

# Sprint 2: Intermedios

- Mgtr. Diego Rubio
- Mgtr. Natalia Andriano
- Ing. Juan Pablo Bruno
- Ing. Mauricio Silclir



# Objetivos específicos

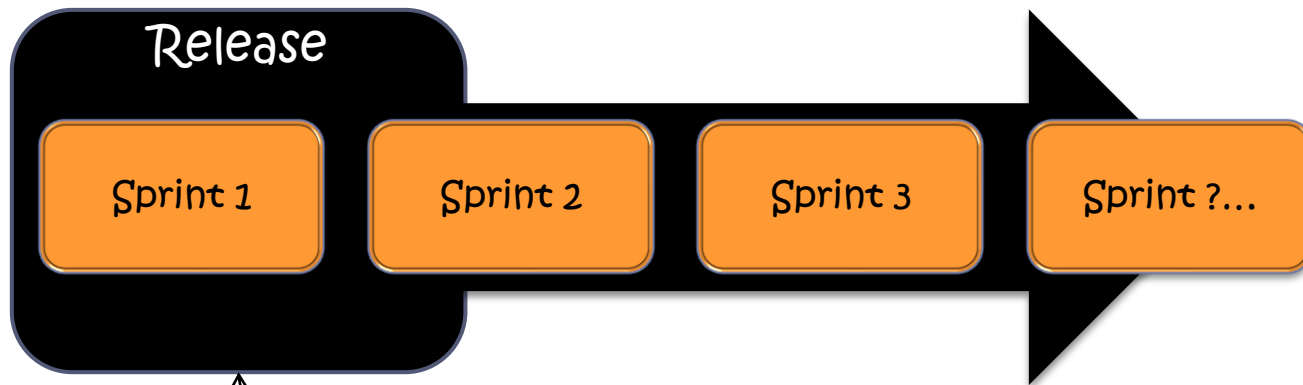
- Desarrollar en el estudiante conceptos básicos acerca de las diferentes técnicas existentes para la **ingeniería de requerimientos** siguiendo una metodología ágil
- Desarrollar en el estudiante conceptos acerca cuales son los atributos a tener en cuenta para un correcto **manejo en el cambio del alcance**.
- Introducir a los estudiantes técnicas intermedias de testing ágil. **Test funcionales y de usabilidad**.
- Presentar otras **métricas** a tener en cuenta para la gestión correcta del proyecto ágil.

# Seguimiento del *release*

- Coordenadas para “llegar a destino”
- Línea recta desde el punto “A” al “B”
  - Volatilidad en los requerimientos
  - Errores en cómo medimos nuestra posición.



# Seguimiento del *release*



Al inicio del release sabemos:

- Por los próximos 4 meses
- Y 4 iteraciones de 2 semanas
- Completaremos 240 puntos de historia

- Considerar:
  - Progreso
  - Cambios en el alcance
  - Re estimaciones en base al conocimiento y experiencia adquiridas.

# Seguimiento del *release*

- Progreso
  - Lo medimos con la VELOCIDAD

**COMPLETO**  
**No** significa el código  
está listo pero no fue  
testeado

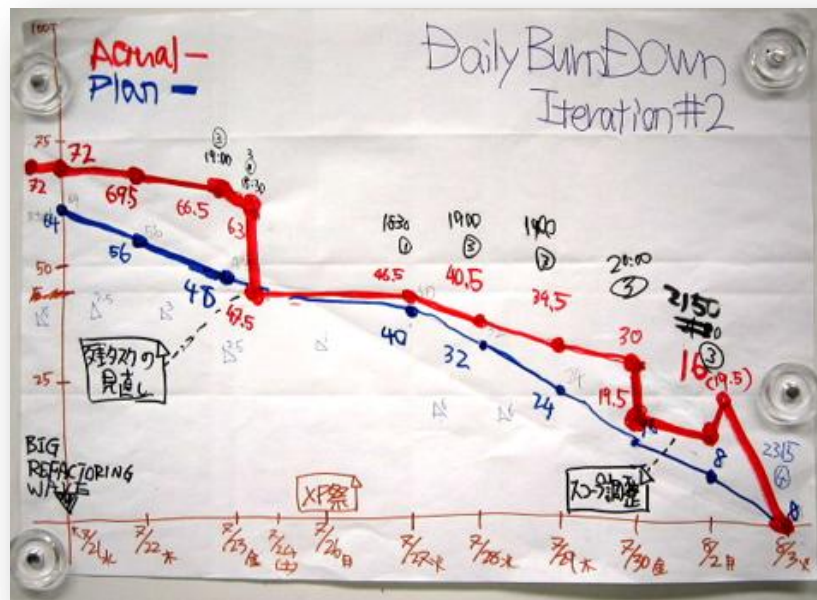
Suma de puntos de historias  
completados en una iteración

Porqué completos????

