Entrenamiento SCRUM

Softtek®

Restricciones

Nombre del documento:
Clasificación de la Información: INTERNO
Restricciones

› Los contenidos de este documento son propiedad de Softtek y son confidenciales. Queda estrictamente prohibido cualquier reproducción total o parcial sin la autorización escrita por parte de Softtek.

› Este documento está sujeto a cambios. Los comentarios, correcciones o dudas deberán ser enviados al autor.

Audiencia	Propósito
Scrum Masters, Team Members, Service & Delivery Managers, ADM, BRM, PreVenta, Líderes y Directores de Desarrollo.	Contar con material que sirva de soporte y guía para la provision de servicios de desarrollo ágil bajo el marco SCRUM.

Tabla de Revisión

› La siguiente tabla enlista las revisiones realizadas a este documento. Debe utilizarse para describir los cambios y adiciones cada vez que este documento vuelva a ser publicado. La descripción debe ser detallada e incluir el nombre de quien solicita los cambios.

Núm. de versión	Fecha de versión	Tipo de cambios	Dueño / Autor	Fecha de revisión / Expiración
0.5	24-feb-2016	Draft Inicial	LGH / SMES	31-dic-2016
1.0	08-mar-2016	Ajustes varios por retro de Stakeholders	LGH/SMES	31-dic-2016
1.5	16-may-2016	Alineación a Scrum Alliance	LGH/SMES	31-dic-2016
2.0	27-may-2016	Reestructura completa con retroalimentación de los equipos SCRUM desde las trincheras	LGH/SMES	31-dic-2016
3.0	26-Oct-2016	Actualización de SCURM Empresarial y adición de ejerciocios prácticos	AIGH	26-Oct-2016

Contenido

DÍA 1

1. El desarrollo ágil de software
   - ✓ Cambio de paradigma
   - ✓ Scrum
   - ✓ Valores y pilares de Scrum
2. El equipo Scrum
   - ✓ Roles y Responsabilidades
3. Las herramientas básicas
   - ✓ Tableros de seguimiento y gestión
   - ✓ Métrica de Velocidad

DÍA 2

1. El Product Backlog
   - ✓ Características del PB
   - ✓ Historias de Usuario
2. La estimación de puntos de historia
   - ✓ Cosmic y Planning Poker
3. Simulación de Proyecto Scrum
4. Modelo del servicio de desarrollo ágil escalado

El desarrollo ágil de software

Softtek®

Value THROUGH TECHNOLOGY

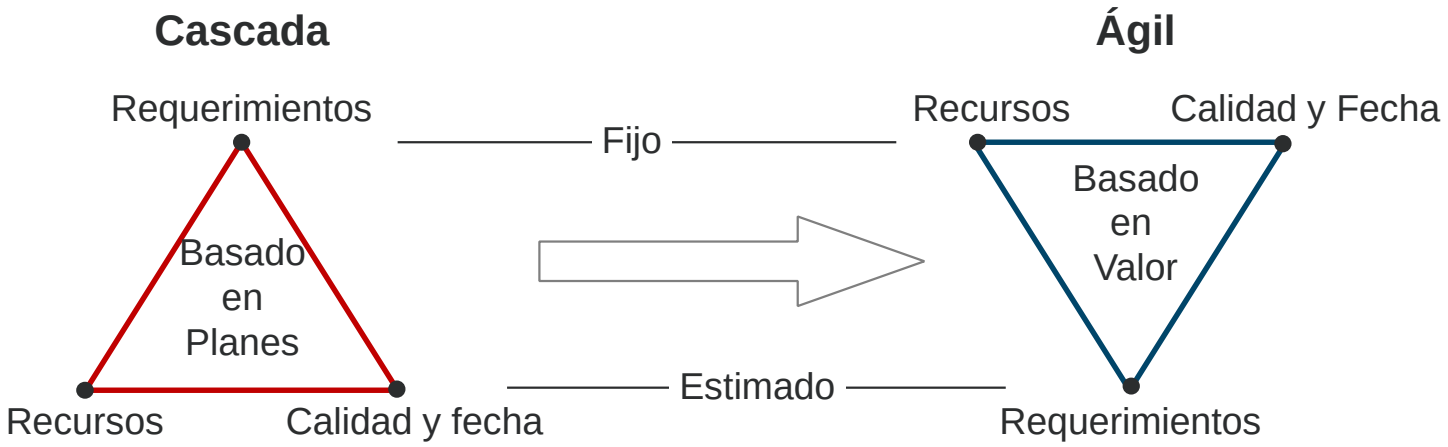
El desarrollo ágil de software
Cambio de Paradigma

Problemas de procesos definidos:

- El progreso es supervisado por el “porcentaje de completado”
- El cliente sabe lo que quiere desde un inicio
- Los cambios sobre la marcha deben ser minimizados lo más posible
- Debemos iniciar con requerimientos precisos, arquitectura y un plan de trabajo
- Los planes de trabajo a largo plazo pueden ser exactos basados en un WBS
- Cada problema puede ser reparado a través de encontrar el proceso adecuado.
- Las evaluaciones individuales miden la efectividad del equipo
- Los desarrolladores y el cliente pueden establecer una comunicación efectiva a través de documentos
- Haz la integración y las pruebas de usuario después de que finalices el trabajo de desarrollo
- El PM estima y asigna las tareas a la gente del equipo

El desarrollo ágil de software

Cambio de Paradigma



Beneficios:

- Reducción del riesgo de producir el producto equivocado
- Time-to-Market reducido
- Flexibilidad mejorada para adaptarse a cambios tardíos
- Visibilidad incrementada del producto
- Clientes satisfechos y alta tasa de éxito

Se ajusta mejor cuando:

- El producto o aplicación es definida solo a alto nivel
- Los requerimientos son desconocidos por adelantado → común para programas o productos largos
- Necesidad de incorporar nuevas características de manera rápida
- Existencia de muchos especialistas (incluyendo clientes y actores externos)
- Enfoque al cliente/experiencia del usuario (UX)

El desarrollo ágil de software

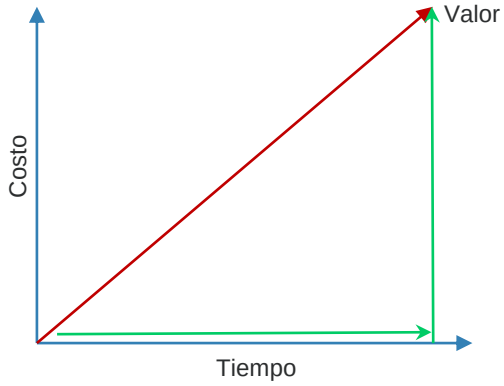
Cambio de Paradigma

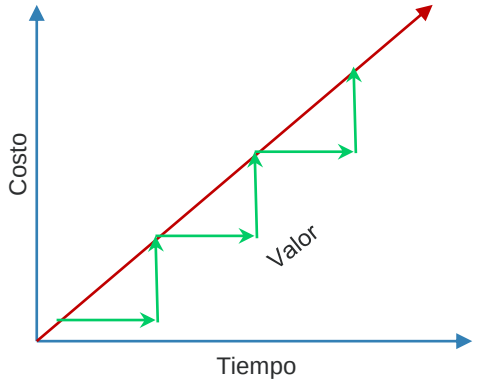
Cascada =
Bien desde el inicio,
TODO el valor es entrado
en una sola vez

