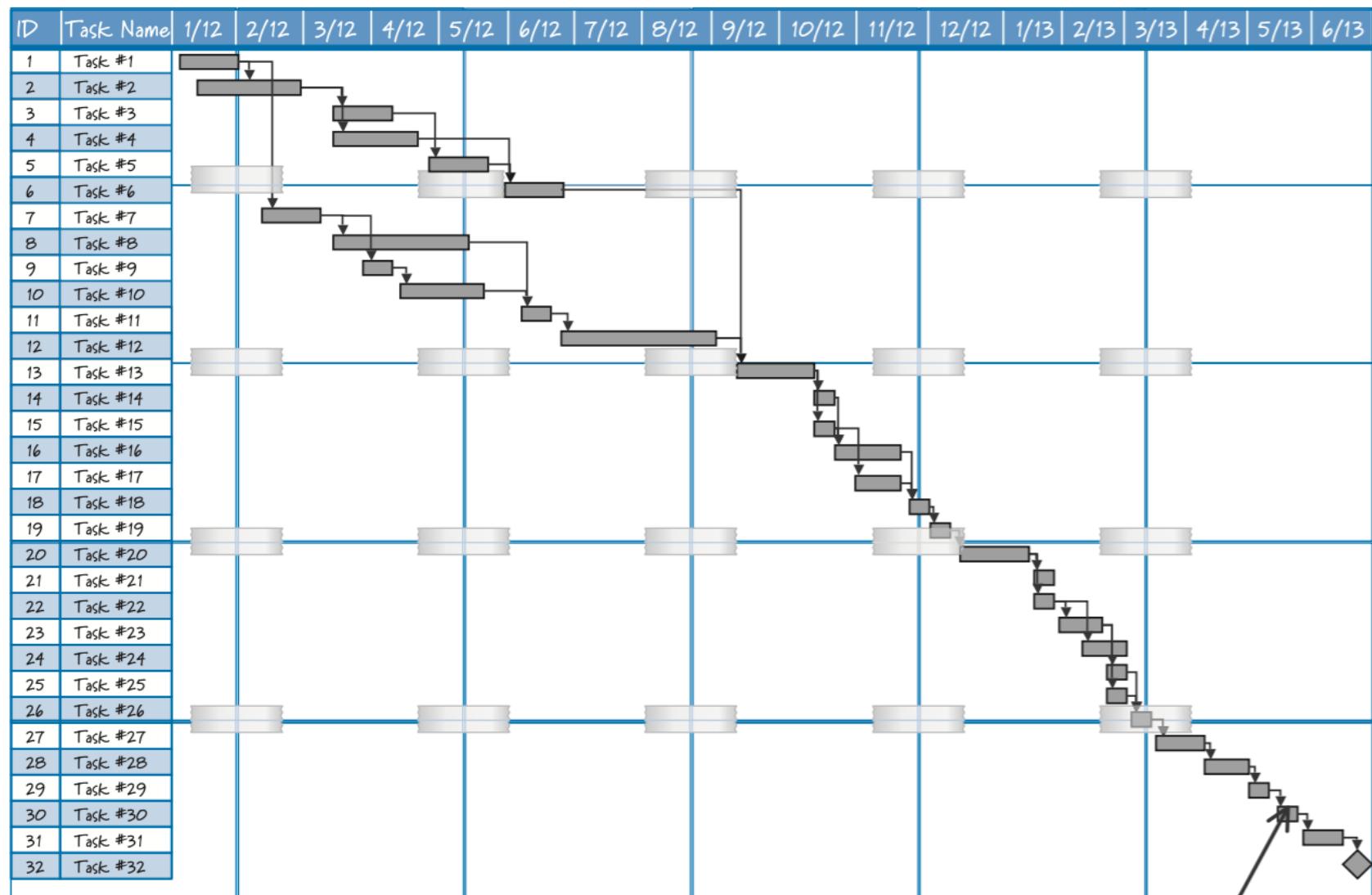


# Planeación



- La analogía del descenso libre en ski
  - ¿ se puede planificar el decenso completo ?
  - ¿ qué se puede planificar ?

# La famosa Carta Gantt



# ¿Por que no es buena idea ?

- Se requiere bastante trabajo en construirla
- Se requiere bastante trabajo para modificarla constantemente
- Rara vez es posible mantenerse dentro del plan inicial
- Puede insistirse en mantener un plan que está fuera de la realidad

Cuando el terreno y el mapa no coinciden hay que creerle al terreno



# Estimación

**Ejecutivo:** Cuanto tiempo crees que tomará el proyecto?  
Necesitamos el software listo en 3 meses para mostrarlo en la expo. No puedo darte mas gente así que sería con el equipo actual.

**Jefe de Proyecto:** OK déjame hacer unos cálculos y te respondo

al día siguiente ...

**Jefe de Proyecto:** Hemos estimado que nos tomará 5 meses

**Ejecutivo:** 5 meses ? No me pusiste atención ? Te dije que lo necesitabamos en 3 meses

# Estimación, Meta y Compromiso

- En el ejemplo anterior los 3 meses era una **meta**
- Los 5 meses era la **estimación**
- Se le pide al jefe de proyecto un **compromiso**

Lo que se le estaba pidiendo al jefe de proyecto no era una estimación sino un plan para lograr la meta

# Diálogo Mejorado

**Ejecutivo:** Cuanto tiempo crees que tomará el proyecto? Necesitamos el software listo en 3 meses para mostrarlo en la expo. No puedo darte mas gente así que sería con el equipo actual.

**Jefe de Proyecto:** A ver si entiendo bien. Es más importante entregar el 100% de las features o tener listo algo para la expo