- Extremadamente difícil de medir trabajo incompleto
- Historias incompletas reducen la confianza del cliente
- Lleva a acumulación de trabajo sin terminar

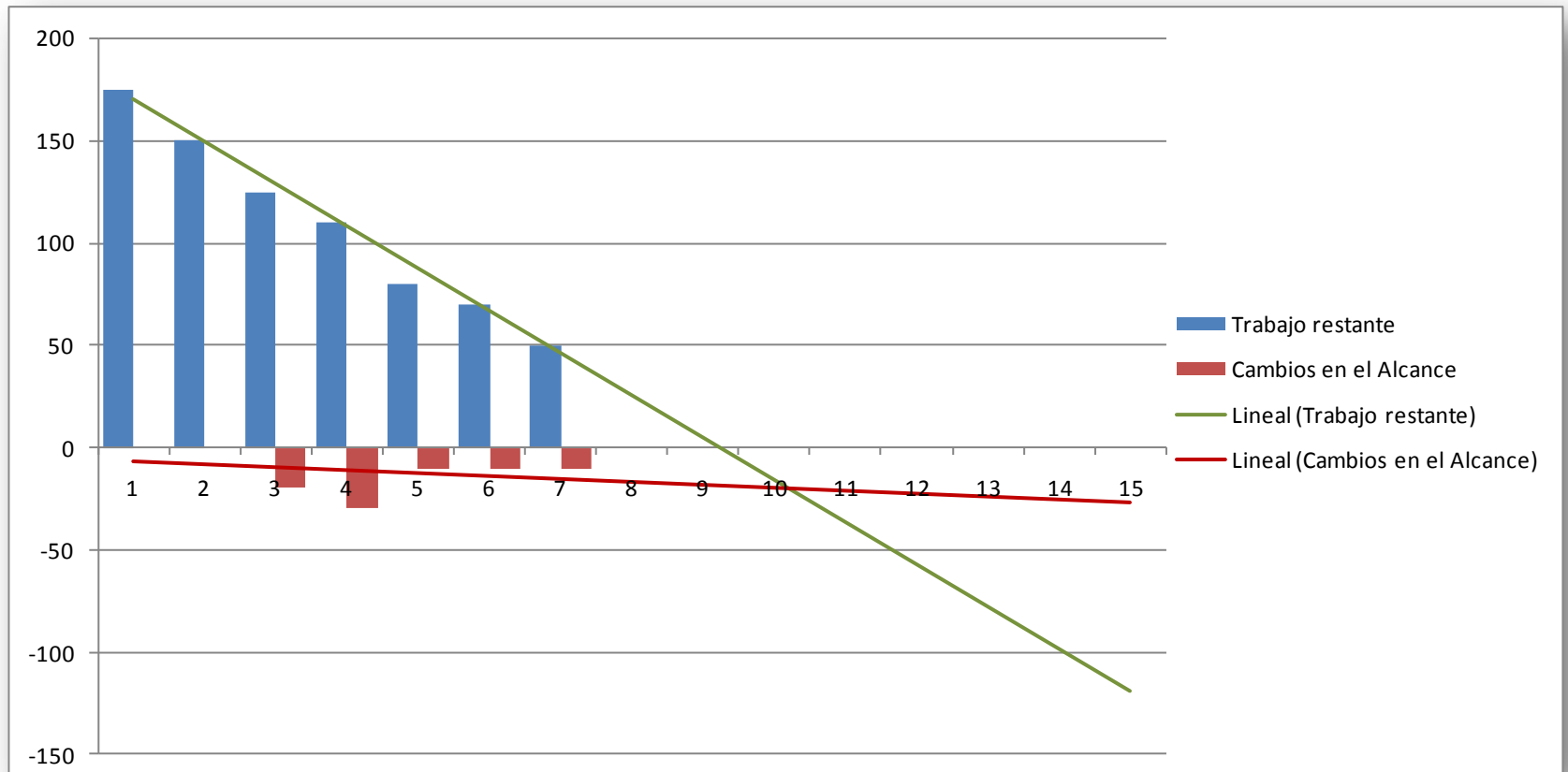
# Seguimiento del *release*

- Cambios en el alcance
  - Release Burndown chart



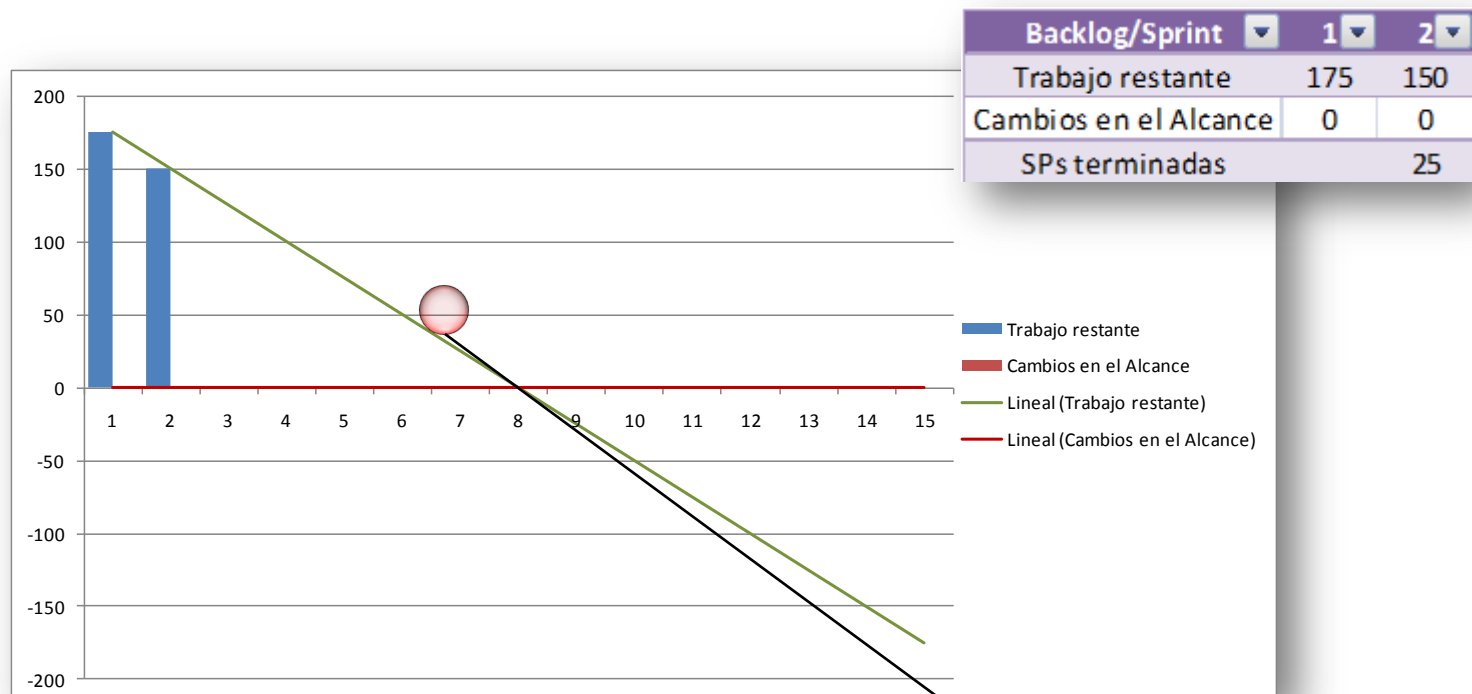