Ágil = Iterativo, Incremental

ITERATIVO = No trates de obtener todo bien desde el inicio

INCREMENTAL = No construyas TODO de un solo jalón





El desarrollo ágil de software
Manifiesto

"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"

Mayor detalle en: http://agilemanifesto.org/

El desarrollo ágil de software
Principios (12)

Nuestra mayor prioridad es satisfacer al cliente mediante la entrega temprana y continua de software con valor.	Aceptamos que los requisitos cambien, incluso en etapas tardías del desarrollo. Los procesos Ágiles aprovechan el cambio para proporcionar ventaja competitiva al cliente.	**Entregamos software funcional frecuentemente, entre dos semanas y dos meses, con referencia al periodo de tiempo más corto posible.**	Los responsables de negocio y los desarrolladores trabajamos juntos de forma cotidiana durante todo el proyecto.
Los proyectos se desarrollan en torno a individuos motivados. Hay que darles el entorno y el apoyo que necesitan, y confiarles la ejecución del trabajo.	El método más eficiente y efectivo de comunicar información al equipo de desarrollo y entre sus miembros es la conversación cara a cara.	**El software funcionando es la medida principal de progreso.**	Los procesos Ágiles promueven el desarrollo sostenible. Los promotores, desarrolladores y usuarios debemos ser capaces de mantener un ritmo constante de forma indefinida.
La atención continua a la excelencia técnica y al buen diseño mejora la Agilidad.	La simplicidad, o el arte de maximizar la cantidad de trabajo no realizado, es esencial.	**Las mejores arquitecturas, requisitos y diseños emergen de equipos auto-organizados.**	A intervalos regulares el equipo reflexiona sobre cómo ser más efectivo para a continuación ajustar y perfeccionar su comportamiento en consecuencia.

Mayor detalle en: http://agilemanifesto.org/principles.html

El desarrollo ágil de software
¿Qué es Scrum?

Softtek®

"Marco de trabajo general, basado en el control empírico, para entregar resultados con el máximo valor de negocio, en el menor tiempo posible, a través de un equipo Scrum con auto-organización, auto-aprendizaje y auto-mejora.

El desarrollo ágil de software
Pilares de la teoría Scrum (Control Empírico)

Transparencia	Los aspectos significativos del proceso deben ser **visibles** para aquellos que son responsables del resultado.
Inspección	El equipo Scrum debe inspeccionar frecuentemente los artefactos construidos y el avance hacia el objetivo, para **detectar variaciones**.
Adaptación	Si el equipo Scrum determina que uno o más aspectos de un proceso se desvían de límites aceptables, y que el producto resultante no será aceptable, el **proceso** o el **producto** que se está construyendo debe ser **ajustado**.

El desarrollo ágil de software
Valores SCRUM

Foco	Porque **nos enfocamos en sólo unas pocas cosas a la vez**, trabajamos bien juntos y producimos un resultado excelente. De este modo logramos entregar ítems valiosos antes.
Coraje	Porque no estamos solos, nos sentimos apoyados y **tenemos más recursos a nuestra disposición**. Esto nos da el coraje para enfrentar desafíos más grandes.
Apertura	Durante el trabajo en conjunto **expresamos cotidianamente cómo nos va y qué problemas encontramos**. Aprendemos que es bueno manifestar las preocupaciones, para que éstas puedan ser tomadas en cuenta.
Compromiso	Porque **tenemos gran control** sobre nuestro destino, nos comprometemos más al éxito.
Respeto	A medida que **trabajamos juntos, compartiendo éxitos y fracasos**, llegamos a respetarnos los unos a los otros, y a ayudarnos mutuamente a convertirnos en merecedores de respeto.

El desarrollo ágil de software
¿Scrum en el Software causa Agilidad?

- El Software manufacturado con métodos tradicionales, no estaban produciendo resultados deseados

- Satisfacer las necesidades de negocio para entregar desarrollo de software con flexibilidad, tan rápido como sea posible, con alta calidad, y con una mínima cantidad de recursos….

- La mayoría de los proyectos ágiles utilizan Scrum: 80%

- Miles de proyectos exitosos a través del mundo: Google, Microsoft, Yahoo, SalesForce, Ericson, Mercer, Fidelity, Siemens, BCBS, CIGNA, Mass Mutual, Federal Reserve, SoftServe, Walgreens, Abbott, JP Morgan Chase, State-Farm, Walmart, Grainger, Caterpillar, Apple, Cisco, HP, Wells Fargo, Telefonica, Accenture, Softtek, etc.

El equipo Scrum

Softtek®

Value THROUGH TECHNOLOGY

El equipo *Scrum*
Responsabilidades

Product Owner

- **Única** persona responsable de delinear el producto más valioso posible, gestionando el flujo de trabajo hacia el equipo. Lo lleva a cabo seleccionando y refinando ítems del *Product Backlog* (**PBI**s por sus siglas en inglés).
- **Representar a los Stakeholders**. Mantener el *Product Backlog* y asegurase que todos sepan qué hay en él y cuáles son las **prioridades**. Definir las Historias de Usuario y asignar la prioridad de atención.
- Es **responsable del retorno de inversión** (ROI) y del éxito o fracaso del proyecto.

Development Team

- Entregar un Incremento de producto con **valor al negocio**, que potencialmente se pueda poner en producción (“Terminado”), al final de cada Sprint.
- Determinar la cantidad de **Historias de Usuario que se van a atender** en el Sprint con base en la métrica de velocidad del equipo.

Scrum Master

- **Experto** en el marco Scrum, **dueño del proceso**, **líder servicial**, además, **entrenador** y **facilitador** responsable de asegurarse que el equipo Scrum trabaja ajustándose a la teoría, prácticas y reglas de Scrum.
- Llevar el control de la métrica de **velocidad** del equipo
- Llevar el **control del presupuesto** de acuerdo al alcance acordado con el Product Owner.

El equipo *Scrum*
Principales actividades

Product Owner

- **Provee** la visión y panorama general del producto.
- **Gestiona y Prioriza** el *Product Backlog*
- **Aprueba** los Product Backlog Items (**PBI**s)
- **Siempre decide** en qué trabajar y qué liberar
- **Escribe** las Épicas e Historias de Usuario
- **Participa** en las reuniones de *Scrum,* **excepto**: *Daily Scrum* que será sólo a petición del equipo de desarrollo y *retrospective* en la que no se recomienda su participación.