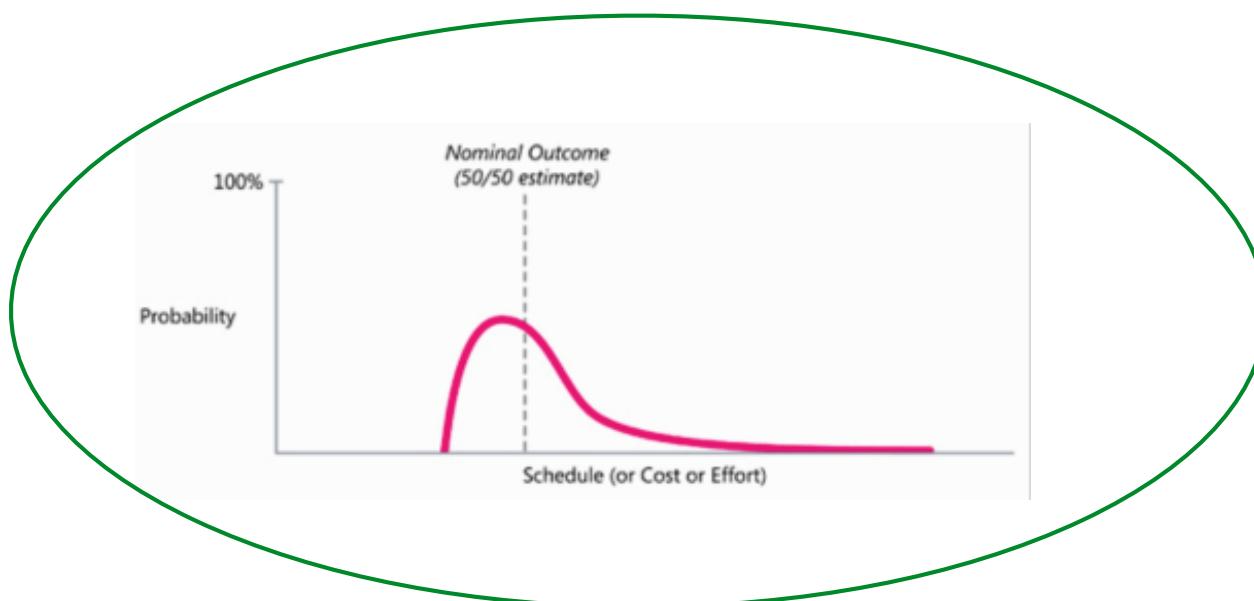
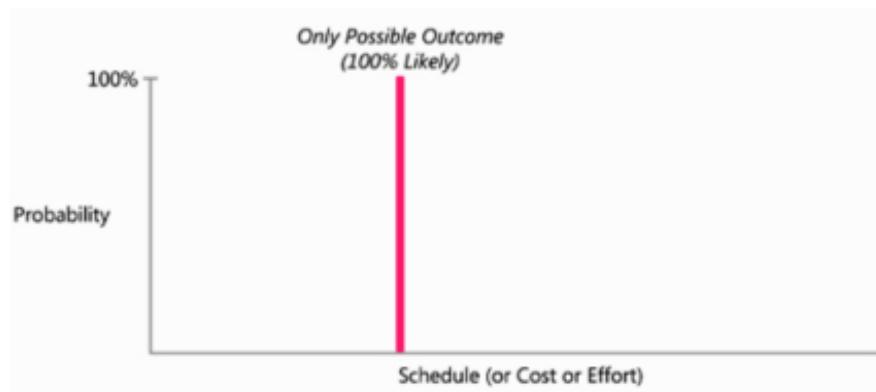
**Ejecutivo:** Necesitamos tener algo para la expo. Ojalá que todas las features

**Jefe de Proyecto:** OK, entonces si resulta que no podemos entregar todas las features para la expo la idea es entregar lo que alcancemos y no correr la fecha de entrega para mas adelante ?

**Ejecutivo:** Tenemos que tener algo para la expo. No me gusta la idea pero tenemos que mostrar algo ese día aún cuando no sea el 100%

**Jefe de Proyecto:** OK. Voy a trabajar en un plan para entregar el máximo de funcionalidad en 3 meses

# Estimaciones son afirmaciones probabilísticas



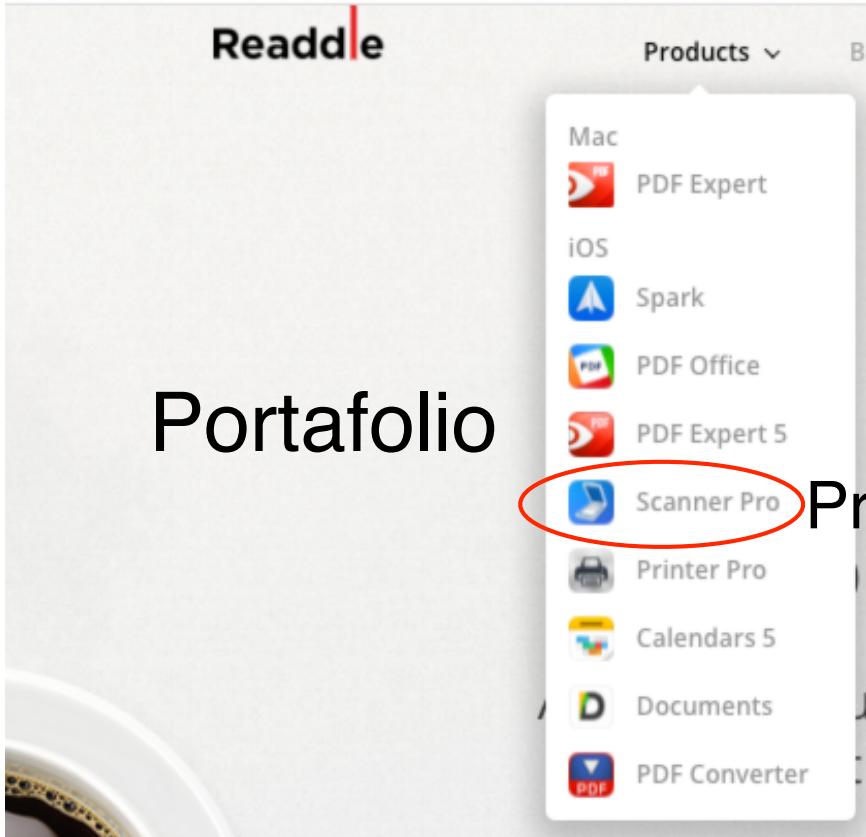
# Estimaciones en Detalle

- Lo estudiaremos mas adelante
- Por ahora nos enfocaremos en la planeación aunque ambas cosas están muy relacionadas

# Varios niveles de planeación

- Iteración - producto potencialmente entregable
- Release - producto entregable al fin del proceso iterativo
- Producto - desarrollo en versiones
- Portafolio - varios productos manejados por la compañía

# Ejemplo



Portafolio Product

**Scanner Pro 7 -**  
By Readdle

Essentials

Open iTunes to buy and c

Release

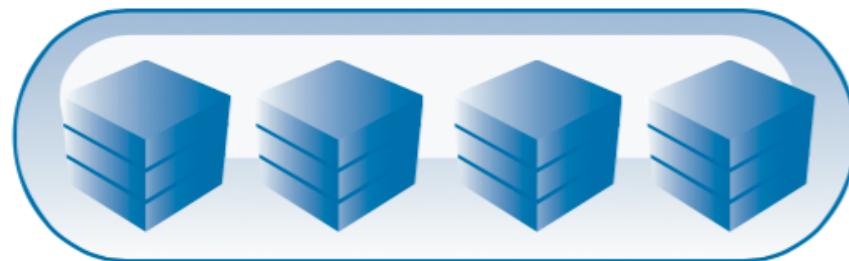


# Planeación de Release

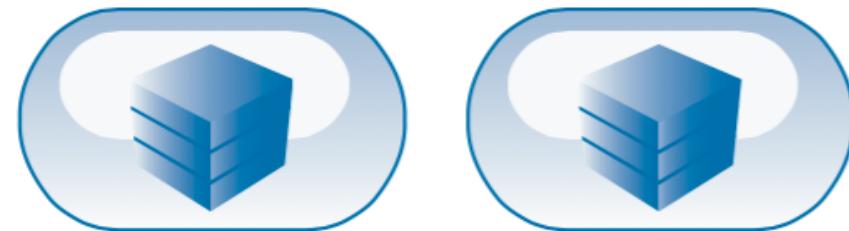
- Salida de un sprint es “potencialmente” entregable
- Organización decide qué y cuando entregar
- Se suelen combinar varios sprints en un release
- Incluso es posible entregar features sin terminar el release