Muestra los puntos de historias o días ideales que quedan por cerrar para el proyecto al inicio de cada iteración.

# Ejemplo: Release Burndown





# Ejemplo: Release Burndown



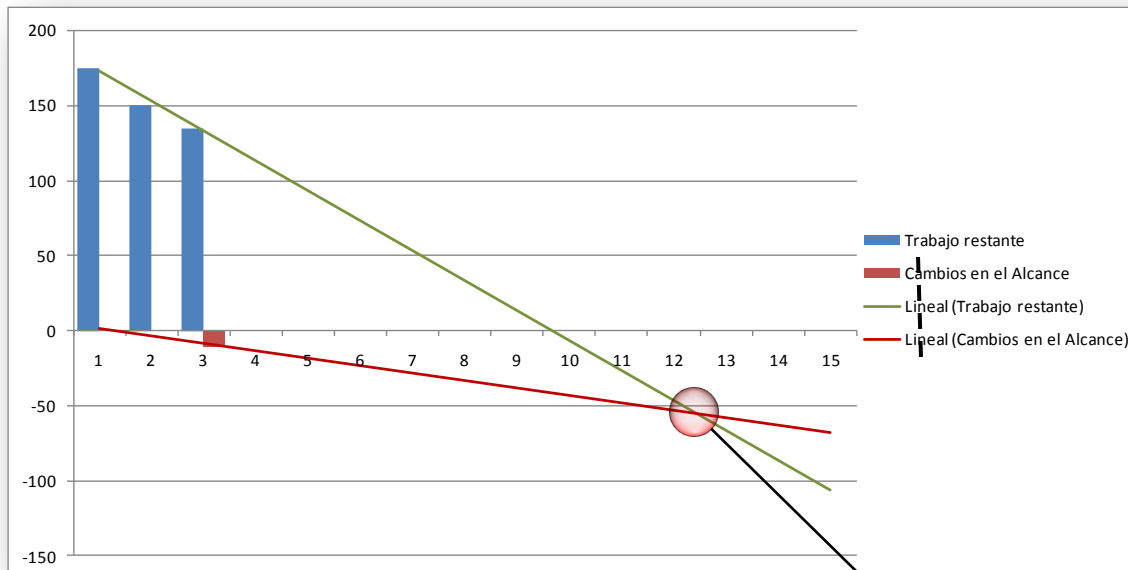
Probable fin = **Sprint 8**



# Cambio de Alcance

Backlog/Sprint	1	2	3
Trabajo restante	175	150	135
Cambios en el Alcance	0	0	-10
SPs terminadas	0	25	25

Se agregaron **10** story points!!