Development Team

- **Construye y se compromete** a entregar *Software funcionando* **de acuerdo a la definición de "Terminado" (DDT)**
- **Proporciona** (siempre) **estimaciones** para todo el trabajo
- **Decide** la cantidad de **trabajo a realizar** en el Sprint. Cuántas Historias de Usuario se pueden realizar y cómo (tareas) lo van a llevar a cabo
- Se **auto-organiza** para realizar su trabajo. Todo el mundo colabora y comparte conocimientos
- **Escribe** los Spikes requeridos para cumplir el objetivo del Sprint y los comparte al Product Owner para que se integren al Product Backlog

Scrum Master

- **Programa** todos los eventos, todas las invitaciones, todas las agendas
- **Facilita** los tiempos de cada evento, se asegura se generen los entregables del mismo
- **Resuelve** problemas e impedimentos del equipo
- **Elimina** los obstáculos al avance del equipo
- **Fomenta** y **facilita** la auto-organización del equipo, asegurándose de la adherencia al marco de trabajo *Scrum*

El equipo Scrum
Requisitos para hacer factible utilizar Scrum

Indispensables	*Product Owner* con **dominio en el negocio y autoridad** para la toma de decisiones
	Disponibilidad del *Product Owner* al 100% (físicamente sólo durante *Release/Sprint Planning* y *Sprint Review*)
	Tamaño máximo del equipo de desarrollo entre 6 y 9 integrantes **dedicados de tiempo completo**
	Administración de la configuración con al menos un **ambiente dedicado** para desarrollo y pruebas
Requeridos	*Product Owner* y *Scrum Master* **empoderados** para gestionar y resolver dependencias con otros equipos/áreas
	Equipo **multidisciplinario** con dominio de la tecnología y de la aplicación, que cuente con el mismo software y permisos para facilitar el que pueda tomar cualquier tarea del Sprint Backlog.
	Disponibilidad y activa participación de arquitectos e Ingenieros de Software responsables de los aplicativos
Deseables	Equipo Scrum trabajando en el **mismo espacio físico** para maximizar la comunicación y la colaboración
	Product Owner y Equipo de Desarrollo **entrenados en el marco *Scrum*** para usar un mismo lenguaje y tener las misma bases que faciliten la comunicación y maximicen la colaboración
	Mantener el equipo a lo largo del tiempo para capitalizar **ganancias de productividad**
	Cada integrante del equipo de desarrollo esté dispuesto a compartir su conocimiento y experiencia con el resto del equipo para maximizar la **velocidad** del mismo

Las herramientas básicas

Softtek®

Value THROUGH TECHNOLOGY

Las herramientas básicas
Para la visibilidad, control y seguimiento

Las herramientas básicas
El tablero SCRUM

El Tablero Scrum es un sistema de información que utiliza el equipo *Scrum* para controlar y dar **visibilidad** de todas las tareas para el desarrollo de software identificadas en el ***Sprint***. Debe incluir las tareas del backlog no seleccionadas, **pendientes del Sprint**, **en proceso** y **terminadas**.

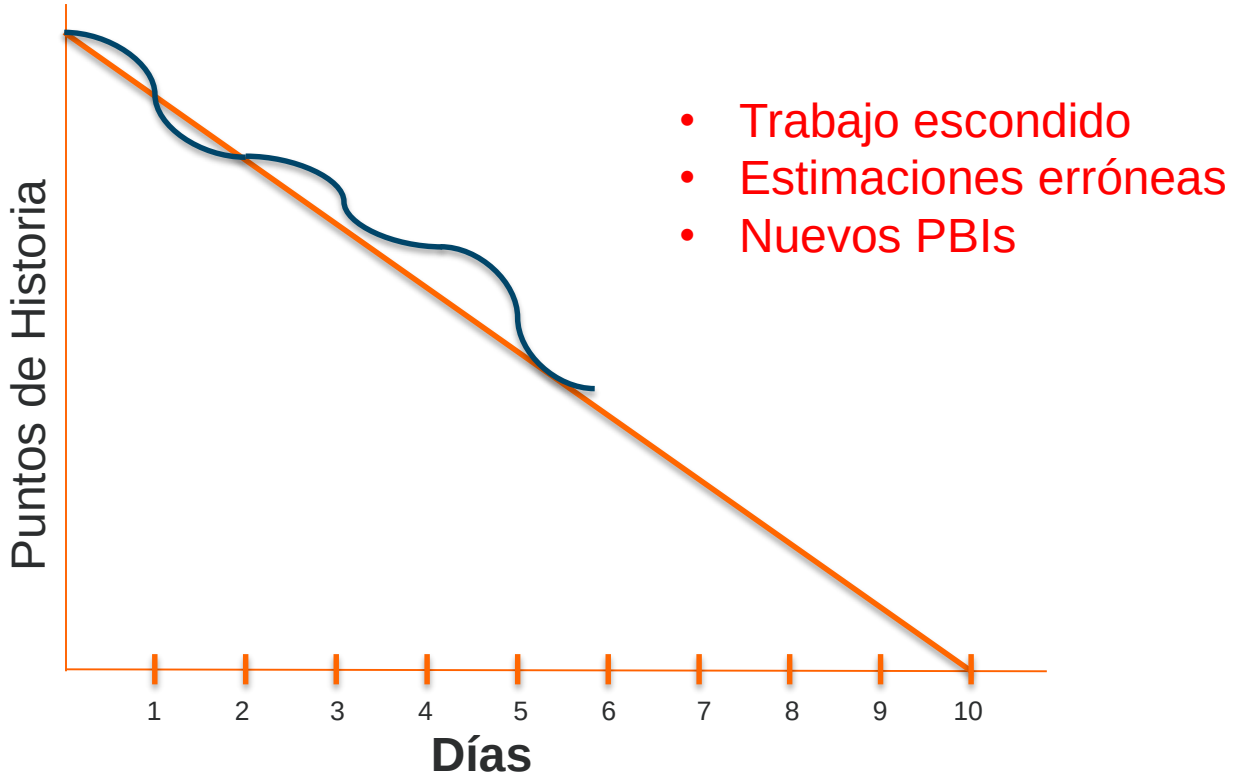
Valor del Negocio

1) Fin del plan
2) Pasa DDT
3) Aprobadas por el PO

No seleccionadas	Seleccionadas Sprint Backlog	En Proceso	Terminadas	
PBIs: US6 US7…. USN	US1 US2 US4	US3	US5	Sprint 1 Incremento del Producto
	DE1 USN			Sprint 2