# Posibles Cadencias para Release

Release after multiple sprints



Release every sprint



Release every feature



# Planeación se hace sujeta a restricciones

- El producto tiene que estar lista en una fecha determinada
- El producto tiene que incluir un set de prestaciones
- El desarrollo no puede superar los \$X millones

# La necesidad de ajustar

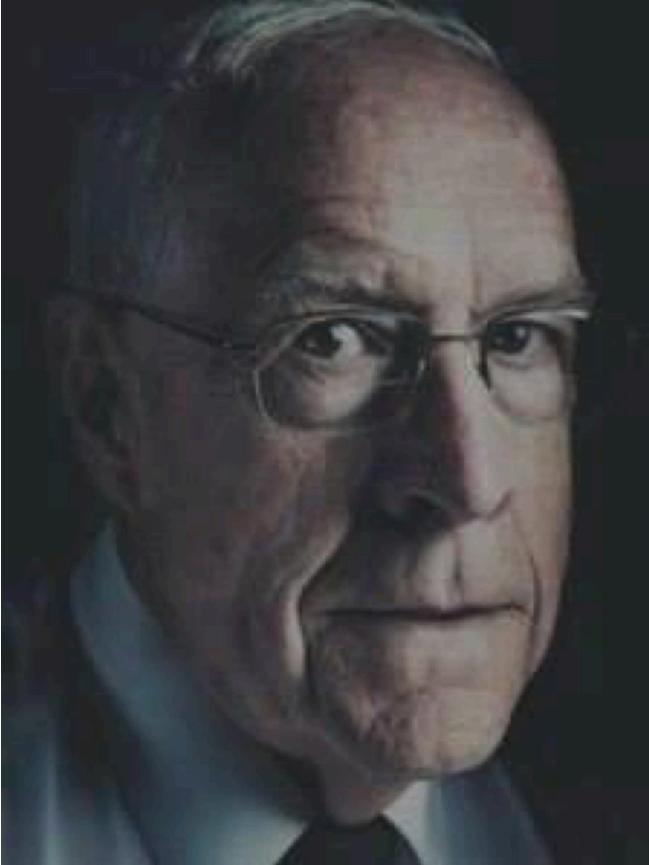
- Es extremadamente difícil lograr entregar la totalidad de las prestaciones en el tiempo disponible ...
- Se debe ajustar a medio camino
  - una nueva fecha de entrega (ajuste por tiempo)
  - un set de prestaciones algo menor (ajuste por alcance)

# Combinaciones Posibles

Project Type	Scope	Date	Budget
Fixed everything (not recommended)	Fixed	Fixed	Fixed
Fixed scope and date (not recommended)	Fixed	Fixed	Flexible
Fixed scope	Fixed	Flexible	Fixed (not really)
Fixed date	Flexible	Fixed	Fixed

“Adding manpower to a late software project makes it later” (Brooks 1995)

“Nine women don’t make a baby in a month” (Brooks 1995)

A close-up portrait of Fred Brooks, an elderly man with glasses and a serious expression, looking slightly to the right.

WHAT ONE PROGRAMMER  
CAN DO IN ONE MONTH, TWO  
PROGRAMMERS CAN DO IN  
TWO MONTHS.

- Fred Brooks

atlaz.io

# Ojo: se suele ajustar por calidad

- muy común
  - acortar trabajo de requisitos, diseño o **testing**
  - eliminar revisión de código
  - eliminar programación en pares
- pésima práctica
- puede generar un tremendo backlog técnico
- puede estar bajo las expectativas del usuario

# Planeación en base a fecha fija

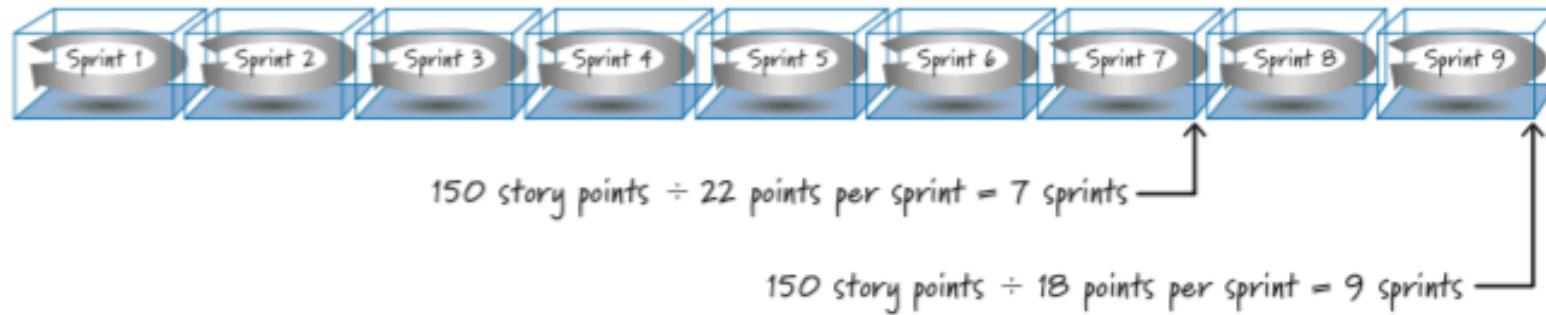
Step	Description	Comments
1	Determine how many sprints are in this release.	If all sprint lengths are equal, this is simple calendar math because you know when the first sprint will start and you know the delivery date.
2	Groom the product backlog to a sufficient depth by creating, estimating the size of, and prioritizing product backlog items.	Because we are trying to determine which PBIs we can get by a fixed date, we need enough of them to plan out to that date.
3	Measure or estimate the team's velocity as a range.	Determine an average faster and an average slower velocity for the team (see Chapter 7).
4	Multiply the slower velocity by the number of sprints. Count down that number of points into the product backlog and draw a line.	This is the “will-have” line.
5	Multiply the faster velocity by the number of sprints. Count down that number of points into the product backlog and draw a second line.	This is the “might-have” line.

# Planeación en base a alcance fijo

Step	Description	Comments
1	Groom the product backlog to include at least the PBIs we would like in this release by creating, estimating the size of, and prioritizing PBIs.	Because this is a fixed-scope release, we need to know which PBIs are in the fixed scope.
2	Determine the total size of the PBIs to be delivered in the release.	If we have a product backlog of estimated items, we simply sum the size estimates of all of the items we want in the release.
3	Measure or estimate the team's velocity as a range.	Determine an average faster and an average slower velocity for the team.
4	Divide the total size of the PBIs by the faster velocity and round up the answer to the next integer.	This will tell us the lowest number of sprints required to deliver the features.
5	Divide the total size of the PBIs by the slower velocity and round up the answer to the next integer.	This will tell us the highest number of sprints required to deliver the features.