Probable fin = **Sprint 12**

# Cambio de Alcance y Velocidad

Backlog/Sprint	1	2	3	4
Trabajo restante	175	150	135	125
Cambios en el Alcance	0	0	-10	-5
SPs terminadas	0	25	25	15

Se volvieron a agregar  
**5** story points

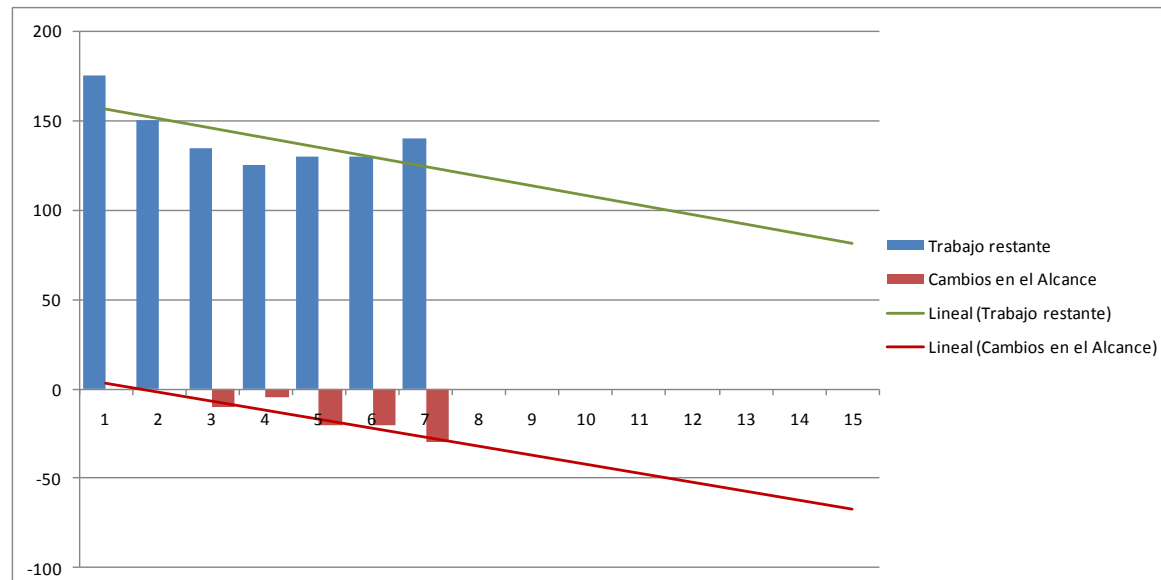
SPs Terminadas = **15**  
(durante el sprint 3)



Probable fin = **Sprint 13**

# ¿Alguna semejanza con la realidad?

Backlog/Sprint	1	2	3	4	5	6	7
Trabajo restante	175	150	135	125	130	130	140
Cambios en el Alcance	0	0	-10	-5	-20	-20	-30
SPs terminadas	0	25	25	15	15	20	20

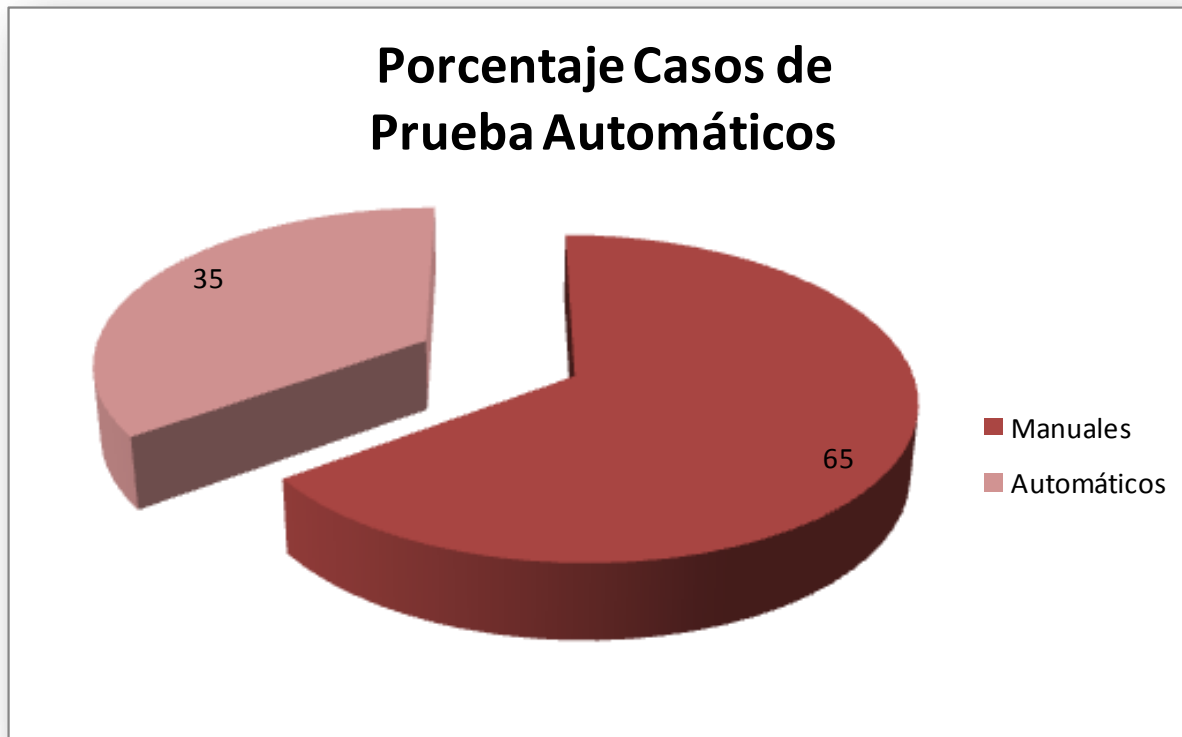


Probable fin = ?