Las herramientas básicas
Burndown Chart del ***Sprint***

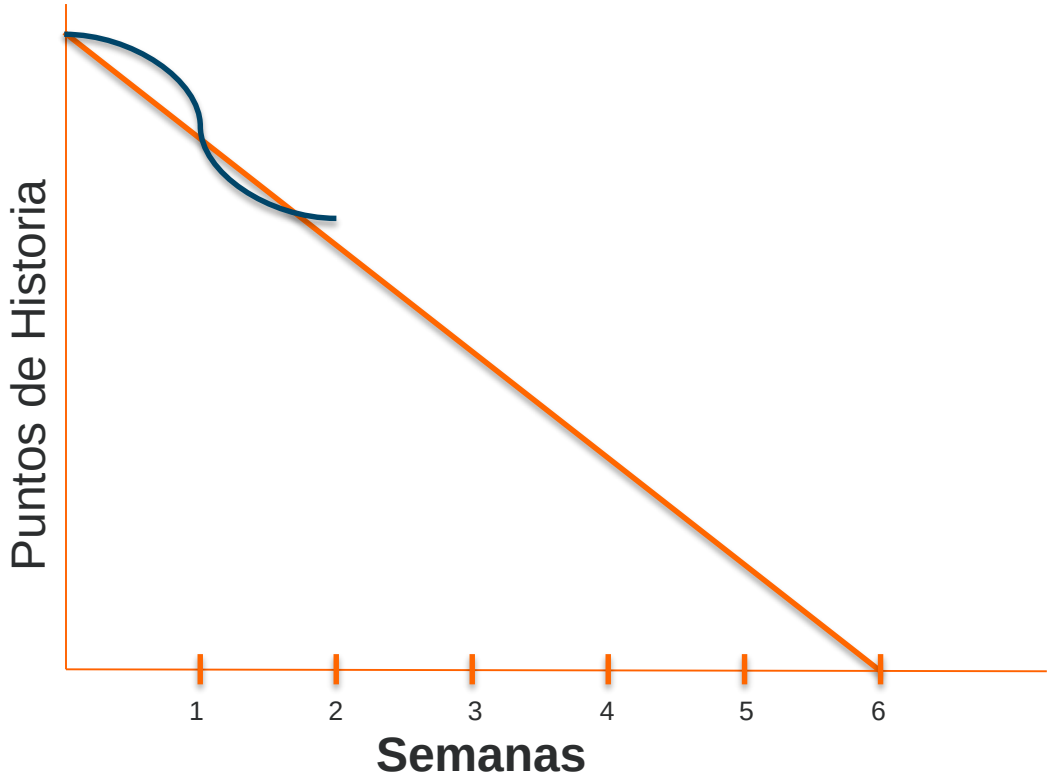
El Burndown chart es un gráfico que utiliza el equipo Scrum para tener **transparencia** y dar **visibilidad** de todos los puntos de historia cubiertos (“quemados”) en el desarrollo de software durante el **Sprint**. El eje **X** representa los días que dura el *Sprint* y el eje **Y** los puntos de historia a “quemar” durante el mismo. Se actualiza en el día a día, normalmente después del *Daily Scrum*.



https://www.scrumalliance.org/community/articles/2013/august/burn-down-chart-%E2%80%93-an-effective-planning-and-tracki

Las herramientas básicas
Burndown Chart del ***Release***

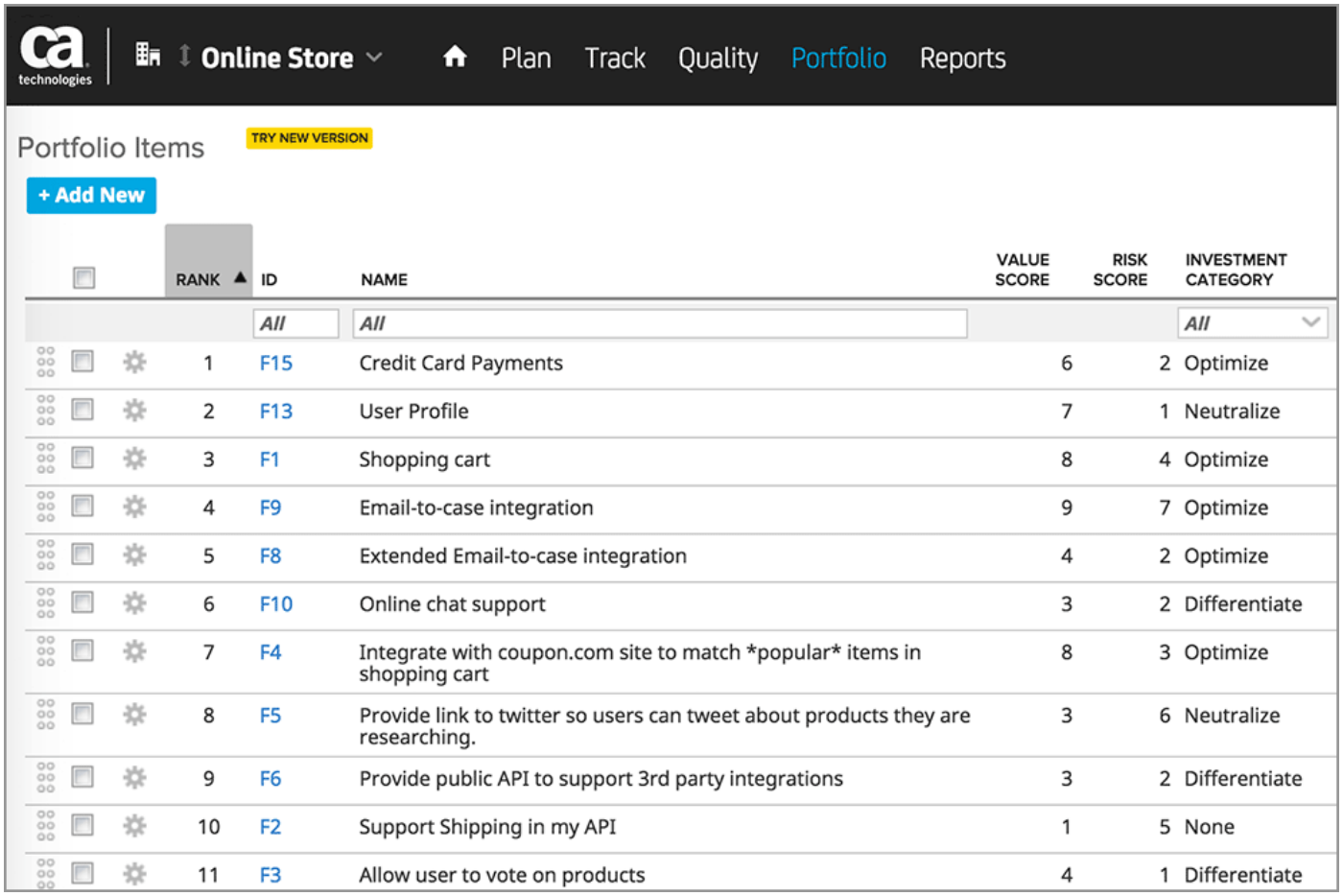
Le sirve al equipo Scrum para dar **visibilidad** del avance logrado en el *Release*. El eje X representa las semanas planeadas para el *Release* y el eje Y los puntos de historia cubiertos “quemados”. Se actualiza semanalmente.



https://www.scrumalliance.org/community/articles/2013/august/burn-down-chart-%E2%80%93-an-effective-planning-and-tracki

Las herramientas básicas
CA Agile Central

Le sirve al equipo Scrum para gestionar las actividades de los Sprint desde nivel US hasta nivel programa.



