Clase 14:

Gestión de Proyectos, metodología agile y UI/UX

profesor: **Patricio López Juri** { patricio@lopezjuri.com }
créditos: 10
horario: J:7-8
sala: H3

Supuesto que hago yo en este curso:

Ustedes son personas inteligentes y pueden aprender y deducir reglas lógicas en base a ejemplos.

Repaso

- ¿Qué es un CMS?
- ¿Por qué usaríamos uno como Wordpress en vez de hacerlo en Rails?

Hasta el momento es relativamente simple trabajar uno solo en un proyecto pequeño.

¿Qué pasa cuando el equipo crece y/o el proyecto también?

Cosas que pasan: **Código duplicado**

Cuando un fragmento de código es escrito en un proyecto donde ya existía un código idéntico o similar que cumple la misma función.

- Por ejemplo, es escribe el CSS del los <h1> dos veces.

Cosas que pasan: **Formato inconsistente**

Cuando el código es heterogéneo y es fácil distinguir que dos o más personas programaron en este. Lo ideal es que el código sea homogéneo.

- Por ejemplo, alguien programa en inglés, otro en español.
- Otro ejemplo es que alguien con más experiencia usa métodos/funciones más avanzadas y en otras partes vemos código más principiante.

Cosas que pasan: **Problemas de versionamiento**

En palabras simples: **mal uso de git.**

Si es usa mal Git se pueden encontrar con problemas de no poder mezclar el trabajo de distintas personas.

Cosas que pasan: **Se programa algo que no sirve**

Cuando lo que debe cumplir el software no es claro, suele pasar que se programa algo que no era lo que realmente se necesitaba. Esto suele ser una de las principales razones por las cuales los proyectos fracasan.

- Se programa una aplicación web que necesita internet para un local donde no hay internet.
- Se programa el sistema de un aeropuerto pero no se pueden cambiar asientos.

Cosas que pasan: **Se programa algo que no sirve**

Otras veces pasa que se programa algo que funciona, pero nunca se necesitó, nunca se solicitó solicitó o no entrega valor a los usuarios.

- Una ventana para chats o reproductor de videos que nunca se necesitó y queda escondido.

Cosas que pasan: **No se terminó en el plazo**

Programar es difícil y uno suele encontrarse con demasiados errores o problemas que uno no tiene contemplados al planificar los plazos. Por lo general se espera que se entregue el software en un plazo fijo porque se tiene agendado su lanzamiento.

- Al estar contra el tiempo el software se vuelve más caro, con más errores o simplemente queda incompleto.

Cosas que pasan: **El software no es usable**

Cuando uno diseña y programa el software por meses, uno lo conoce de memoria. Sin embargo, los usuarios finales se encuentran con muchas vistas y botones y no saben qué hacer.

A veces pasa que realizar ciertas acciones requieren conocimiento previo o realizar acciones poco intuitivas. No siempre se puede capacitar a las personas. Aún así capacitar es caro.

- El usuario no entiende cómo hacer cierta acción, se frustra y nunca más vuelve a utilizar el software.

Cosas que pasan: **Resistencia al cambio**

A la gente no le gusta que le cambien el software que usa porque le tomó mucho esfuerzo entenderlo.

Basta con cambiar de posición de un botón y la gente pierde la cabeza.

Cosas que pasan: **El cliente no sabe lo que quiere**

Los clientes o personas que solicitan un software no suelen ser siempre personas con conocimientos técnicos. Vienen con una idea o intuición muy *idealizada* de lo que necesitan. Muchas veces ni siquiera es posible técnicamente su solución. La mayoría de las veces no tienen claridad de cómo quieren que funcionen las cosas. Esperan que una aplicación *mágica* solucione todos sus problemas.

Cosas que pasan: **El cliente cambia de opinión**

Otras veces se dan cuenta al final o durante el desarrollo que necesitan otra funcionalidad o una existente pero de otra forma.

A veces una funcionalidad que parece pequeña para el cliente resulta que es necesario reestructurar la aplicación desde cero.

La gente prefiere la “zona de confort”.

La mayoría de los proyectos fracasan,
pero no culpa de la tecnología,
sino por problemas con las personas.

Algunas definiciones importantes

Definiciones:

Feature (característica o funcionalidad)**:**

Acción que permite hacer el software.
Por ejemplo:

- Se puede iniciar sesión con el correo.
- Se puede generar una boleta y enviarla al SII todos los meses.

Definiciones:

Product Owner:

Es quién representa el cliente del proyecto. El objetivo es cumplir la idea de proyecto que él o ella tiene. Debe ser capás de resolver todas las dudas que tenga el equipo de desarrollo.

Puede ser alguien mismo del equipo si es un proyecto interno.

