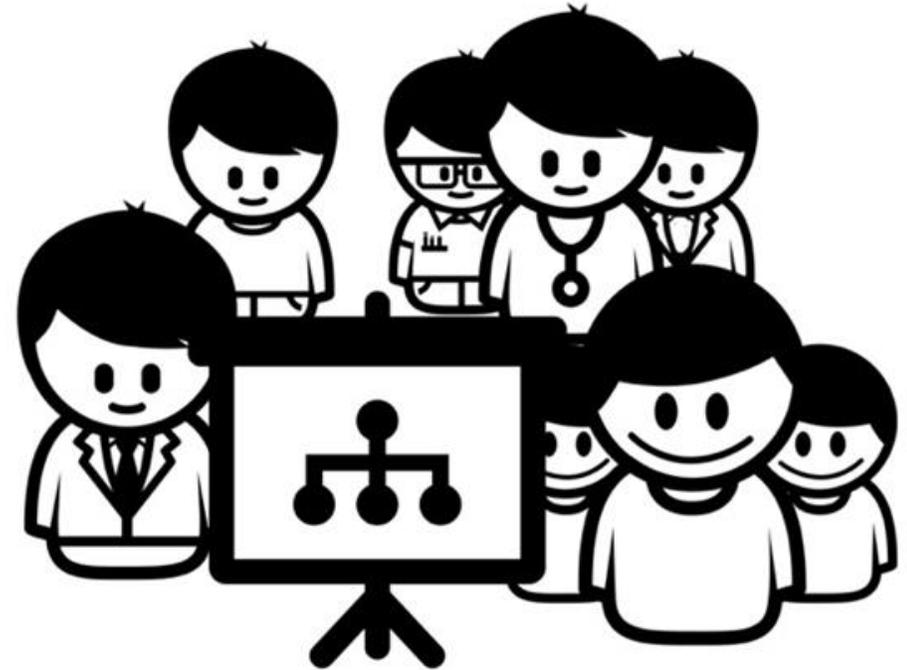


# Paso 3 – Idear



**¿Qué se me ocurre para cubrir esas necesidades?**

**¿Cómo puedo salirme de la norma y ser disruptivo?**

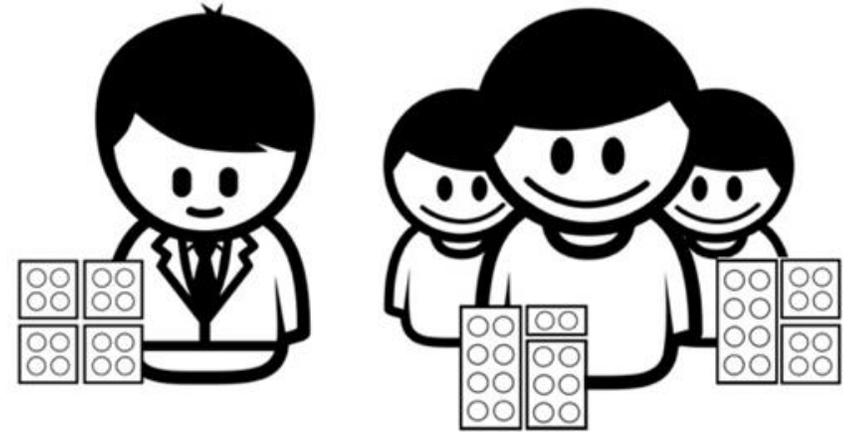


# Paso 4 – Prototipar



**¿Cómo puedo representar mis ideas?**

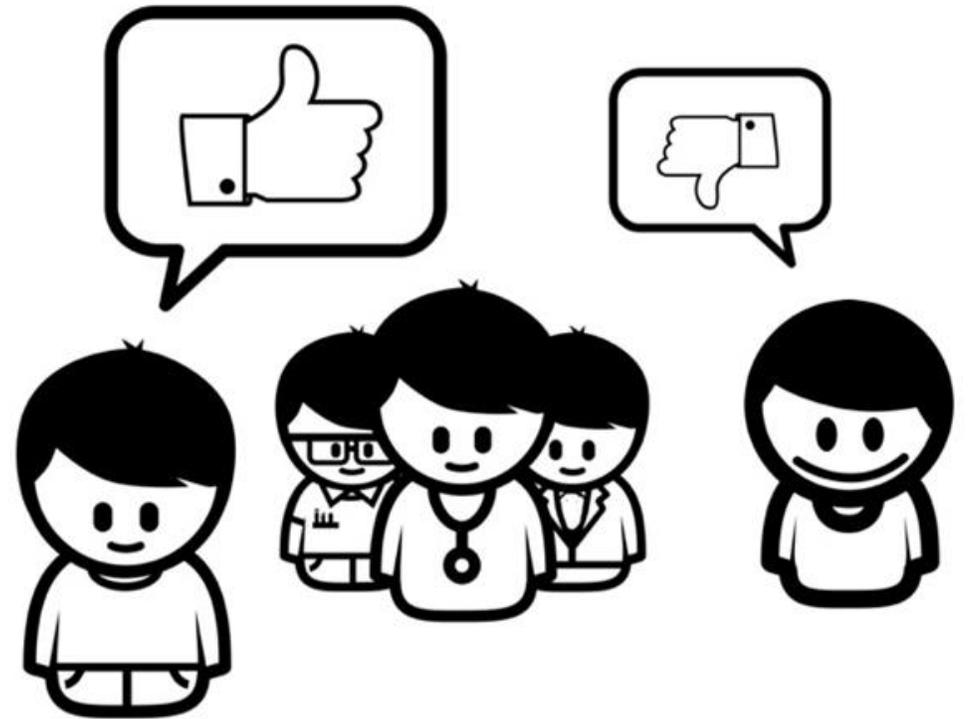
**¿Cómo puedo hacerlo lo más real posible?**