RANK	ID	NAME	VALUE SCORE	RISK SCORE	INVESTMENT CATEGORY
	All	All			All
1	F15	Credit Card Payments	6	2	Optimize
2	F13	User Profile	7	1	Neutralize
3	F1	Shopping cart	8	4	Optimize
4	F9	Email-to-case integration	9	7	Optimize
5	F8	Extended Email-to-case integration	4	2	Optimize
6	F10	Online chat support	3	2	Differentiate
7	F4	Integrate with coupon.com site to match *popular* items in shopping cart	8	3	Optimize
8	F5	Provide link to twitter so users can tweet about products they are researching.	3	6	Neutralize
9	F6	Provide public API to support 3rd party integrations	3	2	Differentiate
10	F2	Support Shipping in my API	1	5	None
11	F3	Allow user to vote on products	4	1	Differentiate

Las herramientas básicas
Elementos de Scrum

Elementos de SCRUM	
Artefactos	Product Backlog Sprint Backlog Sprint Burndown chart Tablero Scrum
Roles	Product Owner (PO) Scrum Master (CSM) Equipo Scrum
Cajas de tiempo	Sprint Día
Eventos	Juntas de Backlog, Planeación de Sprint, Scrum diario, Revisión de Sprint, Retrospectiva y Refinamiento
Ingeniería	Capas de pruebas, integración continua
Social/Ambiente	Valores Scrum, Pilares Scrum, manifiesto ágil, espacio de trabajo abierto, PR espontáneos, coaching/mentoreo.

Las herramientas básicas
El Proceso SCRUM

Táctica	Responsable	Actividades
Visión	PO	Proveer el panorama general del producto a desarrollar
Product Backlog Inicial	PO → Equipo →	Definir y Priorizar Provee PBIs no funcionales y estima tamaño
Planeación del Sprint (tamaño de 1, 2, 3 o 4 semanas) – 1hr/sem	PO, CSM, Equipo	Parte 1: ¿Decidir cuanto trabajo realizar? Parte 2: ¿Cómo? Voluntarios, plan y estimaciones.
Ejecución del Sprint. Scrum Diario (15 minutos, mismo lugar y hora, no celulares ni laptops)	CSM, Equipo	Descubrimiento de dependencias y colaboraciones, resolución del camino crítico. - ¿Qué hice? - ¿Qué haré? - Dependencias y/o asuntos
Revisión del Sprint – 1hr/sem	Equipo a PO	¿Qué fue hecho? DEMO + explicación
Retrospectiva – 1hr/sem	CSM, Equipo	¿Qué hicimos bien? ¿Qué podemos mejorar?
Refinamiento – 1hr	PO, CSM, Equipo	¿Hacia donde vamos ahora? - Visión

Las herramientas básicas
Velocidad (de un equipo de desarrollo ágil)

- La Velocidad de una célula de desarrollo ágil se calcula:
  - Al preguntar al equipo realizar una cierta cantidad de Puntos de Historia en un Sprint
  - Medir cuántos Puntos de Historia el equipo realmente terminó

  … esa es la Velocidad del equipo en un Sprint dado

- La medición de la velocidad de varios Sprints nos da una visibilidad de la capacidad de desarrollo futura del equipo Scrum

- El promedio de las mediciones de las velocidades de varios equipos durante varios Sprints en el mismo ambiente: tecnológico, miembros del equipo, cliente, etc… es la velocidad de desarrollo del programa

- **Tip**: Mide la velocidad promedio después del 2do Sprint…. Los primeros 2 sprints son solo para ganar estabilidad.

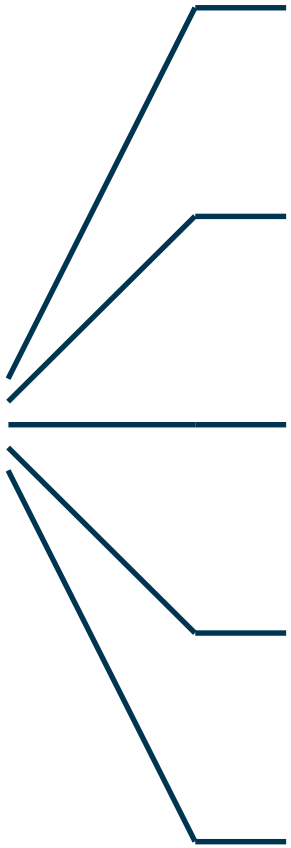
El *Product Backlog*

Softtek®

Value THROUGH TECHNOLOGY

El *Product Backlog*
Lo que debe incluir

El ***Product Backlog*** debe incluir <u>al menos</u> la siguiente información



- **Identificar de Épica** para cada PBI
- **Identificador único** de cada PBI (Product Backlog Item)
- **Descripción corta** (3 a 10 palabras) de cada PBI.
- **Prioridad** (grado de importancia para el negocio) de cada PBI.
- **Estimación Inicial** de cada PBI, hecha por el equipo de desarrollo.

El *Product Backlog*
Sus Características

El ***Product Backlog*** es una lista priorizada de PBIs (Product Backlog Items): Historias de usuario, requerimientos, funcionalidad o ideas para el producto, en la terminología del cliente de negocio.

Es indispensable para *Scrum* ya que es la única fuente posible de requerimientos. Todas y cada una de las tareas que lleva a cabo el equipo de desarrollo, se deben derivar de un PBI.

Aquellos PBIs que sean seleccionadas para el *Sprint,* deberán ser depurados y clarificados por el *Product Owner,* al equipo de desarrollo, para su atención.

El equipo de desarrollo puede identificar historias de usuario NO funcionales y solicitar al Product Owner que se integren al Product Backlog y se prioricen, para su atención.

Épicas e Historias de Usuario (Una implementación de los PBIs)

Historias de Usuario

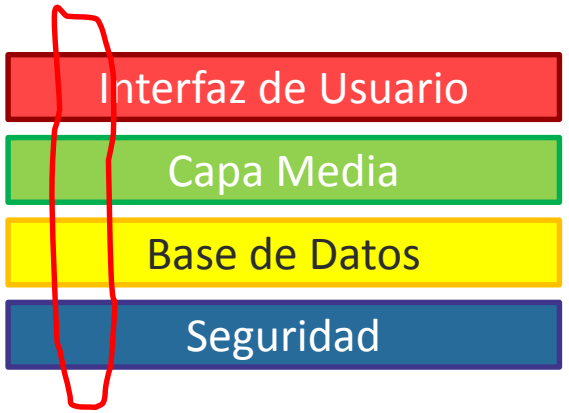
- **¿Qué son? ¿Cómo se ven?**
  - **Fácil, pequeña, eficiente, abreviada y telescópica manera de hacer requerimientos.**
  - Una Historia de Usuario es una “promesa de tener una conversación”
  - Formato simple: 3C = Carta, Conversación y Confirmación (UATs)
  - Reemplazan documentos largos con pequeños documentos y conversaciones !!!
  - Permiten una gradual evolución…. Qué, tamaño, prioridad, UATs, Arquitectura, Diseño, cosas, etc.
  - Unidades auto-trazeables
  - Propiedades y características INVEST

- **¿Por qué utilizarlas?**
  - Son más humanas que los Casos de Uso,

  o requerimientos funcionales!

- **Tipos:**
  - **Regulares – Unidad ejecutable y liberable**
  - **Épicas – Historias largas**
  - **Temas – Categorías de historias de usuario**

Corte vertical de la arquitectura. Es una unidad ejecutable mínima que será integrada a incrementos de funcionalidad con el potencial de ser liberada a Producción.



