La propuesta ágil

"Embrace change"

agustin.villena@gmail.com

Metodologías Ágiles

> ***Software fails to deliver…***
> ***and fails to deliver value!***
> Kent Beck, "*Extreme Programming Explained*", 1999

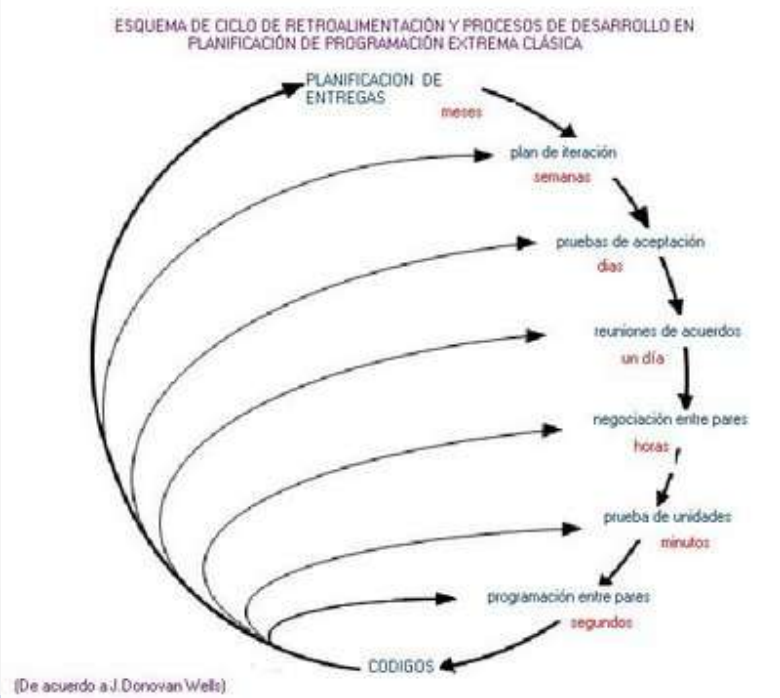
- En 2001, Kent Beck y otros autores de enfoques similares proponen los ***Principios Ágiles:***

Individuos e interacciones	por sobre	Procesos y herramientas.
Software funcional		Documentación exhaustiva
Colaboración con el cliente		Negociación de contratos
Responder al cambio		Seguir un plan

avillena@dcc.uchile.cl

Ambiente de desarrollo ágil

- Caracterizado por la continua
  - Colaboración y Comunicación entre clientes y desarrolladores
  - Retroalimentación y Validación hacia la creación de
    - Productos **Valiosos** a través de Módulos de Software **Funcionales**
  - Reflexión
  - Adaptación



El impacto ágil en la industria

- Líderes en el área están adoptando el enfoque ágil
  - Microsoft, Yahoo, Google
- RUP, la metodología tradicional más importante, se plantea "*compatible*" con XP
  - Aunque su enfoque tecno-céntrico no es muy "agil"

agustin.villena@gmail.com

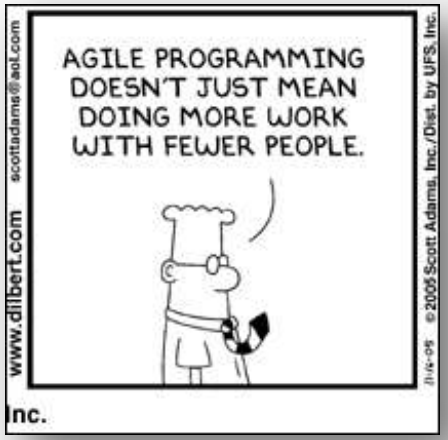
El cambio como certeza

- En un proyecto que implique innovar existe una sola certeza: el **cambio**
  - En las condiciones **internas** o **externas**
  - Porque al **adentrarse en definir la solución** ahora se **conoce mejor** el **problema a resolver**
  - Etc

agustin.villena@gmail.com

La propuesta ágil

- **Aprender** lo **más rápido posible**,
  a partir del **desarrollo** mismo, el cual es realizado
  - **concurrentemente**, en vez de analizar, diseñar, programar, probar… secuencialmente
  - en **experimentos** e **incrementos pequeños**,
  - de **éxito verificable** (para **cliente** y **desarrollador**),
  - buscando **generar el mayor valor** lo **antes posible**



agustin.villena@gmail.com

Elementos de Gestión Ágil

- Definamos un nuevo modelo, donde un proyecto con incertidumbre se gestiona
  - Definiendo un **plan preliminar**
    - Confomado por requerimientos, que son **ambiguos** pero **útiles,** es decir **estimables** y **verificables**
  - Obteniendo **retroalimentación exacta** y **precisa** del **avance y estado** de un proyecto
  - Invirtiendo los **menos recursos posibles** al comienzo
    - se elimina **todo aquello que no genere valor**.
      - Por ejemplo, el sobrediseño
  - Ofreciendo la oportunidad de **ir más rápido**
  - Permitiendo **cambiar drásticamente** los requerimientos (**alcance variable**)