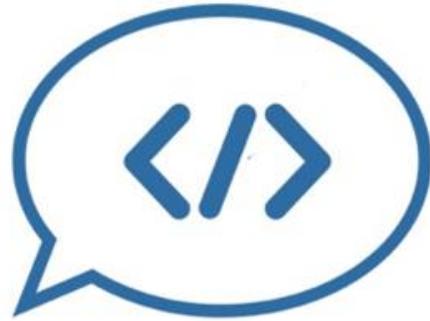
# Paso 5 – Testear



**¿Qué ha parecido bien?  
¿Qué no ha gustado?  
¿Cómo puedo ajustarlo?**



# Paso 6 – Construir



**¿Cómo puedo aportar valor  
cuanto antes?**

**¿Cómo puedo asegurar la  
calidad del producto?**



# Recapitulando...

## EMPATIZAR



---

Descubrir y entender las asunciones, preferencias y sesgos de la organización y las personas respecto al tema o los temas específicos que queremos solventar mediante observación y entrevistas

## DISEÑAR



---

Identificar e interpretar las tendencias y patrones observados respecto a las necesidades y percepciones de los usuarios

## IDEAR



---

Generar sets de mapas conceptuales divergentes, provocativos y diferentes a partir de los datos, los hechos, la creatividad, la intuición y la investigación

## PROTOTIPAR



---

Construir representaciones tangibles en forma de prototipos (lo más realista posible) de un número significativo de las ideas obtenidas para obtener feedback

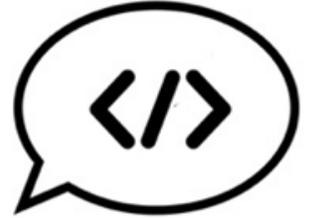
## TESTEAR



---

Compartir las ideas materializadas con los usuarios (principalmente los implicados en las fases anteriores) para conocer su reacción ante los prototipos planteados

## CONSTRUIR



---

Desarrollar e implementar la idea seleccionada de forma ágil e iterativa siguiendo los estándares de calidad que se consensuen con los usuarios