# Planeación en base a Alcance fijo

- set the features es poco transable pero estamos dispuestos a postergar la entrega
- todos los features en lista “must-have”
- número de sprints puede ser variable

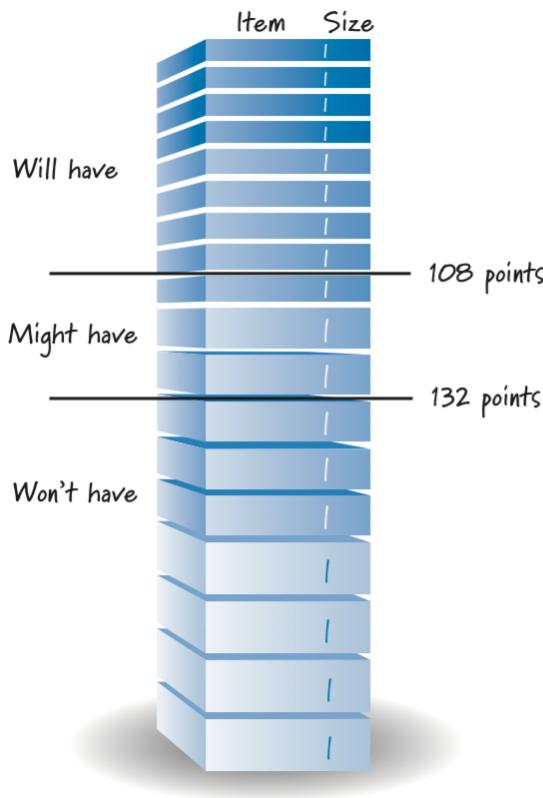


- muchas veces puede cambiarse por una serie de releases de fecha fija

# Minimum Releasable Features (MRF)

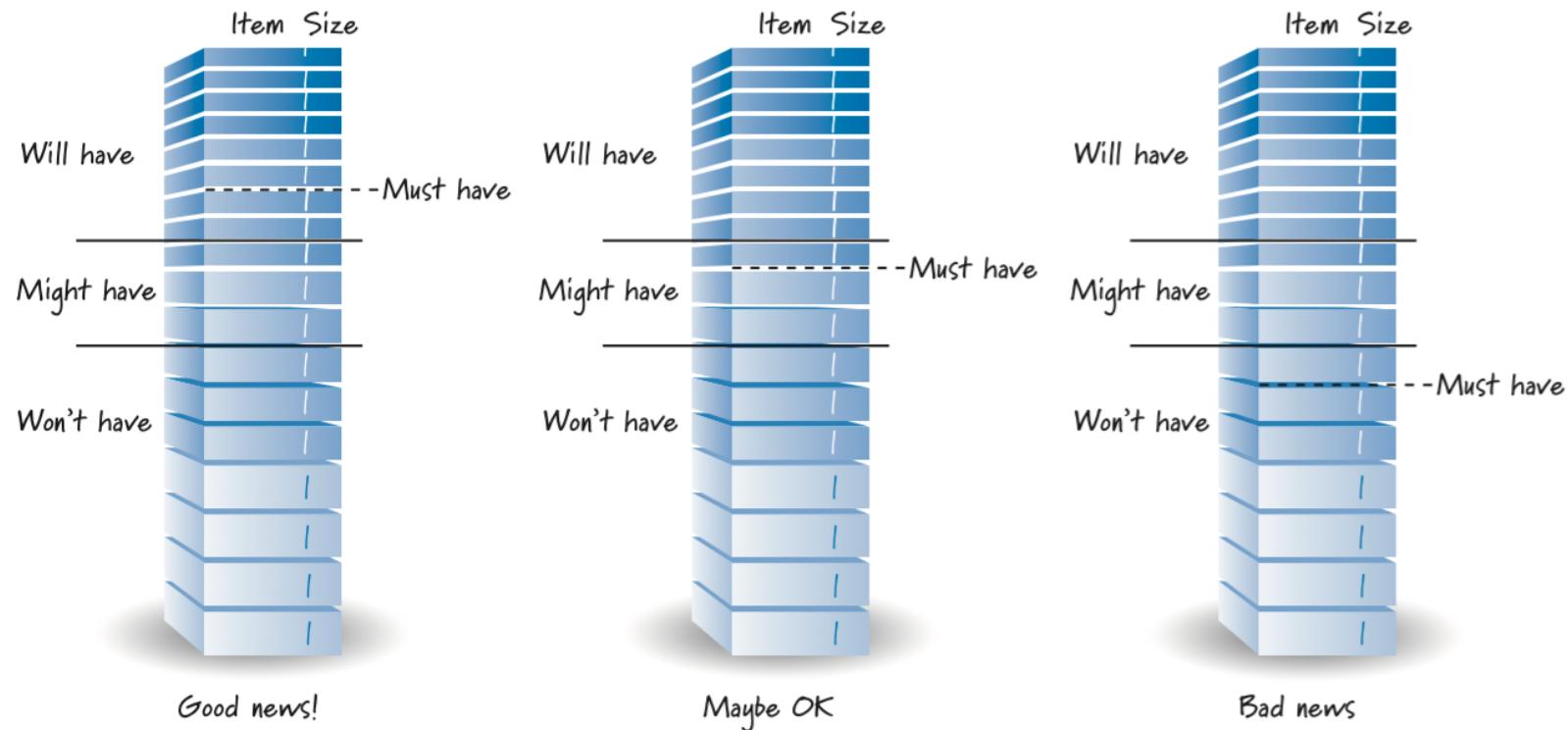
- set mínimo de “must have” features en el release
- responsabilidad del “product owner”
  - una parte importante de planeación del release es
- reevaluar constantemente el MRF a medida que progresan los sprints