Definiciones:

Requisitos:

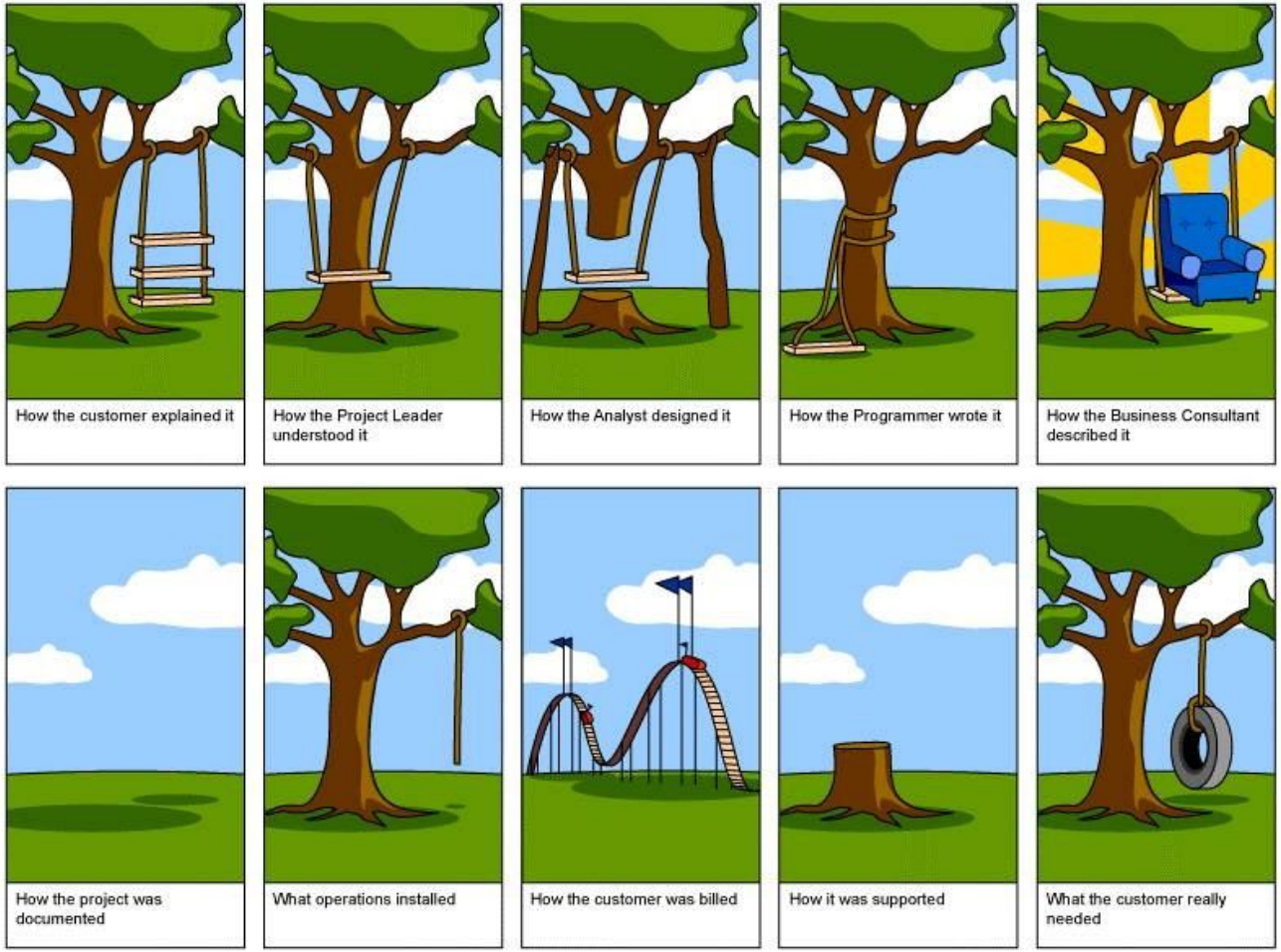
Son especificaciones formales de las cosas que hace o debe hacer el software.