# Además podemos calcular...

- Automatización de tests:
  - Brinda una medida del número de casos de prueba automatizados en relación a la Cantidad total de casos de prueba.
  - $\text{Pruebas Automatizadas} = \frac{\# \text{ Pruebas Automatizadas}}{\# \text{ Pruebas Totales}}$
  - Cuándo? → Es recomendable colectarla semanalmente.

# Automatización de tests

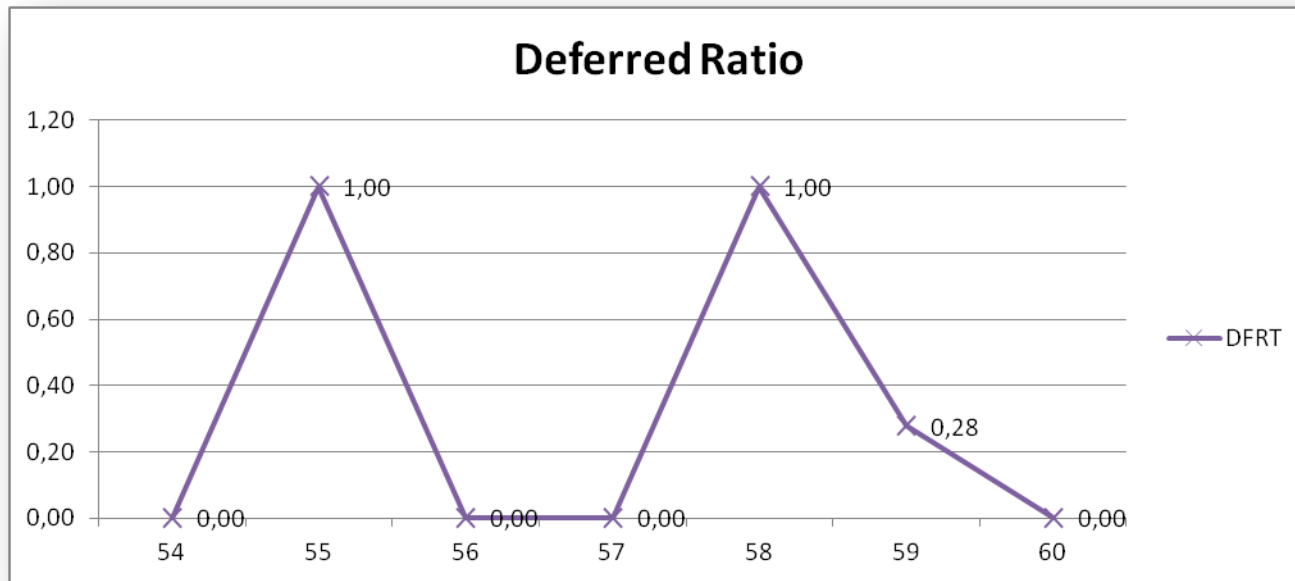


# Y también...

- **Deferred ratio**

- Muestra el número de historias de usuarios diferidas por iteración, en relación a la Cantidad total de historias de usuarios inicialmente comprometidos.
- **Deferred Ratio = # historias de usuarios diferidas / # historias de usuarios totales de la iteración**
- Cuándo? → Se colecta un punto por iteración. Se utiliza para la planificación de cada sprint, y para los ajustes en el release.

# Deferred ratio





# Volviendo hacia el inicio...



# ¿Qué son los requerimientos ágiles?



# Estrategias / Buenas prácticas

## Stakeholders

- Participación activa de los involucrados
- Reconocer la amplia variedad de stakeholders
- Adoptar la terminología de los stakeholders