agustin.villena@gmail.com

Elementos de Gestión Ágil

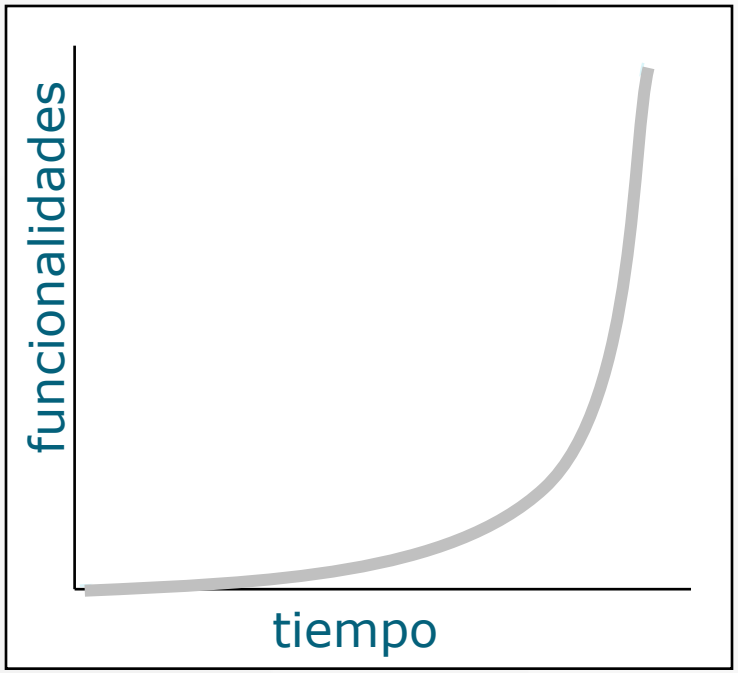
Definición de Gestionar

(según la RAE)

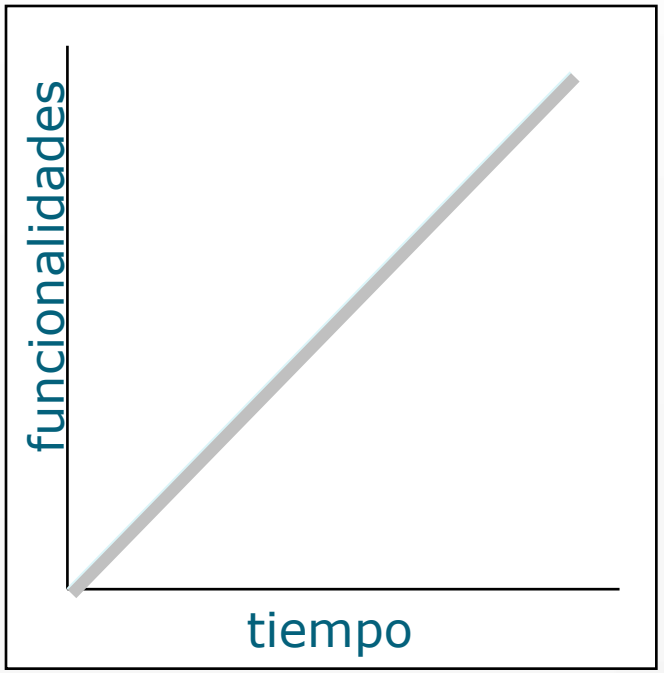
- Graduar o dosificar el uso de algo, para obtener mayor rendimiento de ello o para que produzca mejor efecto.
- (sinónimo): administrar

agustin.villena@gmail.com

El impacto en el valor entregado del proyecto en su tiempo de desarrollo



Valor entregado de un proyecto tradicional



Valor ideal que debiera entregar un proyecto

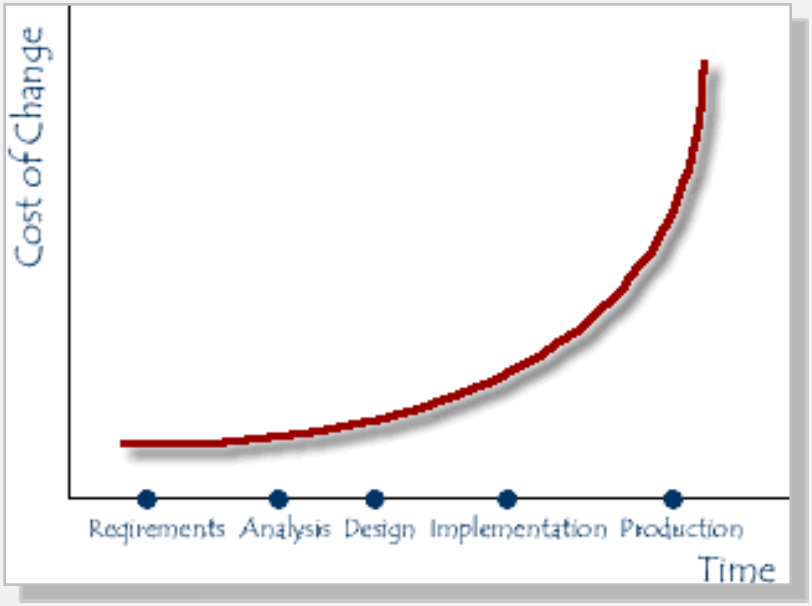
Tradicionalmente

¿No debiera ser así?