Historias de Usuario

Las **Historias de Usuario** deben incluir al menos la siguiente información

Identificador único de la Historia de Usuario

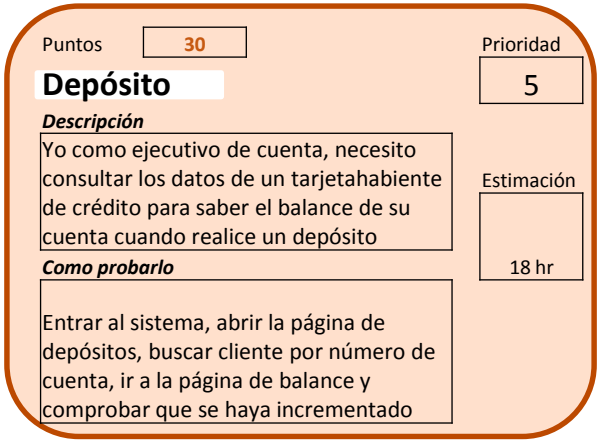
Nombre de la Historia

Descripción de la funcionalidad atómica de lo que el usuario quiere lograr mediante la entrega de valor al negocio, en formato canónico

Criterios de aceptación (cómo probar cada Historia de Usuario) en formato canónico

Puntos de historia asignados por el equipo

Estimación en horas o días de lo que el equipo consideró



Épicas e Historia de Usuario
Descripción de Historia de Usuario

Historia de Usuario	Es la descripción de la funcionalidad atómica de lo que el Product Owner quiere lograr mediante la entrega de valor al negocio. Debe incluir la forma en la que probará esa funcionalidad (criterios de aceptación).
Formato Canónico Historia	**Yo como** <Rol>, **necesito** <Funcionalidad> **que provee** <valor de negocio>
Formato Canónico Criterio de Aceptación	**Dado que** <contexto>, **cuando** <acción> **entonces** <resultado esperado> UATs == Definición de Terminado (DDT)
Características INVEST que debe cumplir	**I**ndependent, **N**egotiable, **V**aluable, **E**stimable, **S**mall y **T**estable
Importancia para el Negocio (Usar Modelo Kano)	**Must Have** – El Release no puedo ir sin ella **Want to Have** – El Release no puede ir sin ella pero queremos incluirla Nice to Have – Sería bueno pero no la necesitamos o queremos tanto

Mayor detalle en: https://www.scrumalliance.org/community/articles/2015/march/user-story

Épicas e Historias de Usuario
Ejemplo de Historia con Criterios de Aceptación

Softtek®

› **Historia:**

Yo como Ejecutivo de cuenta, **necesito** consultar los datos de un solicitante de crédito **para** saber si es cliente o no del banco.

› **Criterios de Aceptación:**

Dado que el solicitante es un cliente actual del banco y se sabe su número de cliente, **cuando** proporciono el número de cliente **entonces** el sistema me muestra los datos de las cuentas del cliente (número de cuenta, tipo de cuenta y saldo).

Dado que el solicitante es un cliente actual del banco y proporciona un número de cliente que no es el suyo, **cuando** proporciono el número de cliente **entonces** el sistema me muestra los datos de las cuentas de ese número de cliente pero no corresponden al del solicitante.

Dado que el solicitante es un cliente actual del banco y proporciona un número de cliente inexistente, **cuando** proporciono el número de cliente **entonces** el sistema me envía el mensaje: “Cliente no existe”.

Dado que el solicitante es un cliente del banco pero no se sabe su número de cliente, **cuando** proporciono el apellido paterno o ambos apellidos del cliente **entonces** el sistema me muestra todos los registros coincidentes, me permite seleccionar uno de ellos y me muestra los datos de las cuentas de ese cliente.

Dado que el solicitante NO es cliente del banco, **cuando** proporciono el apellido paterno o ambos apellidos del prospecto **entonces** el sistema me muestra todos los registros coincidentes y ninguno debe corresponder a los datos del solicitante.

Épicas e Historia de Usuario
Descripción de Épica

Épica	Es una “gran historia de usuario” que se puede dividir en varias historias, a menudo comprende un flujo de trabajo completo para ese usuario. Si bien las historias de usuario que componen una épica se pueden completar de forma independiente, su valor comercial no se obtiene hasta que la épica está completa.
Formato Canónico Épica	**Yo como** <Rol>, **necesito** <Funcionalidad> **para** <valor de negocio>

Mayor detalle en:
https://www.scrumalliance.org/community/articles/2014/march/stories-versus-themes-versus-epics

Épicas e Historias de Usuario

Ejemplo división de Épica en Historias de Usuario

- **Épica**:

  Yo como un comprador, **necesito** poder pagar de forma on-line **para** adquirir los productos que están en mi carrito de compras.

- **Historias**:

  Yo como un comprador, **necesito** elegir la forma **para** hacer el pago.

  Yo como un comprador, **necesito** recibir una notificación por correo electrónico **para** tener el acuse de recibo de mi pago.

  Yo como un comprador, **necesito** recibir una notificación por correo electrónico **para** tener el detalle de mi orden.

Mayor detalle en:
https://www.scrumalliance.org/community/articles/2014/may/split-stories-%E2%80%93-story-splitting-patterns-revisted

Épicas e Historias de Usuario
Historias de Usuario NO funcionales

Historias de Usuario NO funcionales

- Una forma de considerar los requerimientos NO funcionales (RNF), es a través del uso de **restricciones** dentro de la descripción de una Historia de Usuario. Por lo general, las restricciones del Producto, pueden abarcan múltiples historias, o la totalidad de la propia aplicación. Un ejemplo de las restricciones del producto pueden ser entre otras: Seguridad, rendimiento, disponibilidad, usabilidad, comportamiento, apariencia**...**

Ejemplos:

- **Yo como** cliente de la aerolínea, **necesito** consultar los vuelos disponibles con un tiempo de respuesta no mayor a 2 segundos **para** garantizar mi tranquilidad y mantener mi confianza en el servicio
- **Yo como** cliente, **necesito** poder ejecutar el producto en todas las versiones de Windows desde Windows 98 hasta Windows 10 **para** tener una mayor flexibilidad y cobertura tecnológica en el uso del servicio
- **Yo como** usuario del sitio, **necesito** que esté disponible 99,999 por ciento de las veces que intento acceder **para** mantener mi fidelidad en el servicio
- **Yo como** operador del servicio de TPVs **necesito** que el sistema pueda procesar 100,000 transacciones concurrentes **para** asegurar la disponibilidad del servicio

Épicas e Historias de Usuario

Tareas ***Spike*** para investigación y aprendizaje

Softtek®

Spike

- Es una "**tarea de investigación**", a realizar dentro del *Sprint*, orientadas a responder una pregunta o a la recopilación de información, en lugar de agregar funcionalidad al producto. Son una buena herramienta para averiguar si algo es posible o no, comprender mejor un requisito o aumentar la fiabilidad de la estimación de una historia de usuario. Suele ser de dos tipos: **Técnico** o **funcional**. El funcional se utiliza para analizar funcionalidad, su **riesgo** y **complejidad**. El técnico se utilizan para determinar la **viabilidad**.

Características

- Tiene **claros los objetivos a lograr**. Se tiene claro el conocimiento que se está tratando de ganar y el problema que está tratando de resolver (es fácil para un equipo que se desvíen con algo interesante y relacionado, pero irrelevante).
- Tiene una **duración definida**. Para hacer justo el trabajo suficientemente para obtener el valor requerido. Normalmente esa duración es como máximo la de un sprint, ya que debe servir para desbloquear alguna tarea o adquirir el conocimiento necesario para poder estimar y construir la HU asociada en el sprint posterior.