Gestión de proyectos TI

El proyecto informático es solo una parte del proyecto verdadero

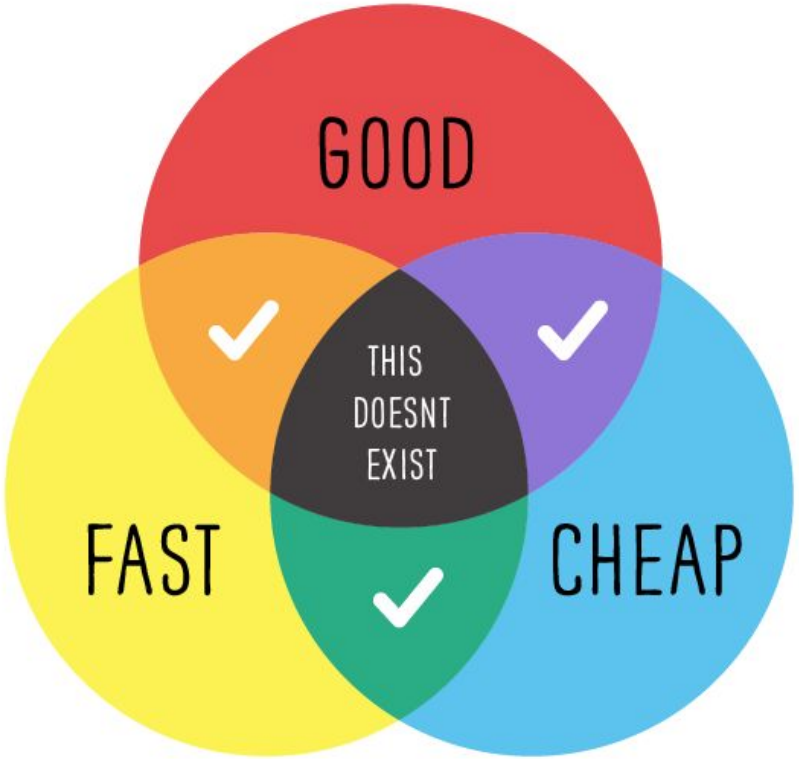
- Todo proyecto informático viene acompañado por un cambio organizacional. Muchas veces la gente no se da cuenta de esto.
- Empresas o Startups que son totalmente digitales deberían poner en el centro de su organización el software y no pensarlo como un “simple agregado”.
- Alrededor del 90% de los proyectos se retrasan y solo el 33% tiene éxito en su implementación.*

* cita pendiente



Disculpen lo trillado de la imagen.

No se puede tener todo



Por lo general es bueno tener una PMO

La ***Project Management Office*** de una empresa estandariza los procesos relacionados a la gobernabilidad de los proyectos y provee los recursos, herramientas y metodologías a usar en los proyectos.

- Monitorear el rendimiento de los proyectos (eficiencia)
- Alinear los proyectos con los objetivos del negocio (eficacia)
- Etc

Un nivel más avanzado se encarga incluso de la mejora contínua de los proyectos.

Metodologías de desarrollo

Existen metodologías para el desarrollo de proyectos

“Aplicarlas no es una receta para el éxito, no hacerlo es una receta para el fracaso”

Las metodologías nos sirven para abordar un proyecto

- Son formas de separar el proyecto en etapas y cómo abordar cada una.
- Son formas de definir tiempos.
- Son formas de definir roles en el equipo.
- Alguna metodologías sugieren el uso de herramientas como pizarras u otras más modernas y digitales.

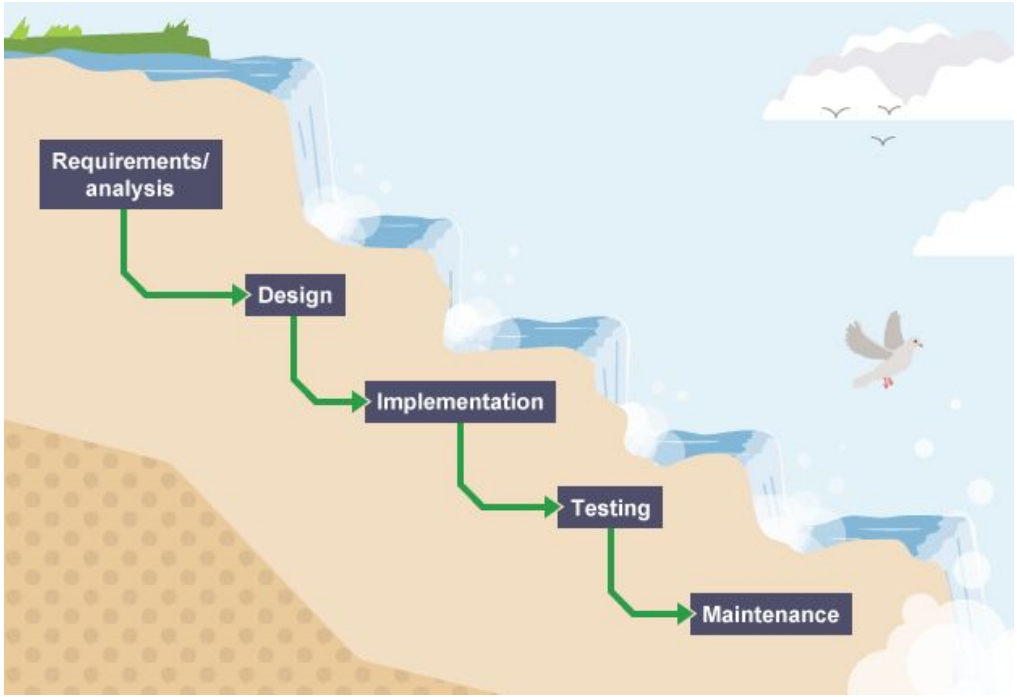
Metodología

CASCADA

- La metodología que es prácticamente una no-metodología.
- Pocas veces sirve efectivamente.
- Usada comúnmente por principiantes que conducen el proyecto al fracaso.