# Rango posible

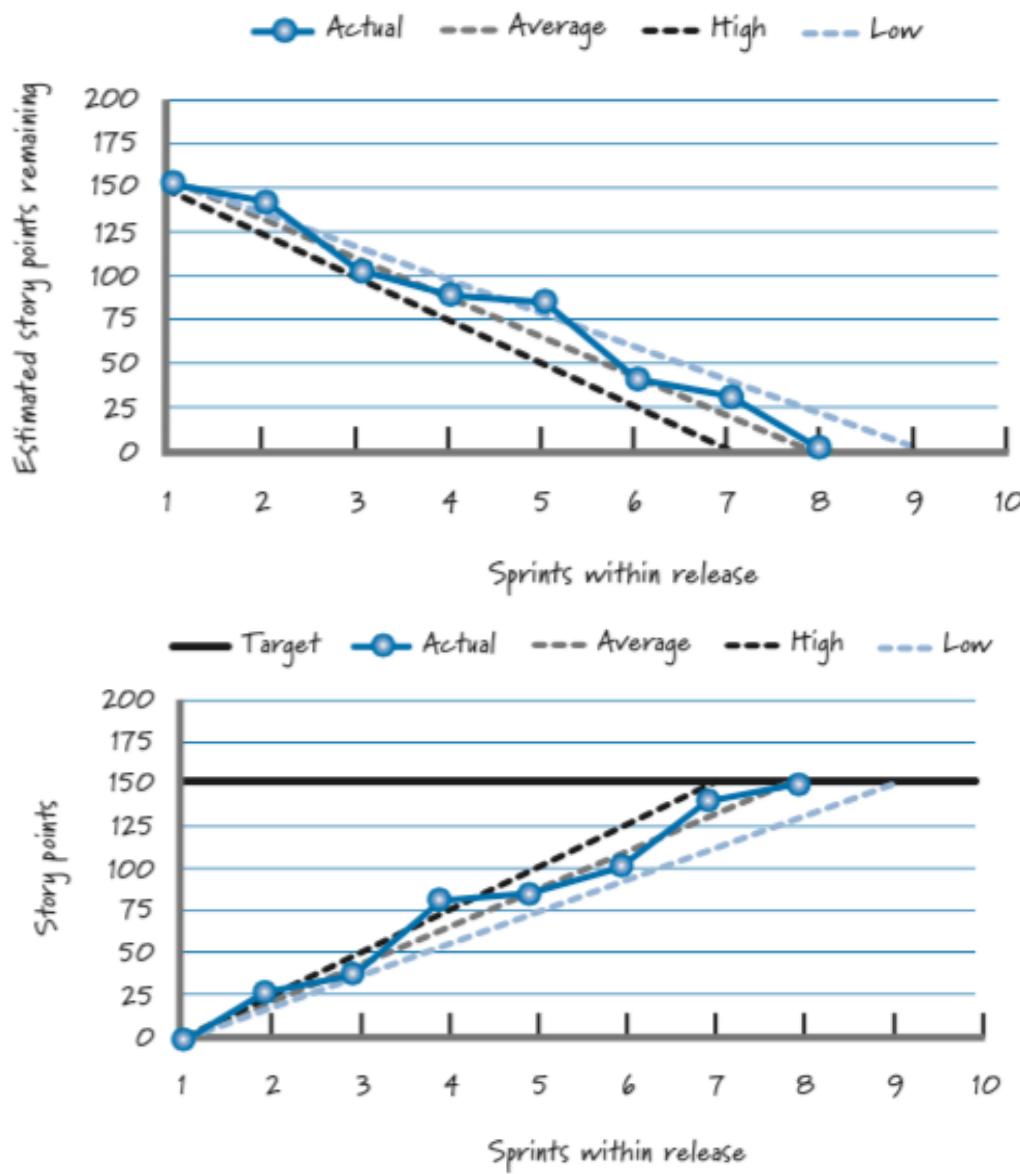


- el PB está priorizado de arriba hacia abajo
- cada item ha sido evaluado en cuanto a esfuerzo (points)
- conocemos la productividad promedio (best case y worst case)
- el primer corte está basado en el worst case
- el segundo corte está basado en el best case
- es conveniente no dejar el MRF muy calzado con el tiempo (pueden surgir otros “must-have”)

# La cruda realidad



# Monitoreando el Progreso



Release burndown

Release burnup

# Por qué es necesario considerar *best case* y *worst case*

## 1. Estimación única

Feature	Estimated Days to Complete
Feature 1	1.5
Feature 2	1.5
Feature 3	2.0
Feature 4	0.5
Feature 5	0.5
Feature 6	0.25
Feature 7	2.0
Feature 8	1.0
Feature 9	0.75
Feature 10	1.25
<b>TOTAL</b>	<b>11.25</b>

Estimación de 11.25 días

## 2. El mismo equipo estima best case y worst case

Feature	Estimated Days to Complete	
	Best Case	Worst Case
Feature 1	1.25	2.0
Feature 2	1.5	2.5
Feature 3	2.0	3.0
Feature 4	0.75	2.0
Feature 5	0.5	1.25
Feature 6	0.25	0.5
Feature 7	1.5	2.5
Feature 8	1.0	1.5
Feature 9	0.5	1.0
Feature 10	1.25	2.0
<b>TOTAL</b>	<b>10.5<sup>1</sup></b>	<b>18.25</b>

Estimación de 11.25 días está  
mucho mas cerca del "best case"

Estimación más real sería de unos 14 días

# ¿Que demonios es un "story point" ?

- es la unidad de esfuerzo mas usada a nivel de release al usar desarrollo ágil
- es una forma de dimensionar en términos relativos el esfuerzo asociado a los distintos relatos
- escalas mas usadas
  - potencias de 2: 1, 2, 4, 8, ...
  - serie de fibonacci : 1, 2, 3, 5, 8, ...

# Ojo: no confiar ciegamente en las escalas

- Uso de una escala como por ejemplo 1, 2, 3, 5, 8 implica que un relato calificado con 5 debe involucrar esfuerzo 5 veces mayor que uno de 1
- Requiere disciplina al asignar los puntos al relato
- Partir con los mas simples (1) y luego pensar en forma relativa

# ¿ Como estimar las velocidades de desarrollo ?

- Examinando lo que ha pasado antes !
  - en la iteración anterior
  - en el release anterior
  - en el proyecto anterior (mismo equipo)
  - promedios de la compañía
  - promedios de la industria

# Por qué es necesario usar información histórica

- Permite contrarrestar tendencia natural al optimismo en un proyecto nuevo
  - esta vez no se nos irá el jefe del proyecto
  - esta vez tenemos mejor tecnología
  - esta vez no perderemos tiempo al comienzo
- Estimación basada en historia nos dice que ocurrirá mas o menos lo mismo

# Ejemplo: Proyecto de 180 story points

---

## **Data for Iteration 1**

---

27 story points delivered

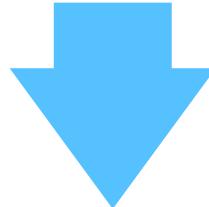
12 staff weeks expended

3 calendar weeks expended

### **Preliminary Calibration**

Effort = 27 story points ÷ 12 staff weeks = 2.25 story points/staff week

Schedule = 27 story points ÷ 3 calendar weeks = 9 story points/calender week



---

## **Data for Iteration 1**

---

### **Assumptions (from Preliminary Calibration)**

Effort = 2.25 story points/staff week

Schedule = 9 story points/calender week

Project size = 180 story points

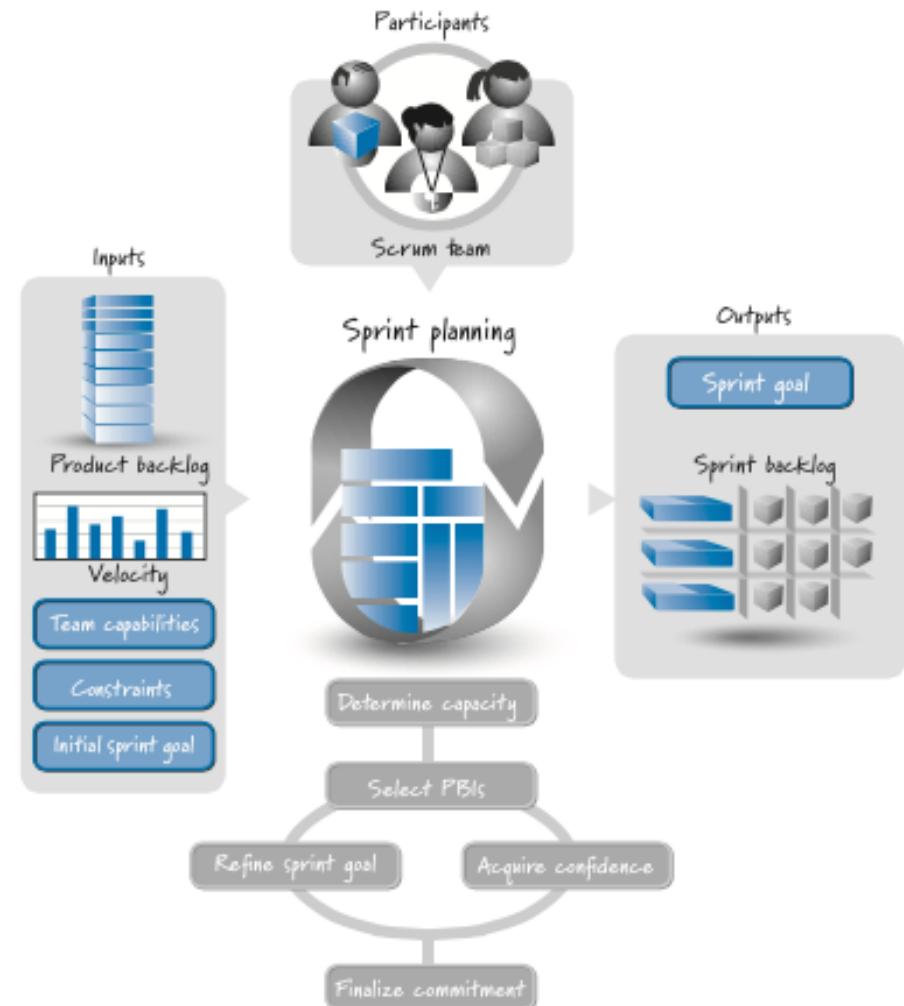
### **Preliminary Whole-Project Estimate**