¿Cuándo utilizarlos?

- Cuando el equipo identifique historias que son difíciles de comprender, dividir y estimar.
- Cuando la herramienta o tecnología no es bien conocida o adecuada.
- Cuando se observa que la posibilidad de completarla una HU es casi nula debido a su complejidad.

Mayor detalle en: https://www.scrumalliance.org/community/articles/2013/march/spikes-and-the-effort-to-grief-ratio

Épicas e Historias de Usuario
Lo que **SÍ** y lo que **NO**

Softtek®

Lo que SÍ

- Historias con orientación al negocio, Funcionales y NO Funcionales
- **Cómo probar cada Historia** (criterios de aceptación) por parte del *Product Owner*
- Historias **Spike** para Investigación o aprendizaje del equipo de desarrollo
- Historias con nombre corto (de 3 a 10 palabras)
- Grado de importancia para el negocio (**prioridad**) de cada Historia
- Estimación inicial de la **complejidad** de cada Historia por parte del equipo de desarrollo

Lo que NO

- Mismo grado de importancia (prioridad) para la mayoría o todas las Historias
- Estimación inicial de cada Historia hecha por "alguien más"
- Definición de cómo probar cada Historia por parte de alguien diferente al *Product Owner*
- Criterios de aceptación "estándar" o genéricos

La estimación de puntos de historia

Softtek®

Value THROUGH TECHNOLOGY

Puntos de historia

- Mecánica
  - Una historia es seleccionada como referencia: Log In == 3
  - Dimensionar las historias de usuario RELATIVAMENTE a cada otra basándose en la experiencia previa del equipo.

- Raciocinio:
  - Los humanos somos TERRIBLES para estimar tiempos absolutos y, a parte, los estimados no son en tiempo real.
  - Somos mucho mejor para proveer estimaciones RELATIVAS
  - Las teorías de números largos y la secuencia de Fibonacci son VERDADES en el promedio !!!
  - Cambiar el pensamiento a “Juguemos un juego para sumar puntos” en vez de “Lo tengo que hacer en X horas”

La estimación de puntos de historia
Método para medir el tamaño funcional

Identificar los *Data movements* **de cada Historia de Usuario**

- Contar **Entradas** de grupo de datos*
- Contar **Salidas** de grupo de datos
- Contar **Lecturas** de grupo de datos **desde** almacenamiento persistente
- Contar **Escrituras** de grupo de datos **hacia** almacenamiento persistente

Una forma de **consensuar** los *Data movements* **identificados**

- Pulgar arriba – De acuerdo
- Pulgar Horizontal – Puedo vivir con ello
- Pulgar Abajo – No estoy de acuerdo y quiero proponer un ajuste **sustentado**

Contabilizar los Puntos de Historia a atender en el *Sprint*

- **1 Punto de Historia = 1 Movimiento de Datos** (*Data Movement*)

*Conjunto de datos que integran un **objeto de negocio** (p.ej. Dirección, Transacción...)

Mayor detalle en: http://cosmic-sizing.org/cosmic-fsm/

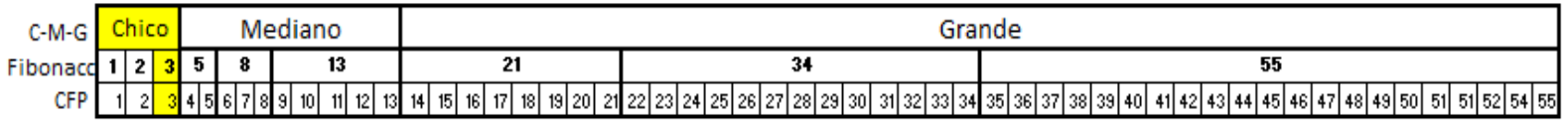
La estimación de puntos de historia
Equivalencia con *Planning Poker* (Fibonacci)

Planning poker es **una técnica** de **estimación** que consiste en una sesión colaborativa con el equipo de desarrollo. Las estimaciones se hacen siguiendo la secuencia de Fibonacci, para expresar la **complejidad**.

Ejemplos

Login

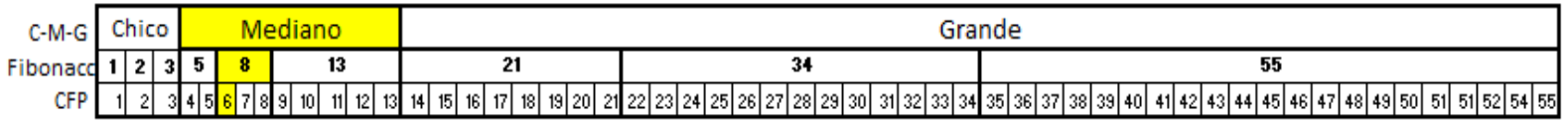
Entradas	1
Salidas	1
Escrituras	
Lecturas	1
Total	3



Puntos de Historia: **3**

Cambio simple

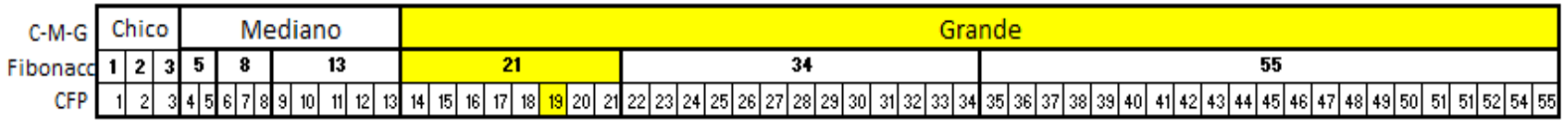
Entradas	2
Salidas	2
Escrituras	1
Lecturas	1
Total	6