Su nombre va a que es un proceso de inicio a fin directo

1. Se toman los requisitos
2. Se construye el software
3. Se hacen algunas pruebas
4. Se entrega el software



¿Por qué casi siempre sale mal?

En los proyectos de software:

- El cliente no sabe lo que quiere o cambia de opinión y es poco probable que el software final sea lo que realmente quiera (o que realmente necesite).
- Nos solemos encontrar errores/problemas que nos atrasan. ¿Qué hacemos con los plazos?
- No dice nada de cómo debe trabajar y distribuirse el equipo.

¿Cuándo podría resultar bien?

Pocas veces, pero es factible usarlo cuando el proyecto es bien definido por agentes técnicos y ya fue evaluado por expertos. Se deben disponen de todas las especificaciones técnicas y no permitir ambigüedades.

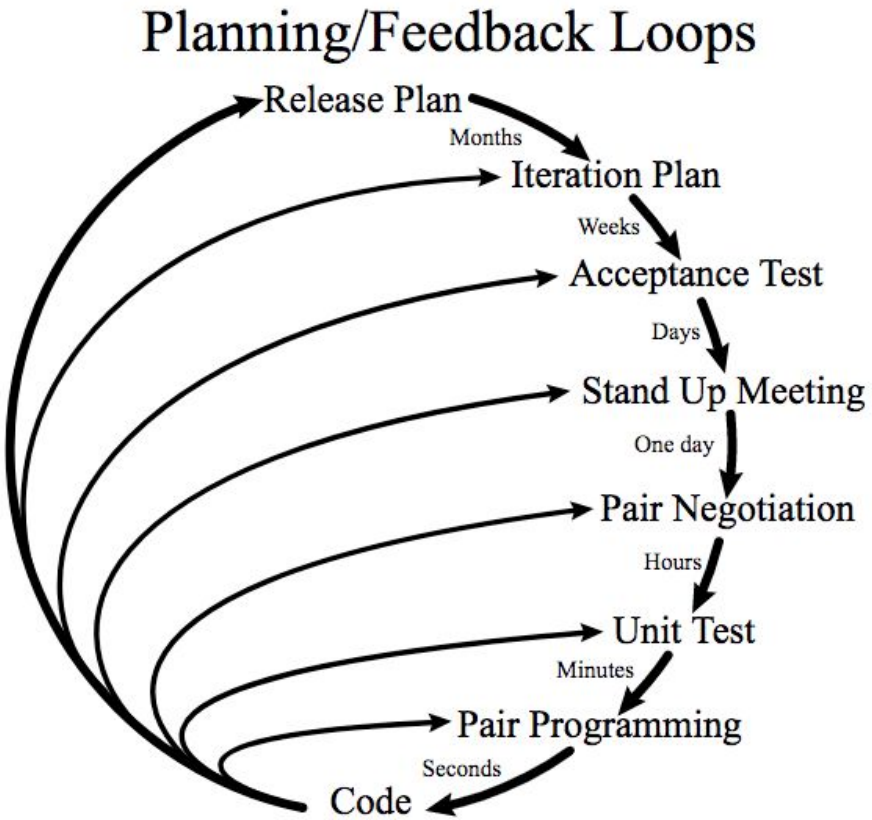
Metodología

XP: Extreme Programming

- Los requisitos del software pueden cambiar muy rápido en el tiempo, especialmente en Startups y en Hackathons.
- Esta metodología se basa en tener algo funcionando con poca o nula formalidad en vez de algo incompleto y con mucha burocracia.
- Se necesita que el product owner esté en todo momento con el equipo, prácticamente sentado con ellos.
- El equipo debe ser muy unido y colaborativo.