Effort = 180 story points ÷ 2.25 story points/staff week = 80 staff weeks

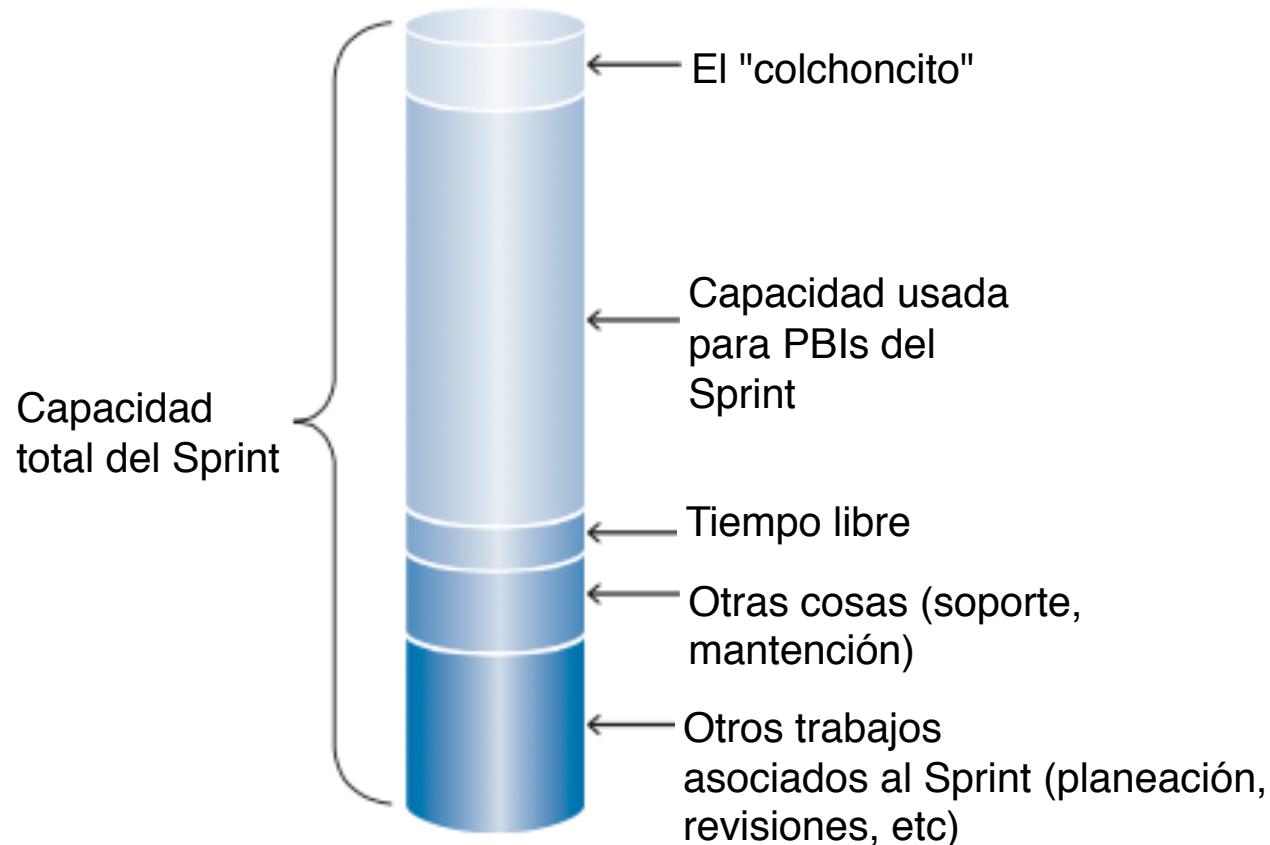
Schedule = 180 story points ÷ 9 story points/calender week = 20 calendar weeks

# Planeación a nivel de Sprint

- participa todo el equipo
- product owner presenta PBIs priorizados
- otros inputs necesarios incluyen velocidad de desarrollo, capacidad del equipo, restricciones
- ScrumMaster debe ayudar a que el equipo incluya set razonable de items