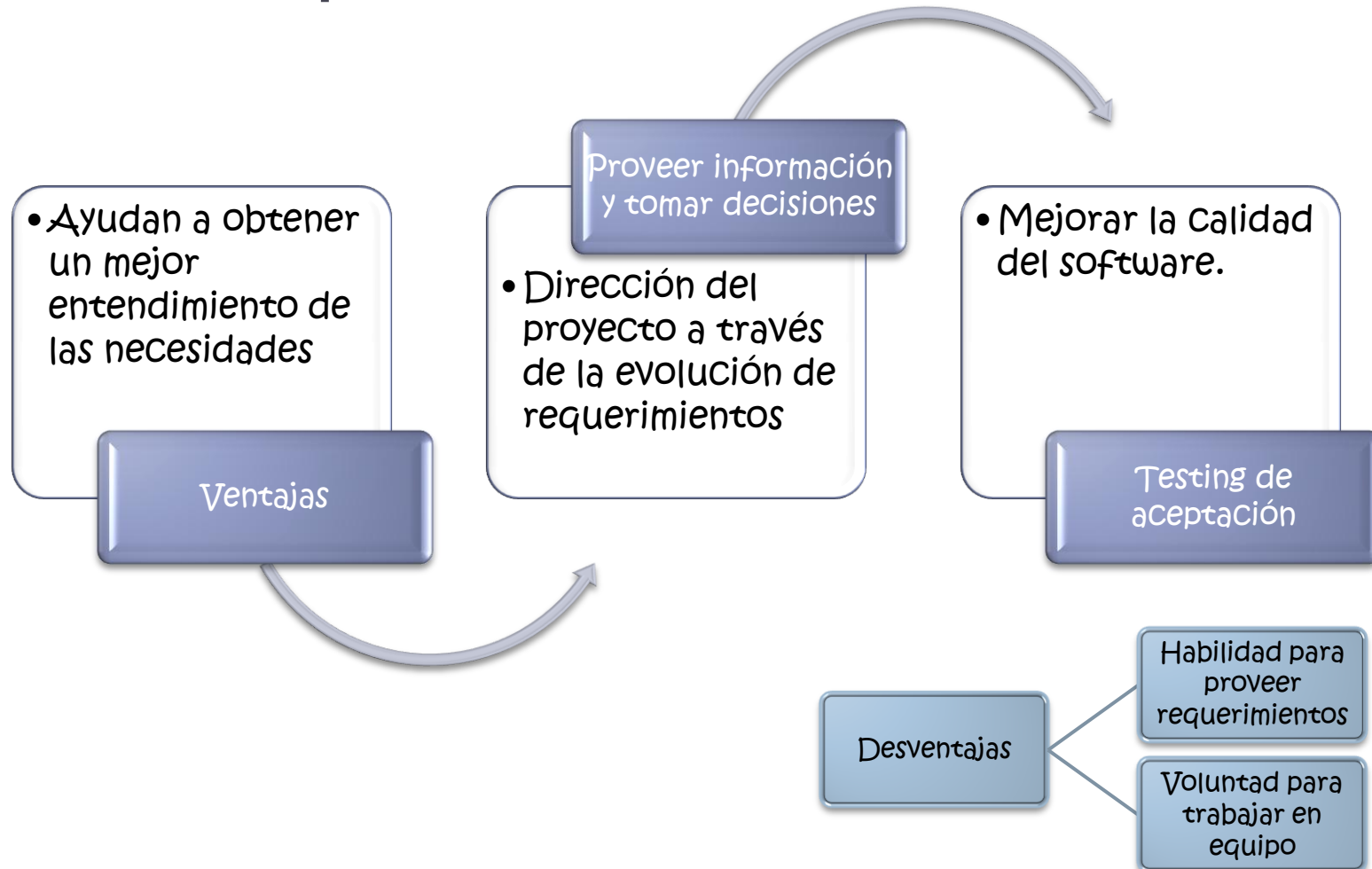
## Requerimientos

- Priorización de requerimientos
- Prever los requerimientos iniciales
- Preferir requerimientos ejecutables a documentación estática
- Requerimientos no funcionales