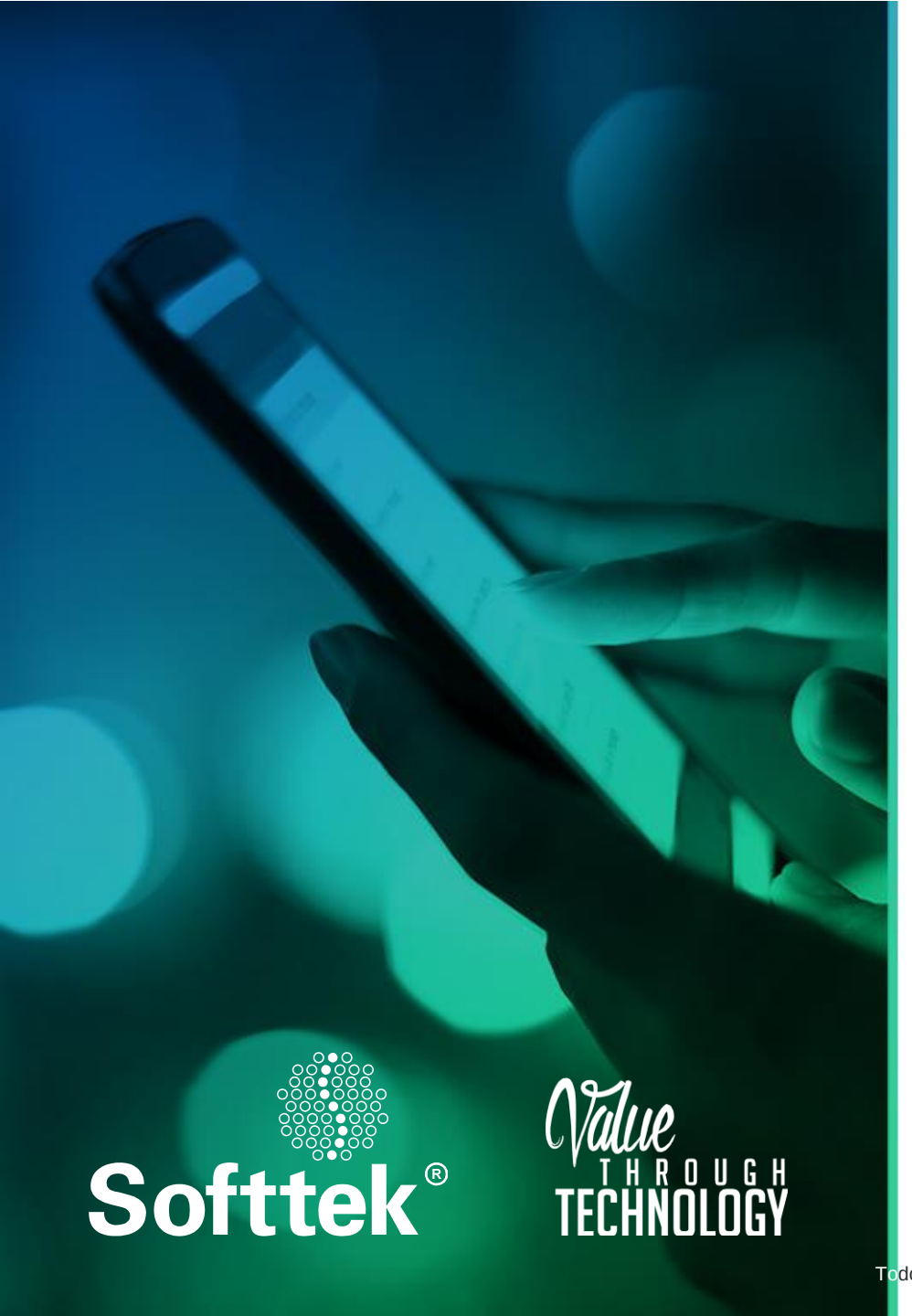
Puntos de Historia: **8**

Disposición TDC

Entradas	3
Salidas	2
Escrituras	3
Lecturas	11
	19



Puntos de Historia: **21**



Manifiesto ágil

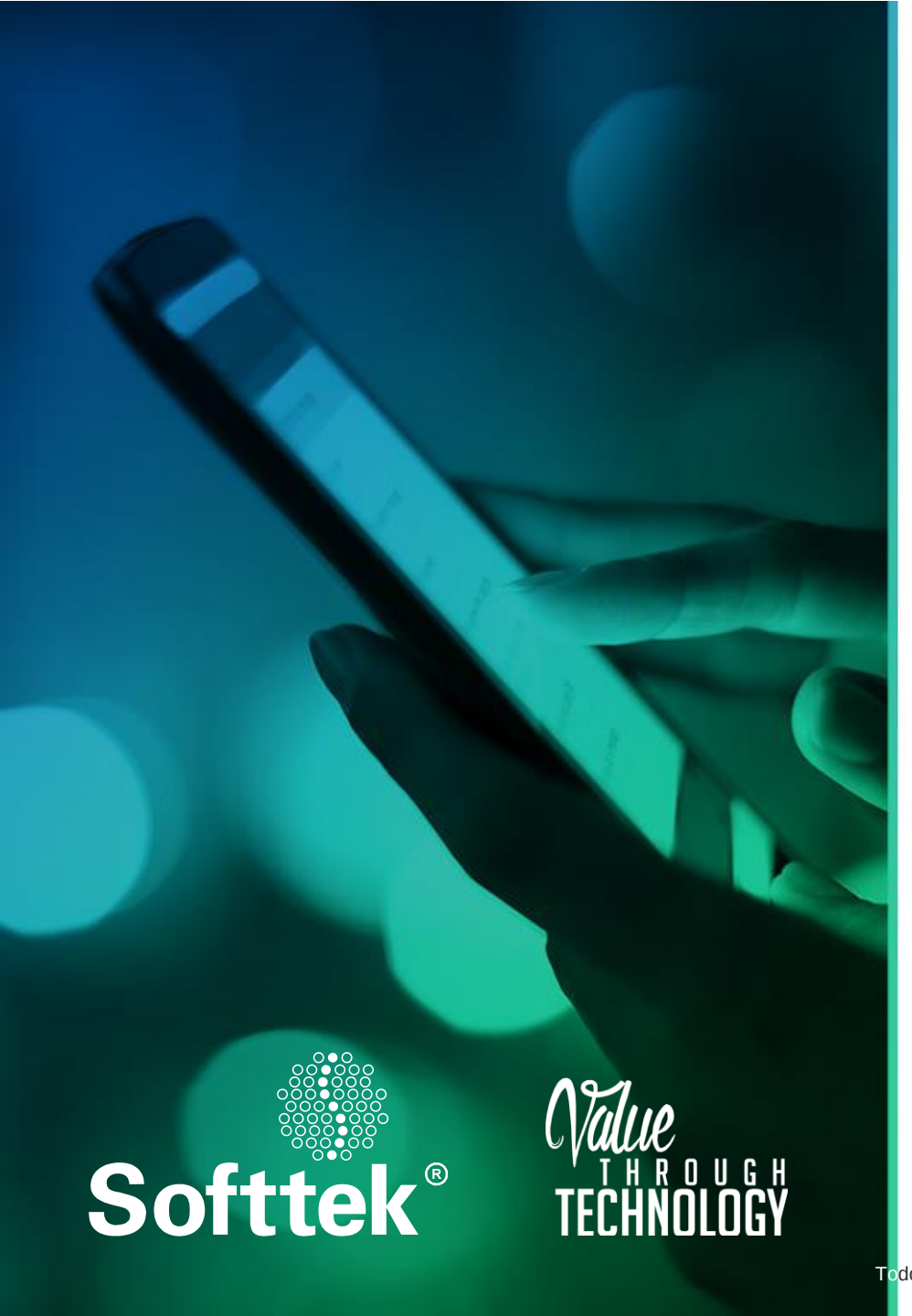
http://agilemanifesto.org/

Scrum Alliance

https://www.scrumalliance.org/

eXtreme Programing

http://www.extremeprogramming.org/



Agile SSDP

http://collaboration.softtek.com/practices/appdev/assdp_en/SitePages/Home.aspx

COSMIC Function Points

http://cosmic-sizing.org/cosmic-fsm/

Scrum y XP desde las trincheras

http://www.proyectalis.com/wp-content/uploads/2008/02/scrum-y-xp-desde-las-trincheras.pdf