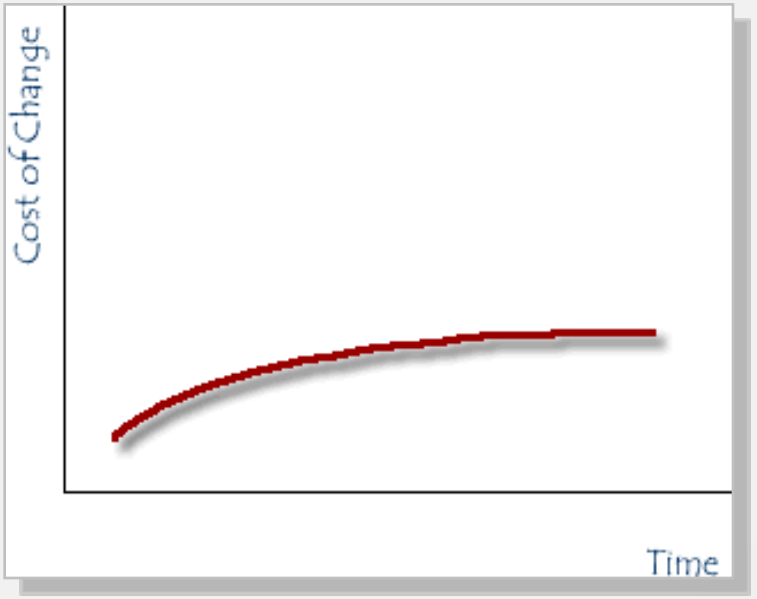
agustin.villena@gmail.com

El impacto en la agilidad del proyecto

- Visión tradicional



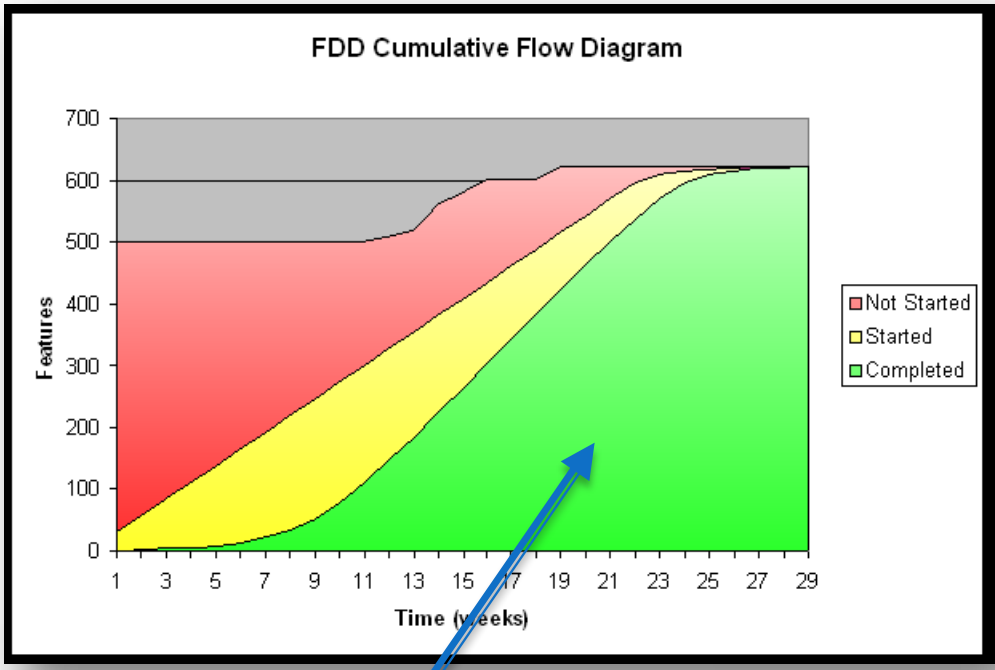
- ¿Y si fuera así?



Elementos de Gestión Ágil

Cómo se logra el mayor valor de un proyecto

- El desarrollo se debe **dividir** en **iteraciones cortas** de **tiempo fijo**
- Al fin de cada iteración del desarrollo, debe tenerse un **producto funcional** y **útil**



Alto valor promedio para el cliente

agustin.villena@gmail.com

Elementos de Gestión Ágil

Variables de gestión de un proyecto

- **Tiempo** disponible para el proyecto
- **Costo (Recursos)** que se dispondrán
- **Alcance,** es decir conjunto de funcionalidades que se desarrollarán
- Y la que se suele olvidar
  - **Calidad** (eficacia, resistencia a fallas, eficiencia, etc.)

agustin.villena@gmail.com

Elementos de Gestión Ágil

El cambio y la gestión tradicional

- Por lo regular se fijan las 3 primeras variables:
  - *"Hazme un administrador de estas tablas de la base de datos, en una semana, con dos programadores"*

- ¿Qué sucede al haber cambio?:
  - **Tiempo** y **Costo** no son en realidad muy flexibles
  - **El Alcance está fijo** por el diseño **y el plan definido**
  - Variable que sufre: la **Calidad,**
    lo que afecta el **valor generado** para el cliente

  - Esto genera **conflicto**
    - El cliente siempre querrá el mayor valor por sus recursos invertidos
    - Y el desarrollador la mayor calidad
      ¿Qué persona sana desea hacer un trabajo mediocre?

agustin.villena@gmail.com

Elementos de Gestión Ágil

Timeboxing y Contratos de Alcance Variable

- Definamos contratos en que la duración de cada iteración esté **prefijada,** inamoviblemente
  - ¡No existen “atrasos”!