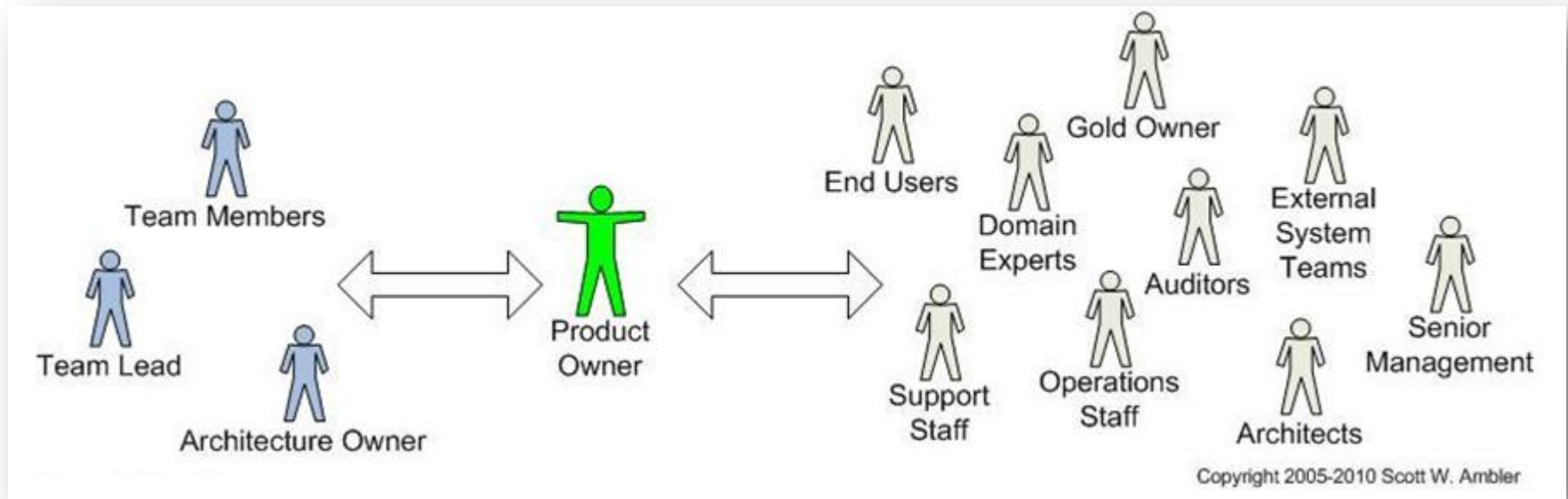
## Modelo iterativo

## Modelo Justo a tiempo (JIT)

# Participación activa de los involucrados



# Reconocer la amplia variedad de stakeholders

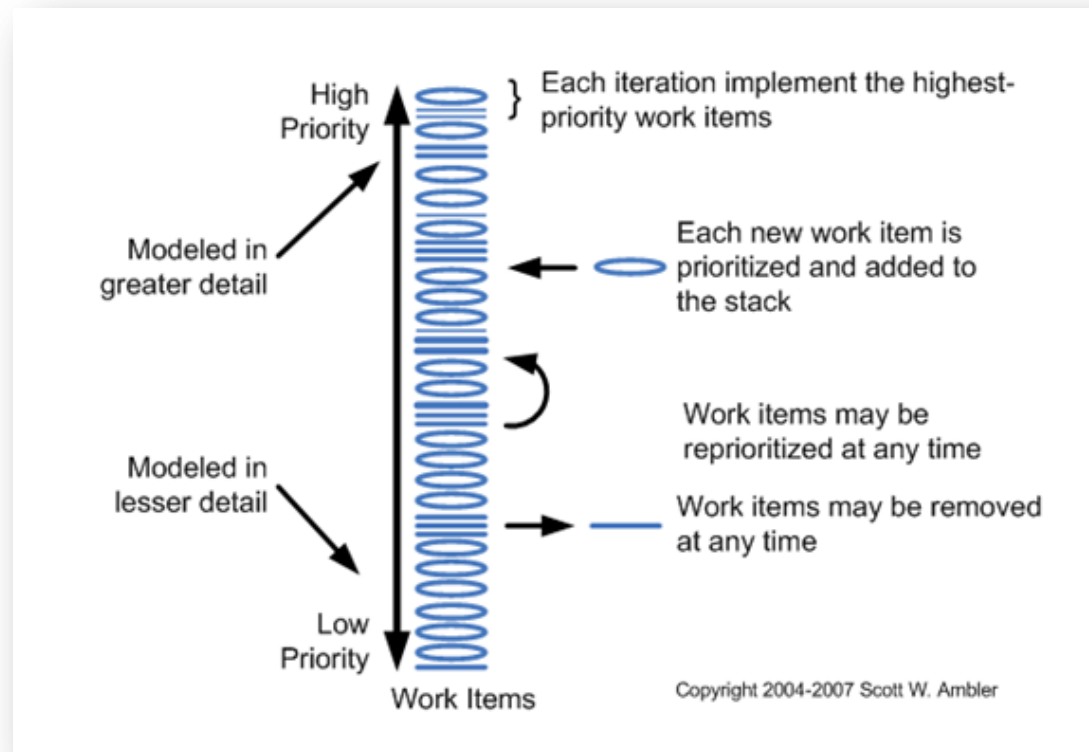


# Adoptar la terminología de los stakeholders

- No se debe forzar una terminología extraña
- Utilizar SU terminología.
- Confeccionar un glosario de términos
  - Incluir términos técnicos y del negocio.
  - Debe estar disponible a todos
    - (Ej: wiki)



# Priorización de requerimientos





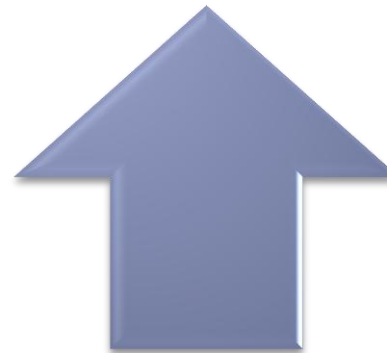
# Preveer los requerimientos iniciales



- El objetivo es modelar lo suficiente para identificar el alcance del sistema y producir una lista inicial de requerimientos.



- El objetivo no es crear una especificación detallada de requerimientos.



# Preveer los requerimientos iniciales - Buenas prácticas

- Modelo de uso
- Modelo de dominio
- Modelo de interfaz de usuario



# Preveer los requerimientos iniciales - Modelo de uso

- ▶ Permite explorar cómo los usuarios van a trabajar con el sistema.

agram.