Es un proceso con mucha iteración inmediata

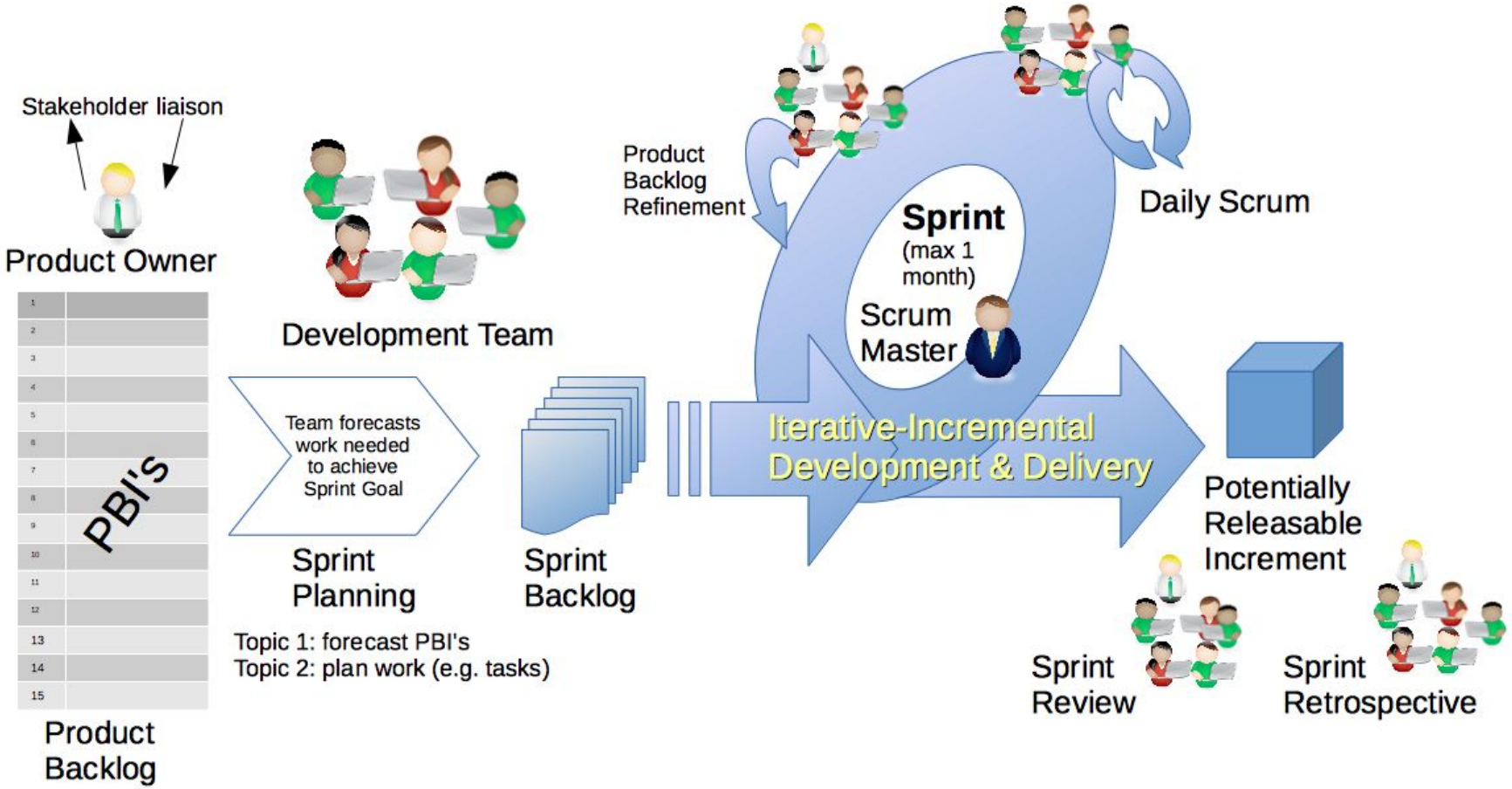
Por esto es importante que el equipo tenga buenas comunicaciones y exista confianza entre ellos.