- Y lo que se ajusta es el **alcance** (cantidad y cualidad de las funcionalidades) del proyecto para la iteración

agustin.villena@gmail.com

Reglas para gestionar el timeboxing + alcance variable

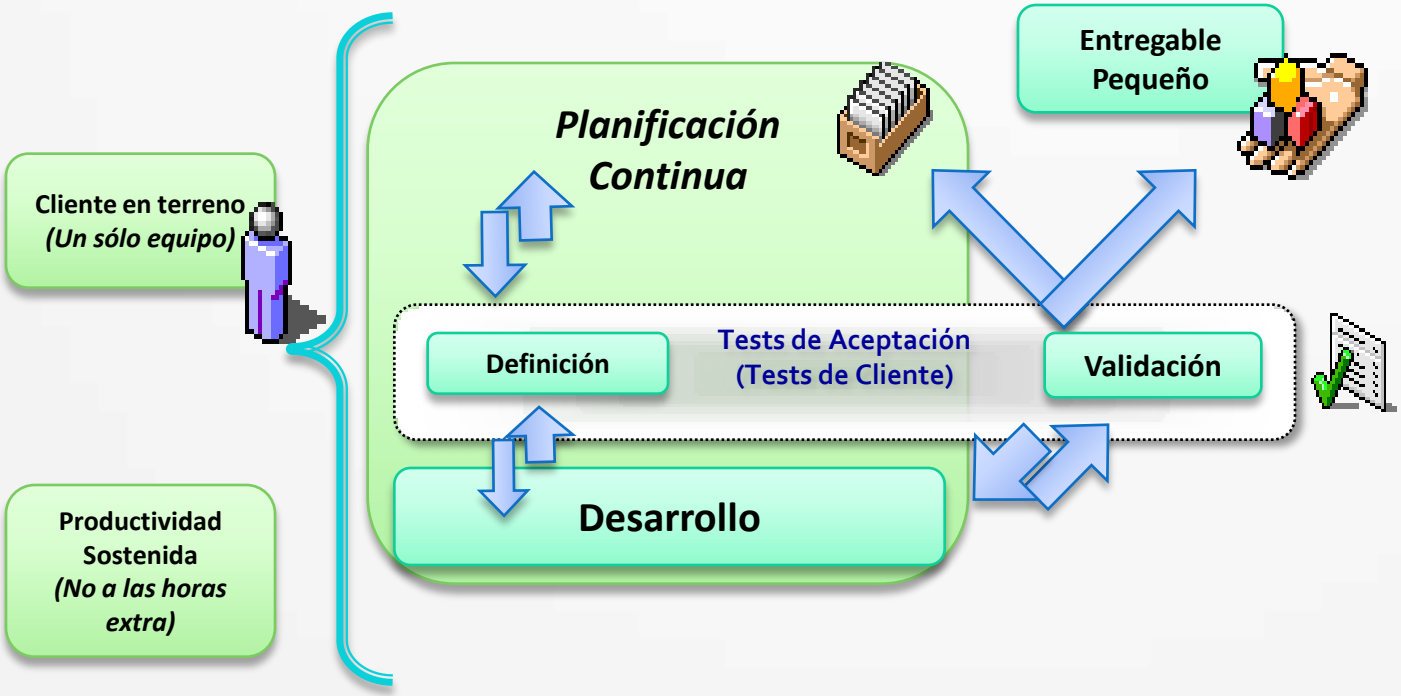
- El objetivo es que **todos adquieran la confianza** de que las metas del proyecto (de negocio y técnicas) **son alcanzables**

- Aquí el cliente **comparte** su responsabilidad sobre el proyecto con el desarrollador, en una relación de **sinergia**
  - (inspirado en el modelo Toyota de gestión)

	 Cliente	 Desarrollador
Desea maximizar	**Valor recibido** por cada semana de desarrollo	**Calidad del trabajo** realizado
Puede definir	**Qué será implementado**, y en **qué prioridad**, según las **necesidades de su negocio**	**Cuánto** se **estima que demorará** una tarea (idealmente)
Puede cambiar	**Funcionalidades solicitadas** por otras no implementadas de costo equivalente (canjear)	Sus **estimaciones** en base a nuevos descubrimientos

agustin.villena@gmail.com

Como funciona la gestión ágil



agustin.villena@gmail.com

Prácticas ágiles

- Cliente en terreno / Un sólo equipo /Real involucramiento del cliente
  - El Cliente es parte integral del equipo aportando su conocimiento del negocio, especificando requerimientos y validando funcionalidades
  - `http://c2.com/xp/OnsiteCustomer.html`
- Productividad Sostenida *(No a las horas extra)*
  - *"Un programador cansado no es más productivo que uno descansado"*
  - `http://c2.com/cgi/wiki?OverTime`
- Planificación Continua
  - Sigue las reglas de gestión del alcance variable
  - `http://c2.com/cgi/wiki?PlanningGame`
- Tests de Cliente/Funcional
  - Tests que ayudan a acotar el desarrollo y hacen objetivos los resultados
  - `http://c2.com/cgi/wiki?AcceptanceTest`
  - `http://fit.c2.com/wiki.cgi?CustomerTests`
- Entregables pequeños
  - Entregable que representa un incremento pequeño de funcionalidades valiosas y que se genera frecuentemente
  - `http://c2.com/cgi/wiki?SmallReleases`

agustin.villena@gmail.com

Modelo general de gestión

agustin.villena@gmail.com

Productos de un desarrollo de software

- **Software**
  - ejecutables, código fuente
- Documentación de **requerimientos y pruebas**
  - Contrato **dinámico** entre clientes y desarrolladores para definir los productos del proyecto
- Documentación de **diseño**
  - Toda aquella información requerida **por las personas** para registrar soluciones y orientar el trabajo técnico
    - Arquitectura
    - Clases / Modulos
    - Flujos
    - Interfaces de Usuario / Mapas de navegación
    - etc.