# BENEFICIOS Y PENITENCIAS



**Dos opciones a elegir**

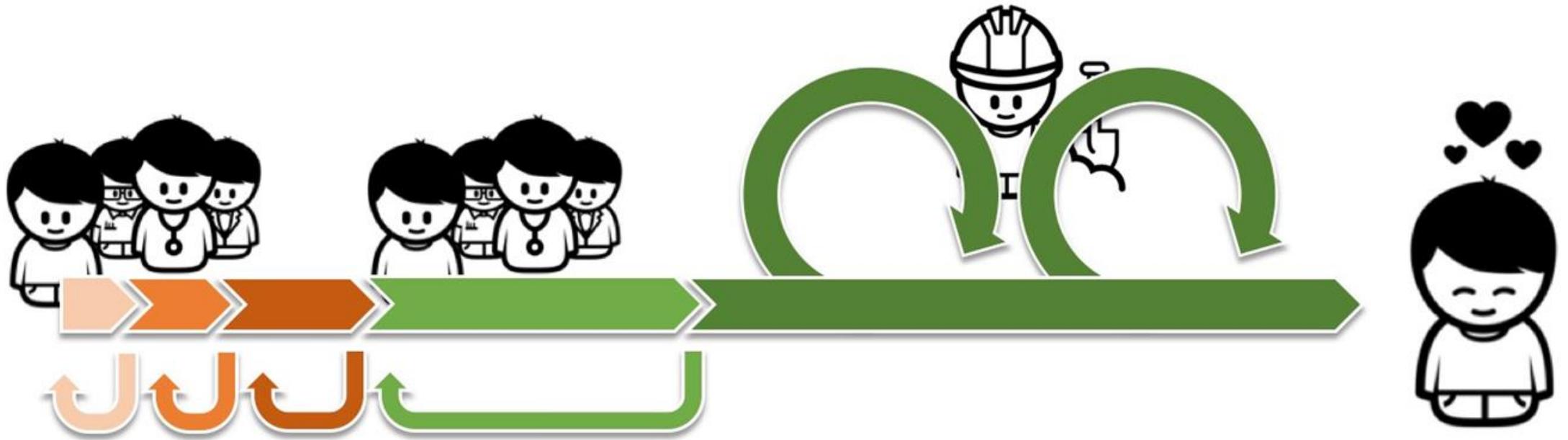


# Lo de siempre...

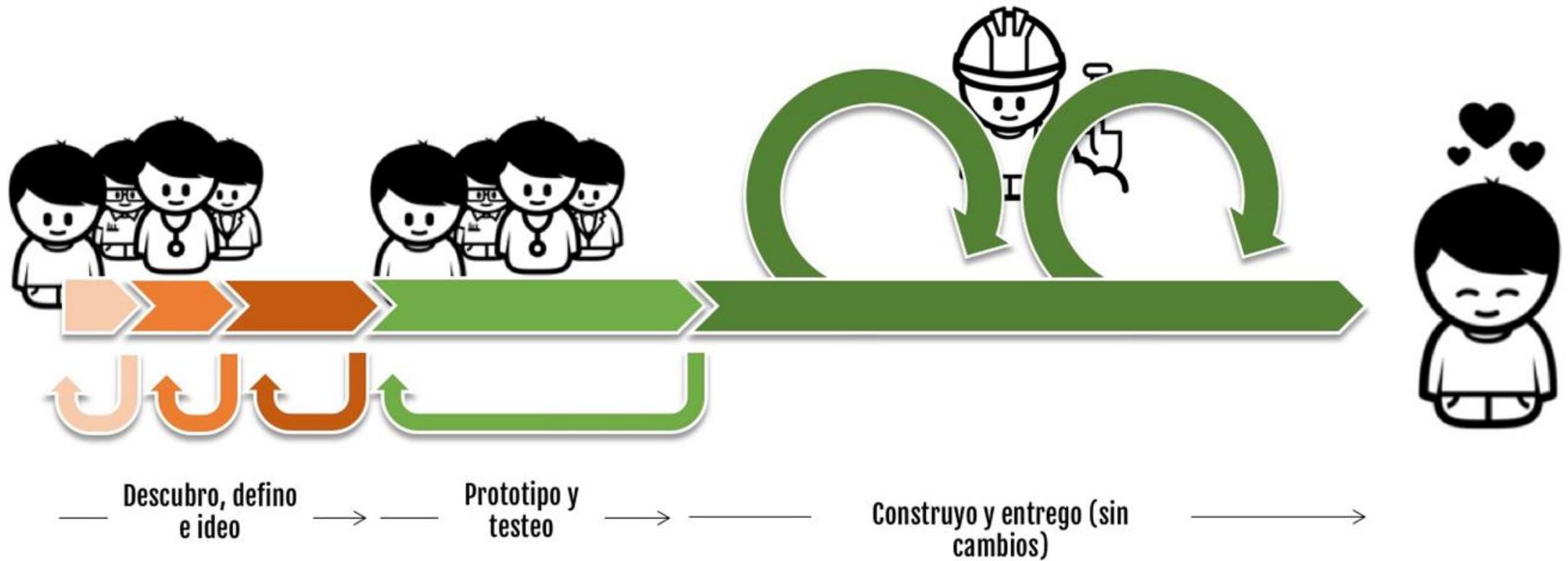


...de la que ya sabemos el resultado

# 0 una nueva visión



# Cuando el DESIGN THINKING se encuentra con AGILE



**Muy bien pero...**

**¿Qué ganamos?**

**Aseguramos el tiro**



**Lo que obtienes es válido aunque  
para ello tengas que dar mil  
vueltas**

**Si no llegas al final es que el  
proyecto no era necesario**

**Mejor eso que llegar al final para  
darse cuenta de que no funciona**



**Garantizamos el uso  
futuro**



**El resultado es algo consensuado  
entre todos por lo que se minimiza  
el riesgo de rechazo**

**Aseguramos la implicación total de  
los usuarios...**

**...aún a costa de una mayor  
inversión de su tiempo**



**Reducimos el tiempo...  
y el coste**



**Prototipamos y validamos con los usuarios...**

**...reduciendo el tiempo de desarrollo y la posibilidad de cambios**

**Esas son las fases donde tradicionalmente se gasta más tiempo**



**Pero ojo...**

**No hacer caso de estas recomendaciones (una razón, un sponsor, mimar al usuario, prototipar...) puede hacernos caer en una excesiva iteración y por consiguiente en que nos comamos el tiempo ganado**





**¿Quién quiere pasar por el  
confesionario?**

**danicardelus@gmail.com**

**@dcarTech**

# IMÁGENES / CRÉDITOS

[HTTP://THENOUNPROJECT.COM/](http://thenounproject.com/)

JAVIER CABEZA  
VICON DESIGN  
LORENA SALAGRE  
CHRISTOPHER HOLM-HANSEN  
MUNDO  
JACK DUNHAM  
NICOLAS VINCENT  
CREATIVE STALL  
PETR PASASOV  
MUSKET  
JONATHAN LI  
BOHDAN BURMICH

ANBILERU AMALERU  
ARTHUR SHIAIN  
GREGOR CRESNAR  
SHMIDT SERGEI  
ICON FAIR  
UMESH VGI  
UNLIMICON  
KARTHIK AATHIS  
DAVO SIME  
SARA