Metodología

Scrum

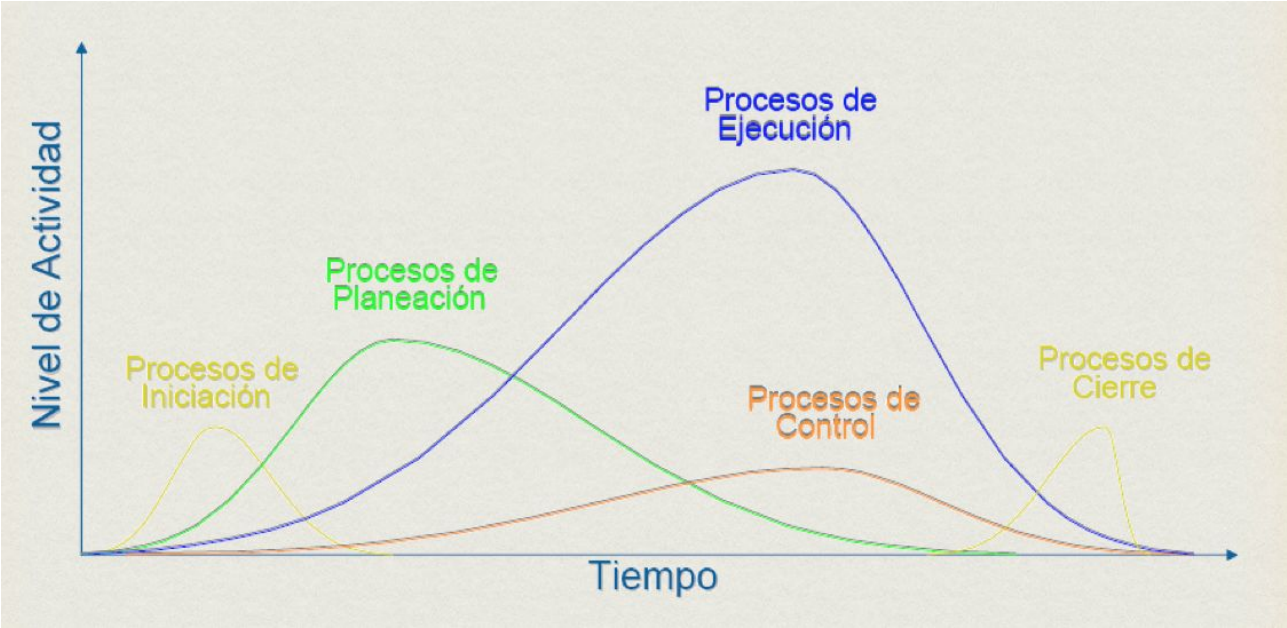
- Se divide el proyecto en tareas pequeñas. Estas tareas se agrupan en periodos de tiempo (por lo general 3 semanas) que se llaman *sprints*.
- Al estar dividido en ciclos se pueden hacer cambios al plan original entre ciclos. Al final de cada *sprint* el proyecto debe avanzar incrementalmente
- Se define un *Scrum master* quien lleva el control del *sprint* actual.
- Todos los días se comienza con un *Stand-up meeting*


Stakeholder liaison
Product Owner
PBI's
Product Backlog
Development Team
Team forecasts work needed to achieve Sprint Goal
Sprint Planning
Topic 1: forecast PBI's
Topic 2: plan work (e.g. tasks)
Sprint Backlog
Product Backlog Refinement
Sprint
(max 1 month)
Scrum Master
Daily Scrum
Iterative-Incremental Development & Delivery
Potentially Releasable Increment
Sprint Review
Sprint Retrospective

Existen más, pero lo importante es quedarse con el concepto de **ciclos.**

Dividir el proyecto en ciclos permite aumentar la probabilidad de éxito porque uno es flexible ante cambios y problemas.