agustin.villena@gmail.com

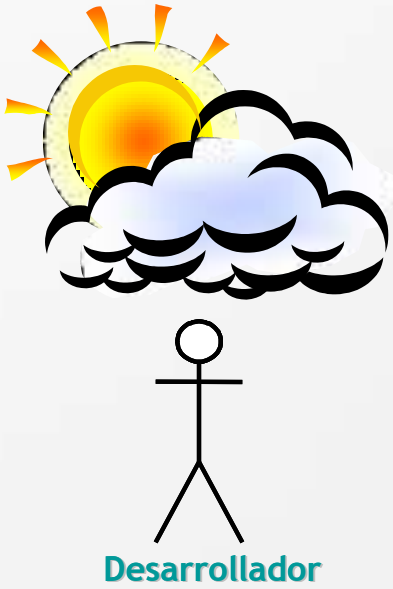
¿Qué se gana al Gestionar?

- El Proyecto
  - Se establece un **nivel suficiente** de **disciplina** en todas las **actividades de desarrollo**
  - Se **enriquecer la comunicación** entre los miembros de un equipo de trabajo (incluido el cliente) y con el resto del mundo
  - Se **retroalimenta** (aprender y mejorar) adecuadamente el plan de trabajo

agustin.villena@gmail.com

¿Y qué gana el desarrollador al apoyar la gestión?

- *"Llenar esa información (de gestión) es pérdida de tiempo, porque hay trabajo que hacer"*
  - Actitud típica del desarrollador
  - Y muchas veces, los sistemas de gestión le dan la razón, porque a él no le apoyan directamente su labor
- **Idea:** Integrar la gestión como una actividad que realmente le aporte valor a todos, incluyendo al desarrollo
- **Solución:** Darle al desarrollador información que le aclare:
  - Qué es lo que está hecho,
  - Qué es lo que ha avanzado el resto
  - Qué le queda por hacer, y cual es su importancia



agustin.villena@gmail.com

Algunas definiciones

Planificación

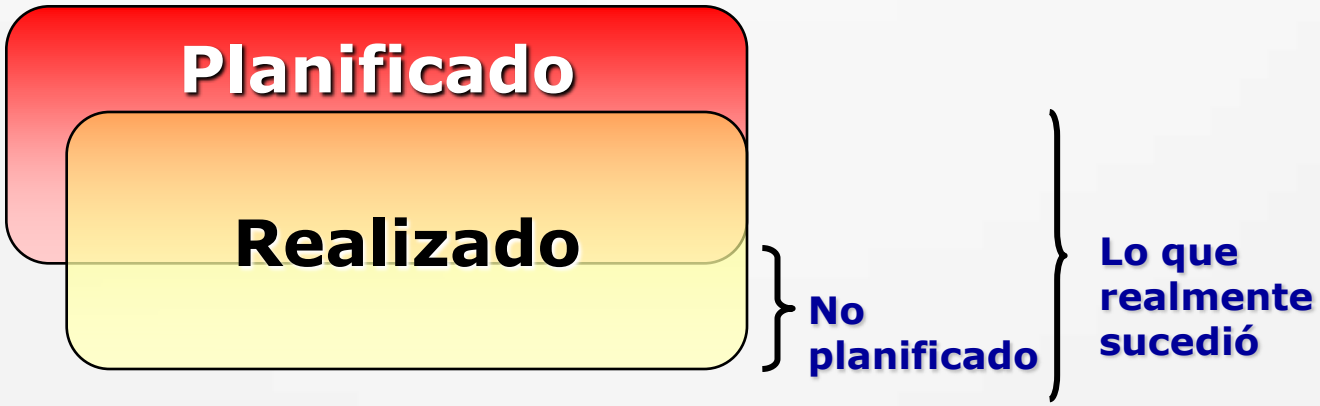
- *Planificar*
  - Es **regular un proceso productivo** conforme a un **plan** determinado
- *Plan*
  - Es una **organización** de **tareas** orientadas a lograr **funcionalidades**
- *Producto*
  - es un **logro apreciable de valor**
- *Tarea*
  - Es un **trabajo a realizar** en un **tiempo usando recursos** determinados para lograr una **producto**

agustin.villena@gmail.com

Algunas definiciones

Seguimiento y control

- Son las acciones orientadas a **recopilar** y **contrastar** información sobre los avances del proyecto, con el objetivo de
  - Proveer **visibilidad adecuada** a todas las **personas involucradas** en el proyecto acerca del **resto de actividades** que se estén desarrollando
  - **Retroalimentar** la planificación original



agustin.villena@gmail.com

Principios Inspiradores de una Gestión *Indolora*

- **Transparente:** una buena gestión no debe notarse
  - Debe estar integrada naturalmente en la labor diaria del proyecto
  - Simple y con poco esfuerzo (pocos minutos al día)

- **Valiosa:** permite **conocer** de manera precisa el avance actual de un proyecto
  - Avance = **generación de valor** lograda
  - **=> Poder tomar decisiones ágilmente**