# Capacidad del equipo y capacidad disponible para el Sprint

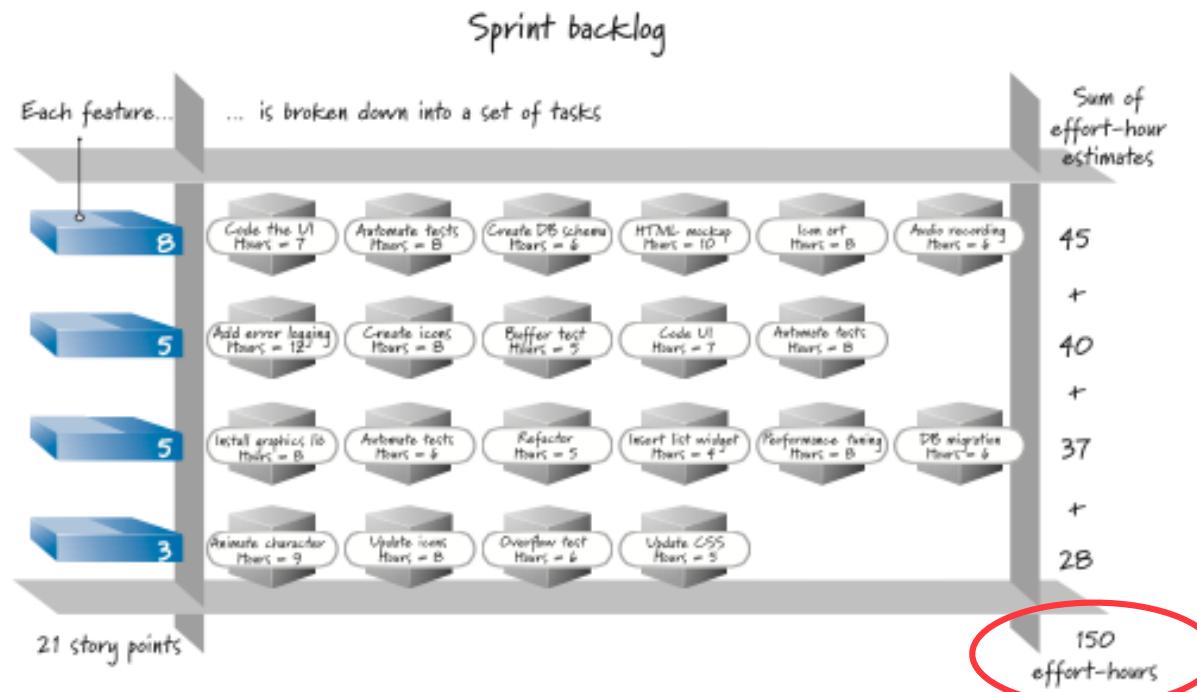


# Ejemplo de Planificación del Sprint

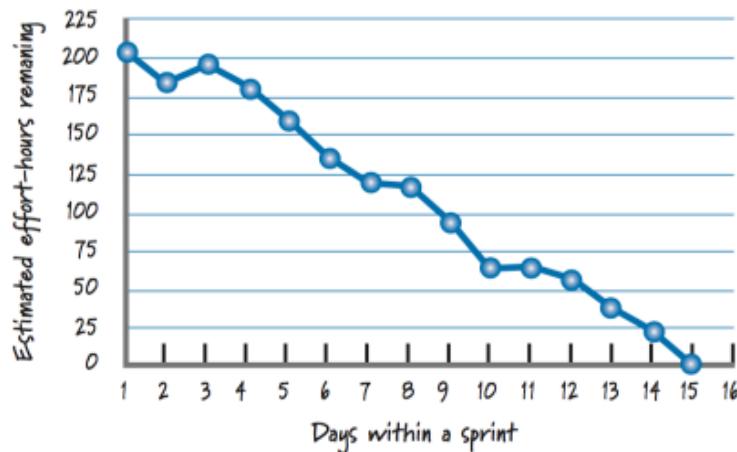
- Capacidad disponible para trabajar en PBIs durante un sprint

Person	Days Available (Less Personal Time)	Days for Other Scrum Activities	Hours per Day	Available Effort-Hours
Jorge	10	2	4–7	32–56
Betty	8	2	5–6	30–36
Rajesh	8	2	4–6	24–36
Simon	9	2	2–3	14–21
Heidi	10	2	5–6	40–48
Total				140–197

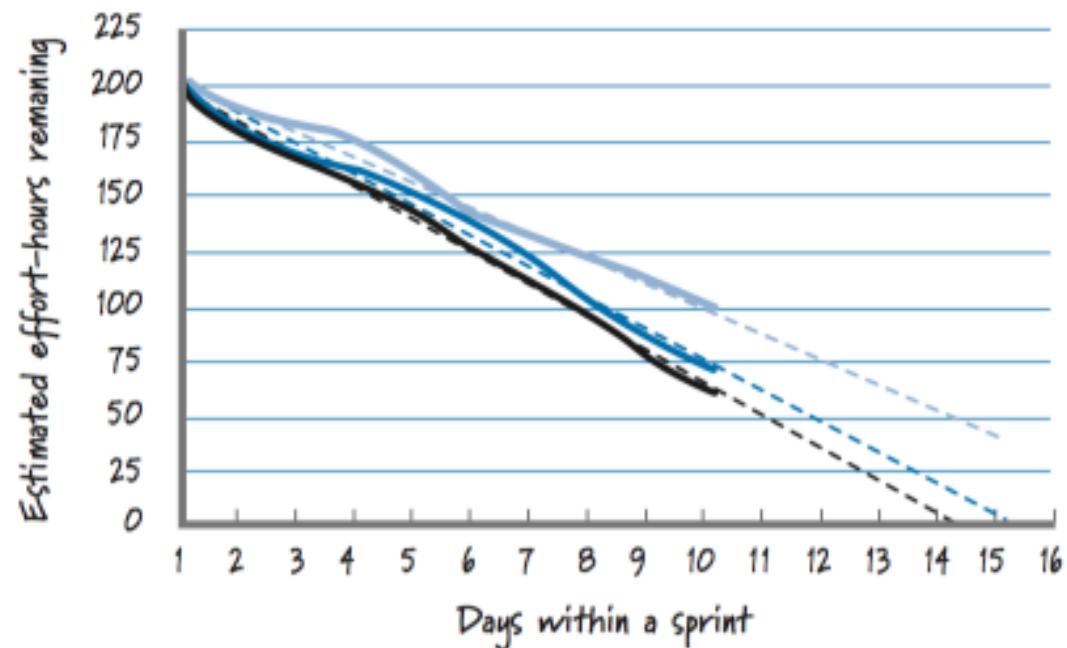
- Capacidad disponible para trabajar en PBIs durante un sprint