Entonces en cada ciclo se abarcan distintas partes del proceso



UI / UX

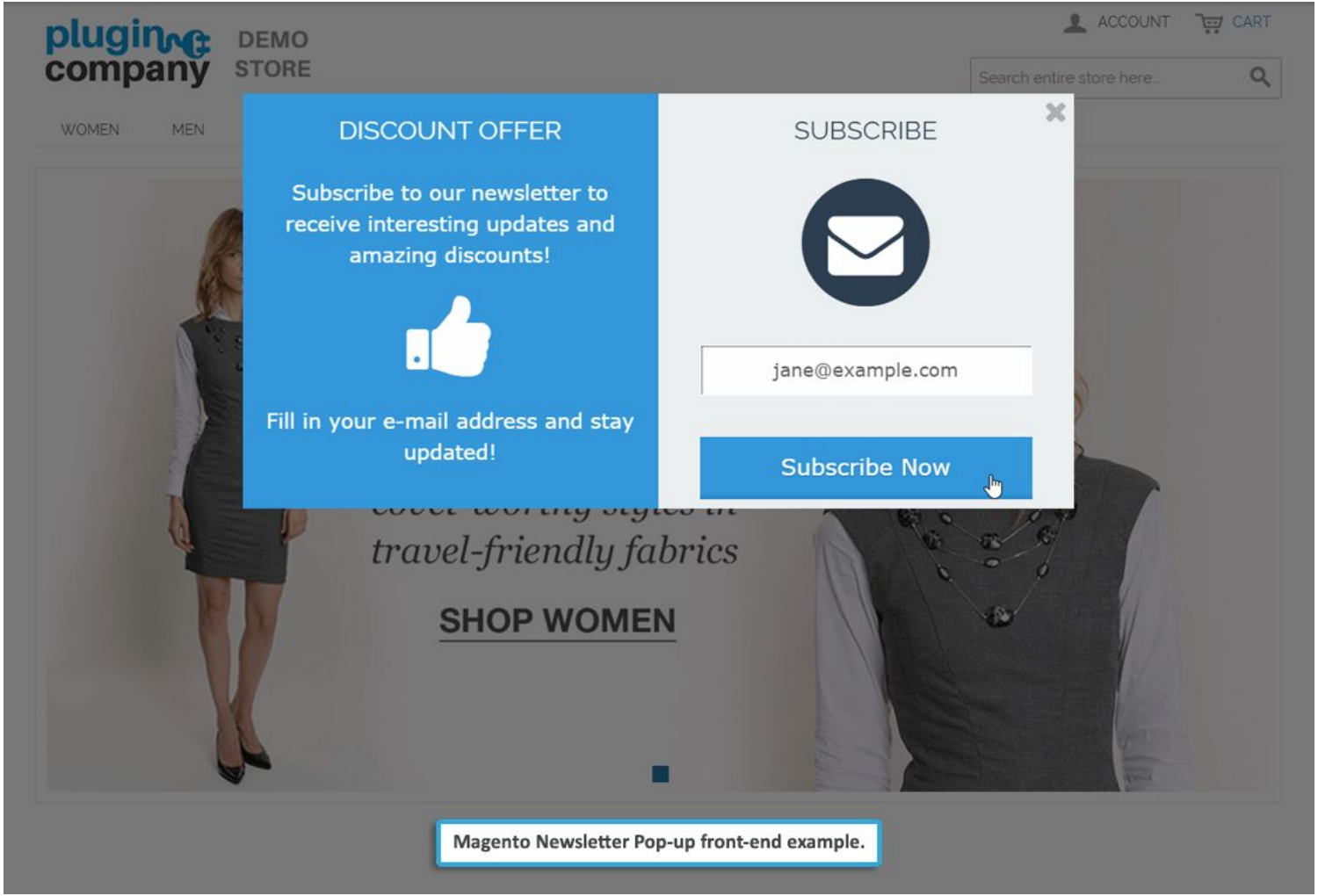
UI / UX

User Interface / User Experience

Una cosa es que se vea bonito, otra cosa es que sea funcional


+
DURACELL
PLUS
+
DURACELL
PLUS


AT&T
replaytv
SONIC blue
DVD
Pioneer
SONY
TV
Pioneer
Panasonic
LIGHT TOWER



Magento Newsletter Pop-up front-end example.