- **Sencilla** y **barata** de reproducir y operar
  - Se utiliza una planilla de cálculo convencional

agustin.villena@gmail.com

Qué no queremos

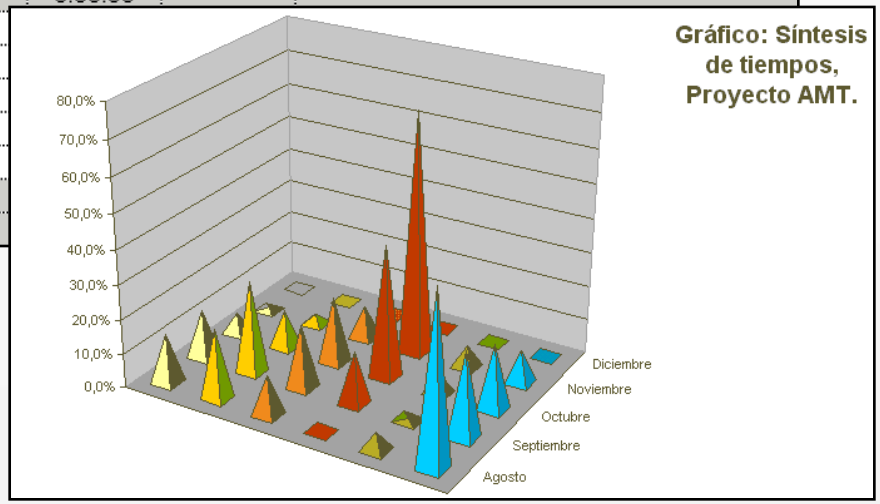
- Gastar tiempo en registrar métricas que no aporten real valor

agustin.villena@gmail.com

Qué no queremos
ejemplo de nuestro pasado

C3 =='OCT-mparedes'!C3+'OCT-ccespede'!C3+'OCT-afierro'!C3+'OCT-mortega'!C3+'OCT-franmuno'!C3+'OC robarrie'!C3

	Fecha		Total Tiempo Actividad						Observaciones
			Req.	Diseño	Document.	Construcc.	Pruebas	Gestión/Otros	
3	L	1			2:00:00	2:00:00			Ciclo 2/2.
4	M	2	1:50:00		0:30:00	3:00:00		6:50:00	
5	M	3			1:15:00	4:00:00			
6	J	4		2:00:00		8:20:00			Revision 2 Ciclo 1.
7	V	5			1:00:00	2:00:00		0:20:00	
8	S	6				15:30:00	1:00:00		
9	D	7			3:00:00	0:30:00	0:30:00		
10	L	8	1:00:00	3:00:00	5:00:00	8:00:00			
11	M	9		2:50:00		1:00:00			
12	M	10	2:00:00	4:00:00		2:00:00			
13	J	11	1:15:00						
14	V	12							
15	S	13				8:30:00			
16	D	14		3:00:00					

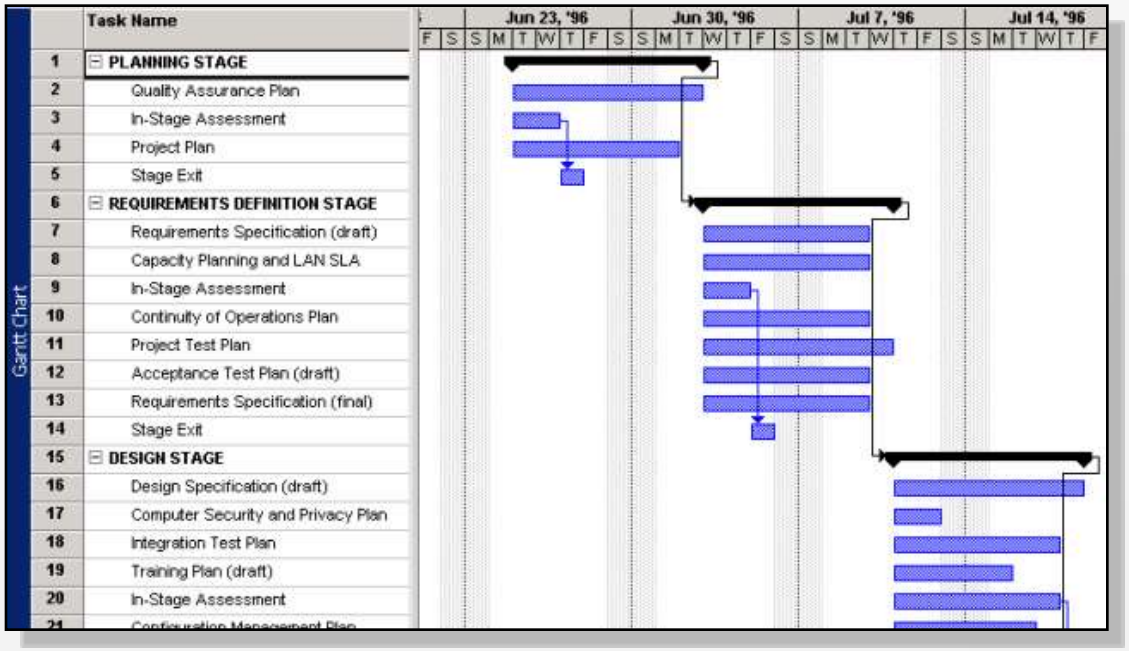


- Métricas que usamos anteriormente en CC61A

agustin.villena@gmail.com

Qué no queremos
Típica Carta Gantt

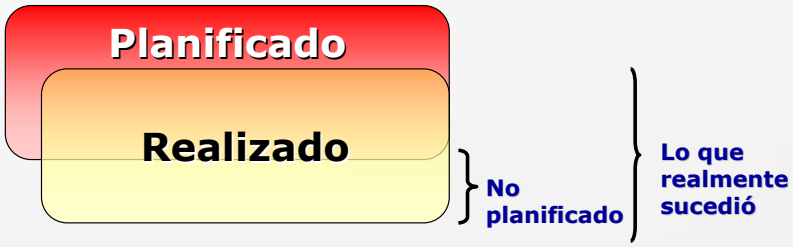
- Tareas que no apuntan **productos**
  - *Diseñar, plan de pruebas, saber requerimientos…*
- Manejar dependencias entre tareas que luego (casi seguro) van a cambiar