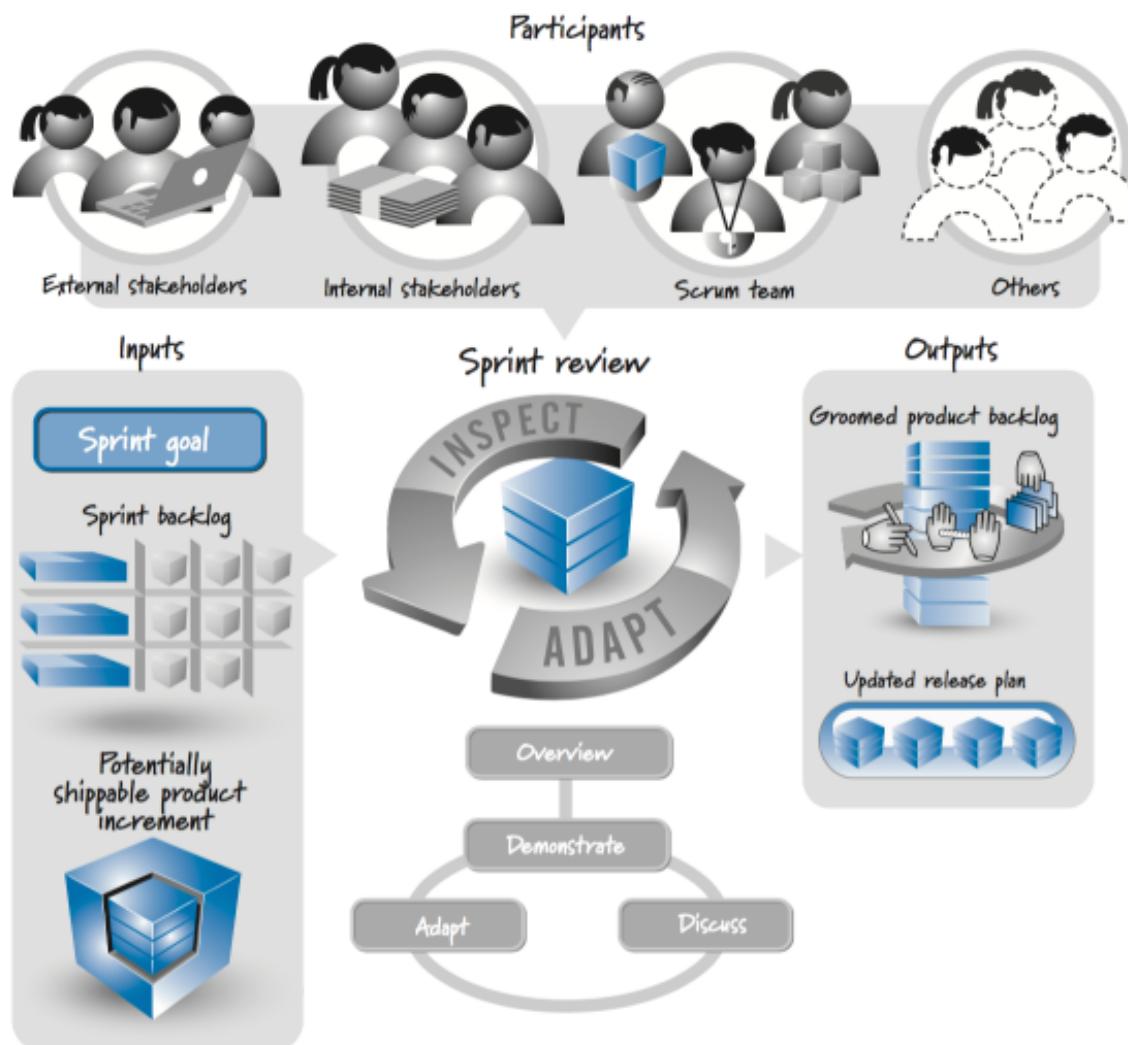
# Sprint burndown charts



On time      Early      Late



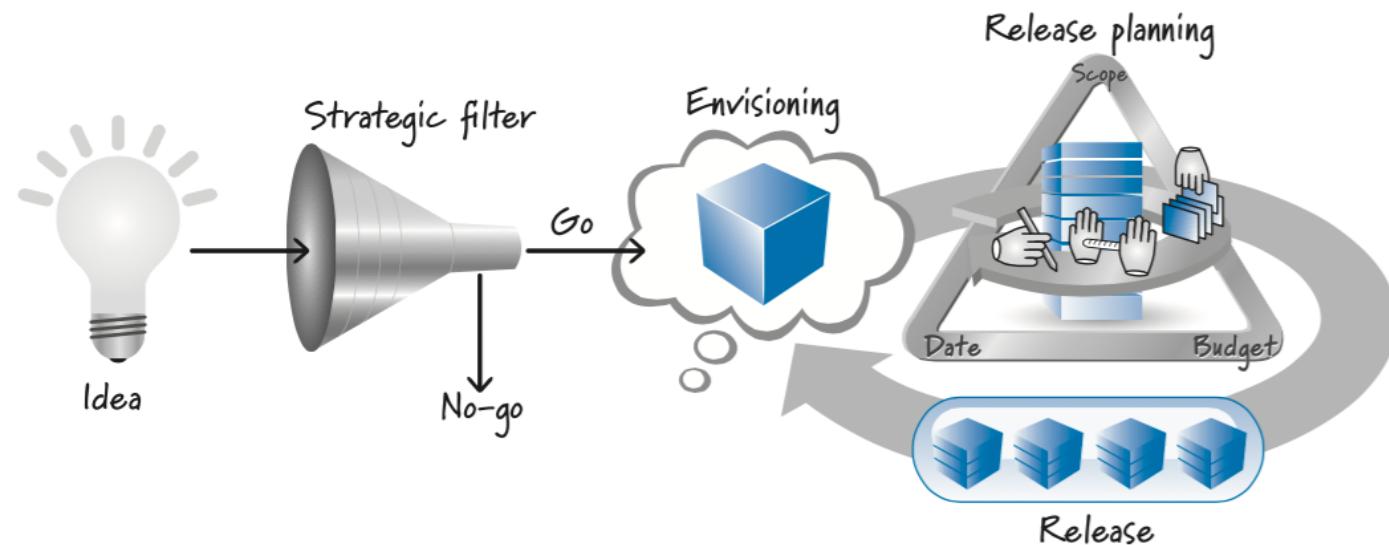
# Output del Sprint Review



nuevo product backlog

ajuste al plan para el release del producto

# Planeación del Producto



- Visión de producto genera un backlog inicial (PBI)
- Proceso de planeación genera una "hoja de ruta" para los futuros releases

# Formatos para “Visión”

Format	Description
Elevator statement	Write a 30-second to one-minute quick pitch of the product vision. Imagine you have stepped into an elevator with a venture capitalist and you have to pitch him on your product vision. Could you do it in a short elevator ride?
Product datasheet	Write the product datasheet on the first day. Try to fit it on the front side of a one-page marketing piece.
Product vision box	Draw the box in which you want to put the product when it ships. Can you come up with three or four salient points to illustrate on the box? (Drafting 15 points is easier than drafting three or four.)
User conference slides	Create the two or three presentation slides that you would use to introduce the product at your user conference (or equivalent). Try to avoid any bullet points on your slides.
Press release	Write the press release you want to issue when the product becomes available. Good press releases clearly communicate what is newsworthy in one page or less.
Magazine review	Draft a fictitious magazine review bylined by the solution reviewer in your industry's most popular trade magazine.

# Ejemplo: SR4U

## Set Asociado a Visión

As a typical user I want to teach SR4U what types of reviews to discard so that SR4U will know what characteristics to use when discarding reviews on my behalf.

As a typical user I want a simple, Google-like interface for requesting a review search so that I don't have to spend much time describing what I want.

As a typical user I want to have SR4U monitor the Internet for new reviews on products or services of interest and automatically filter and report them to me so that I don't have to keep asking SR4U to do it for me.

As a sophisticated user I want to tell SR4U which sources to use when searching on my behalf so that I don't get back reviews from sites I don't like or trust.

As a product vendor I want to be able to show an SR4U-branded review summary for my product on my website so that people can immediately see what the marketplace thinks of my product as determined by a trusted source like SR4U.

# La hoja de ruta del producto (SR4U)

N		Q3-Year 1	Q4-Year 1	Q1-Year 2
Market map		Initial launch	Better results More platforms	Sophisticated users
Feature/benefit map		Basic learning Basic filtering	Improved learning Complex queries	Define sources Learn by example
Architecture map		100K concurrent web users	iOS and Android	Web services interface
Market events		Social Media Expo	Review Everything User Conference	
Release schedule		1.0	2.0	3.0

# Planeación de Portafolio

- cuando organización maneja más de un producto
- en que backlog items del portafolio trabajar y en que orden
  - ¿ mejorar producto A o extender producto B primero ?
- participantes deben tener una visión amplia del negocio

# Planeación Multinivel

Level	Horizon	Who	Focus	Deliverables
Portfolio	Possibly a year or more	Stakeholders and product owners	Managing a portfolio of products	Portfolio backlog and collection of in-process products
Product (envisioning)	Up to many months or longer	Product owner, stakeholders	Vision and product evolution over time	Product vision, roadmap, and high-level features
Release	Three (or fewer) to nine months	Entire Scrum team, stakeholders	Continuously balance customer value and overall quality against the constraints of scope, schedule, and budget	Release plan
Sprint	Every iteration (one week to one calendar month)	Entire Scrum team	What features to deliver in the next sprint	Sprint goal and sprint backlog
Daily	Every day	ScrumMaster, development team	How to complete committed features	Inspection of current progress and adaptation of how best to organize the upcoming day's work