BURGER KING® App
BURGER KING
FREE - In the Windows Store
INSTALL

Find Your *BK*® Location

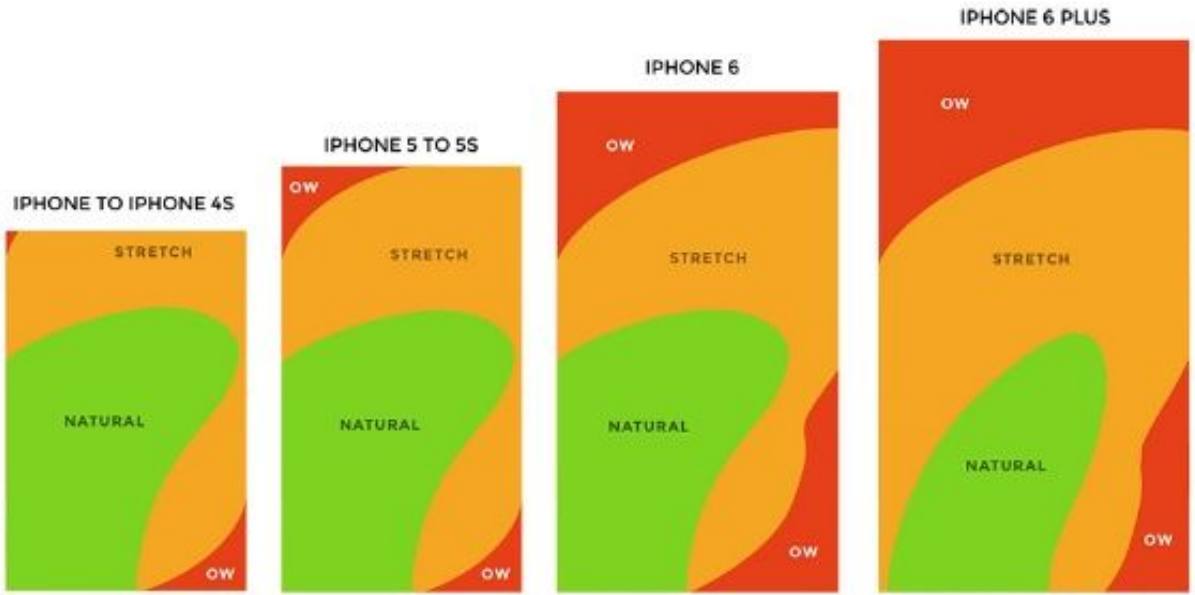
Bacon & Cheese Whopper® Sandwich

Extra Long Cheeseburger

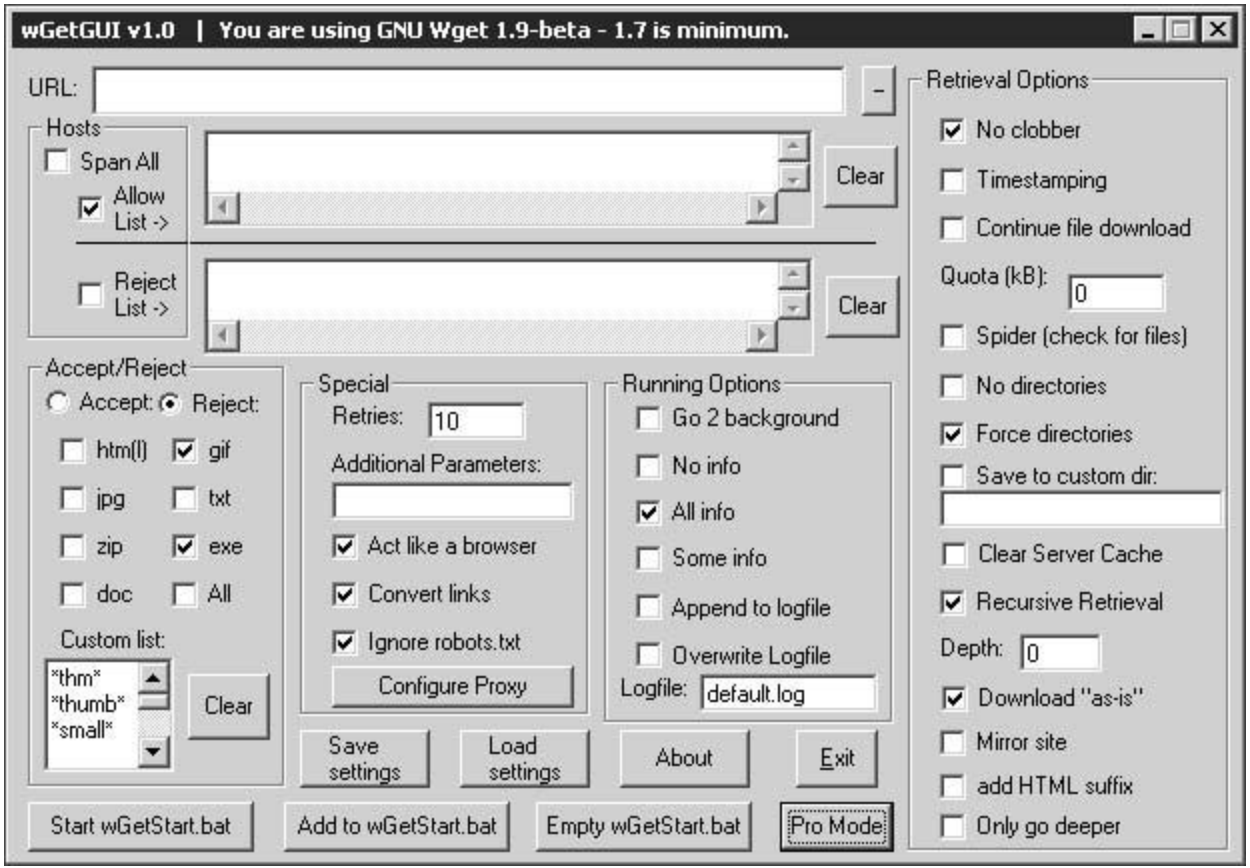
www.bk.com/menu/burg

The Thumb Zone evolution

One Handed



Credit: Scott Hurff


wGetGUI v1.0 | You are using GNU Wget 1.9-beta - 1.7 is minimum.
URL:
Hosts
Span All
Allow List ->
Reject List ->
Clear
Clear
Accept/Reject
Accept:
Reject:
htm(l)
gif
jpg
txt
zip
exe
doc
All
Custom list:
thm
thumb
small
Clear
Special
Retries: 10
Additional Parameters:
Act like a browser
Convert links
Ignore robots.txt
Configure Proxy
Running Options
Go 2 background
No info
All info
Some info
Append to logfile
Overwrite Logfile
Logfile: default.log
Retrieval Options
No clobber
Timestamping
Continue file download
Quota (kB): 0
Spider (check for files)
No directories
Force directories
Save to custom dir:
Clear Server Cache
Recursive Retrieval
Depth: 0
Download "as-is"
Mirror site
add HTML suffix
Only go deeper
Save settings
Load settings
About
Exit
Start wGetStart.bat
Add to wGetStart.bat
Empty wGetStart.bat
Pro Mode

Recapitulando

- ¿Qué puede pasar si no tenemos cuidado al hacer un proyecto TI?
- ¿Cuál es el problema principal de la metodología Cascada?
- ¿Cuál es la diferencia entre UI y UX?