agustin.villena@gmail.com

Painless Tracking

- Herramienta evolucionada a lo largo del curso
  - Inspirada en Painless Schedules
- Componentes
  - **Plan**
    - Lo que se estima que se hará
  - **Tracking**
    - Registro de lo que realmente pasó
  - **Métricas y Reportes**



agustin.villena@gmail.com

Método
1. Preparación

- Definir un listado de **funcionalidades** (productos) a implementar en el software
  - Cada funcionalidad debe ser
    - **estimable** en horas *ideales* y
    - **verificable** mediante un algoritmo definido de validación (caso de prueba)
  - Si no se puede estimar o verificar, debe **especificarse más**
  - Debe formularse en lenguaje **entendible para el cliente**, de tal manera que muestre un **valor apreciable**

Criterio de definición de Funcionalidad:

Minimun Marketable Feature:

Aquella funcionalidad de menor tamaño que aporte un nuevo valor al cliente

agustin.villena@gmail.com

1. Preparación (cont.)

- Descomponer dichas funcionalidades en tareas
  - Cuyo tamaño debe ser **posible de estimar**
    - Si no se pudiese estimar, la tarea debe ser descompuesta en pedazos más pequeños
    - ¡Al descomponer, comenzamos a resolver el problema!
  - El costo en tiempo de implementar una tarea se expresa en horas ideales
    - Se asume total dedicación al desarrollo, sin imprevistos ni interrupciones
  - Regla heurística
    - Una tarea no debe durar más de dos días ideales

agustin.villena@gmail.com

Painless Tracking

1. Preparación : Registro

- Los datos se vuelcan en una planilla de cálculo

Funcionalidades	Prioridad	ID Tarea	Descripción Tarea	Orig. Est.	Ocup.	Rest. Est	Total Actual
Bugs Browser	1	Plug.Bugs.1	Algunos problemas que corrigir	7	8	0	8
CallBack	1	Plug.CallBack.1	Ultima funcion del plugin	3	5	0	5
Funcionalidades	2	Plug.func.1	Funcion abrir y about	4	1	0	1
Prompt hidden	2	Plug.Prompt.1	Escondar la clave	3	0	0	0
Installar LXR	3	AceptaDoc.2	Ver si es facil de usar	4	0	0	0
Integrar UI conAPI	1	Plug.Int.2	Implementar invocacion de firma	1	0	0	0
Bugs Mozilla	1	Plug.Bug.1	Ver si no se puede usar XPCOM en forma remota	2	5	0	5
Orden de los certificados NSS	1	Plug.Cert.1	Ordernar cert ante ponerlos en CCryptStore	5	0	0	0
Integrar UI con API	1	Plug.Int.1	Averiguar como invocar la criptoAPI desde el pug-in	3	0	0	0
Implementacion RSA PKCS #1	2	Plug.PKCS1.2	Implementacion	10	0,5	0	0,5
Validacion PKCS1 sin NSS	3	**Plug.PKC.1**	**Estudiar norma de PKCS1**	**20**	**4**	**1**	**5**
Interfaz Grafica como IE	2	Plug.UI.3	Interfaz de selección de certificado	4	0	0	0
Funcionalidades	2	Plug.func.2	Funcion infos	4	0	0	0
Pruebas de memoria con el archivo pruebas.cpp	1	Plug.Mem.1	Ver con MemoryTracker lo que esta pasando	2	0	0	0
Firmas multiples	1	Plug.Firma.1	Implementar las firmas multiples	10	0	0	0

agustin.villena@gmail.com

Painless Tracking

2. Actualización

- Se registra diariamente lo avanzado
  - En qué tarea se avanzó y cuanto se ocupó
  - Qué falta y cuanto se estima que falta (idealmente)
  - Si se trabajó en tareas no planificadas, se registra como “no planificada”

Fecha	ID	Ocup	Rest	Qué se hizo	Qué falta por hacer
20-Oct	Otros	2	1	Reorganizacion de carpetas / cambios en los Makefiles	Pruebas de Integración
20-Oct	Otros	1	0	Documentation en TeX de la reunion c	
20-Oct	e-mail	1	0	Ajuste de envíos de mails	-
21-Oct	obj.NSS.1	6	5	Es mas que un bug: differencia de semantica que induce varios cambios.	Ajustar código al nuevo contexto
21-Oct	Otros	1	0	Correcion de un bug en CStore::Update	
21-Oct	Otros	1	0	Reorganizacion de carpetas / cambios en los Makefiles	
22-Oct	obj.NSS.2	2	0	Ya habia estudiado un poquito el codigo.	
22-Oct	obj.NSS.1	4	0	El problema era la ambiguidad de la semantica de CERT_DestroyCertificate.	
22-Oct	Otros	2	0	Correcion de un bug en CCryptCertificate::Export (return true y no false)	

- La herramienta automáticamente actualiza
  - Tiempos ocupados por tarea
  - Tiempos restantes
  - Métricas interesantes

agustin.villena@gmail.com

Painless Tracking

3. Métricas interesante

Datos del proyecto	
Cantidad de Desarrolladores	5
Horas de trabajo por desarrollador por semana	16
Tareas Finalizadas	
Total Estimaciones	0,5
Total Ocupado	8
Velocidad relativa	**6%**
Planificabilidad	
Total ocupado en tareas no planificadas	0
Total de horas trabajadas	8
Disponibilidad	**100%**
Velocidad de Desarrollo Calculada	**6%**
Velocidad de Desarrollo Estimada	**50%**
Tiempo Estimado de Llegada	
Horas restantes planificadas	488,5
Tiempo Estimado de Arribo (horas)	977,0
Tiempo Estimado de Arribo **Corregido** (semanas)	12,21

Velocidad Relativa:
Proporción **real** entre la suma de horas estimadas y las horas realmente ocupadas. Ej: "50%" implica que se avanza a la mitad de la rapidez estimada

Proporción entre las horas **planificadas** y las realmente ocupadas

Velocidad Relativa multiplicada por Disponibilidad

Velocidad realtiva futura que el equipo estima

Considera un desarrollador

Estimación de semanas que quedan para terminar, considerando numero de desarrolladores y jornada semanal

agustin.villena@gmail.com

Painless Tracking
Beneficios

- Al avanzar el proyecto
  - Se va **aprendiendo de la experiencia**
  - Y se **mejoran** nuevas estimaciones
- De esta forma, la planificación siempre servirá para
  - Negociar(Recortar/canjear) características
  - O alargar la estimación

	1	2	3	4	5	6
1	**Feature**	**Task**	**Priority**	**Orig Est**	**Curr Est**	**Elapsed**
2	Spell Checker	Add Menu Item	1	12	8	8
3	Spell Checker	Main Dialog	1	8	12	8

agustin.villena@gmail.com

Painless Tracking
Reportes

- La Painless trae diversos reportes. El más usado es el de **avance**

Iteración	Prioridad	Funcionalidad	Tarea	Último "Que Falta"	Datos: Ocupado Actualmente	Restante Estimado	Promedio de % Avance
1					25	1	96%
	Fundamental				19	1	95%
		Crear ambiente desarrollo			4	0	100%
			Instalacion y configuracion ambiente desarrollo		4	0	100%
		Mostrar la hora en formato digital y mostrar la hora en formato análogo			14	0	100%
			Implementación clase Pulso		7	0	100%
			Implementación clase Reloj		7	0	100%
		Validar software con gerente			1	1	50%
			validación de software	Documentar	1	1	50%
	Importante				6	0	100%
		Mostrar la hora en formato digital 24 horas			4	0	100%
			Implementación clase TiempoDigital24		4	0	100%
		Mostrar la hora en formato digital AM/PM			2	0	100%
			Implementación clase TiempoDigitalAMPM		2	0	100%
2					0	11	0%
	Fundamental				0	11	0%
		Implementar cronómetro			0	10	0%
			implementacion clase TimeCronometro		0	4	0%
			implementacion metodo acceptSignal		0	3	0%
			implementacion metodo ignoreSignal		0	3	0%
			implementacion metodo toZero SIN ESTIMAR		0	0	#¡DIV/0!
		Validar software con gerente			0	1	0%
			Validar funcionalidades de la 2da iteracion		0	1	0%
Total general					25	12	68%

agustin.villena@gmail.com

Gestión de Riesgos

Recordar:
incertidumbre => riesgo

- Orientado a **evaluar** y **prevenir** periódicamente **los problemas y riesgos** que enfrenta el equipo, ayudando a definir medidas de contingencia que minimicen el impacto

- Y así **retroalimentar** la planificación

Descripción de los Riesgos

Ponderación de Riesgos

Nº Reporte:		
Fecha Reporte:		

Nº	Nombre Riesgo	Breve descripción	Prob.	Impact.	Evaluación
1					0,0
2					0,0
3					0,0
4					0,0
5					0,0
6					0,0

Nº	Breve Descripción de Medidas de Contingencia	Riesgos atacados	Responsable	Total de riesgos:	0
1				RISCON:	0,0
2					
3					
4					
5					

Descripción de medidas de contingencia

agustin.villena@gmail.c

Gestión de Riesgos
Ejemplo

Nº Reporte:	5
Fecha Reporte:	16-Ago-04

Nº	Nombre del riesgo	Breve descripción	Prob.	Impact.	Evaluación
1	Disponibilidad de Eqipos	El cliente no posee suficientes PCs para el grupo	4,0	3,0	12,0
2	Desconocimiento de J2EE	La pltaforma no es dominada por la mayoría del grupo	5,0	4,0	20,0

Nº	Breve Descripción de Medidas de Contingencia	Riesgos atacados	Responsable	Total de riesgos:	2
1	Capacitación en J2EE	2	Juan Pérez	RISCON:	7,9
2	Uso de computadores de a dos personas (Pair Programming)	1	Grupal		
3	Capacitación en Pair programming	1	Agustín Moya		
4					
5					

- Las medidas de contingencia
  - Son acciones **concretas** con su respectivo **responsable**
  - Retroalimentan la planificación

agustin.villena@gmail.com

(Algunos) Recursos
Enlaces de Internet

- Extreme programming
  - www.extremeprogramming.org
- WikiWiki entry point for XP
  - http://c2.com/cgi/wiki?ExtremeProgrammingRoadmap
- Scrum Development Process
  - www.controlchaos.org
- Gestión Ágil de Software
  - www.agilemanagement.net
- Lean Software Development
  - www.poppendieck.com
- Agile Alliance
  - www.agilealliance.org
- ChileAgil
  - www.chileagil.cl

agustin.villena@gmail.com