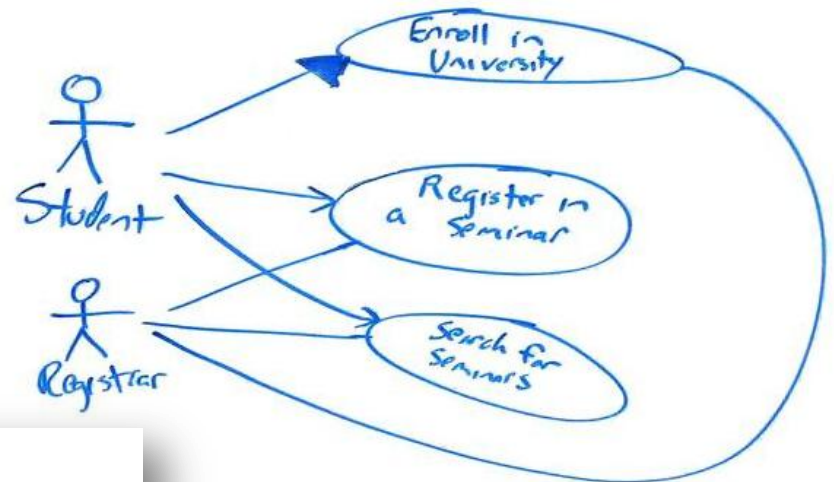


Figure 2. Enroll in seminar as a high-level use case.

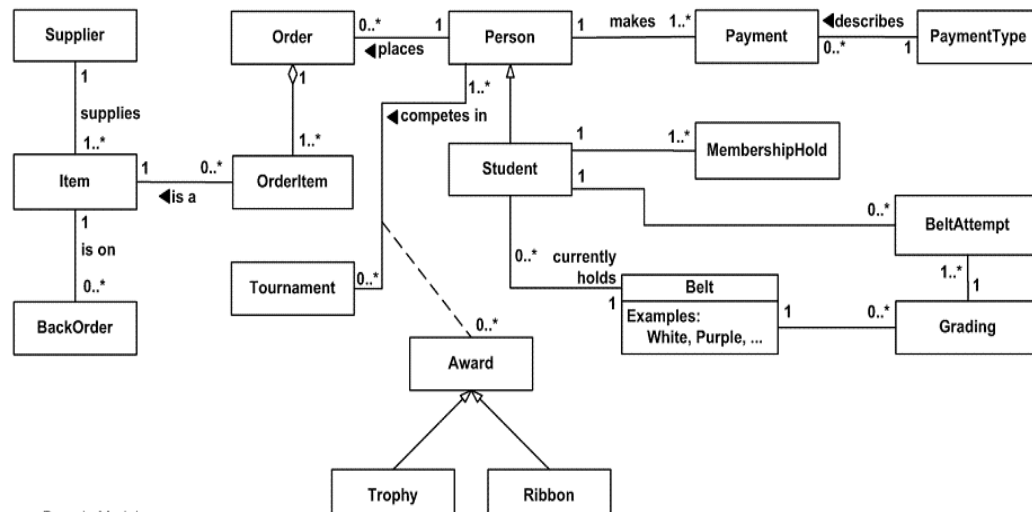
## Enroll in Seminar

- Student chooses a seminar to enroll in
- System checks that the student can enroll in the seminar
- System calculates fees
- Student pays fees and is enrolled

# Preveer los requerimientos iniciales - Modelo de Dominio

- Debe capturar las principales entidades y relaciones del negocio.

Figure 4. The initial domain model.



# Preveer los requerimientos iniciales - Modelos de interfaz

Student Information

Student Number: 789-S07-234

First Name: Scott

Middle: William

Surname: Pinner

Solution: Mr. P

Date First Enrol: June 11 2003

Semesters:

Semester	Term	Att	Status
CSC 100 Intro to CS	Fall 2003	A+	Passed
CSC 200 Intro to AI	Fall 2003	A	Passed
CSC 203 Advanced AI	Spring 2004	-	Enrolled

Add

Before

Transcript

Close

Add a server

Server Number: CSC #

Name: Apple #

Stock

Results

Server	Term	Sets/Ans
CSC 250 Apple Enigmas	Fall 2003	4
CSC 200 Apple Exp	Spring 2005	17
CSC 310 Apple October Enigmas	Spring 2004	8

Course description:

CSC 310 Apple October Teachers

This course describes evolutionary development strictly for data oriented development. See [www.sqribble.net](http://www.sqribble.net) details.

This course currently has 39 people wait for it.

Student number: 789-456-123

First name:

Surname:

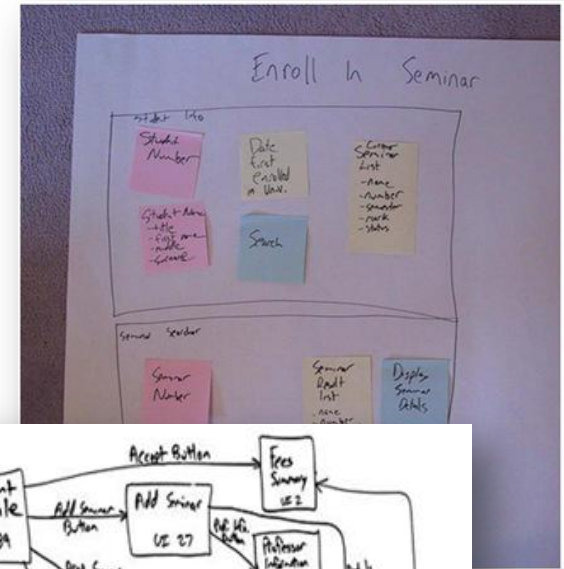
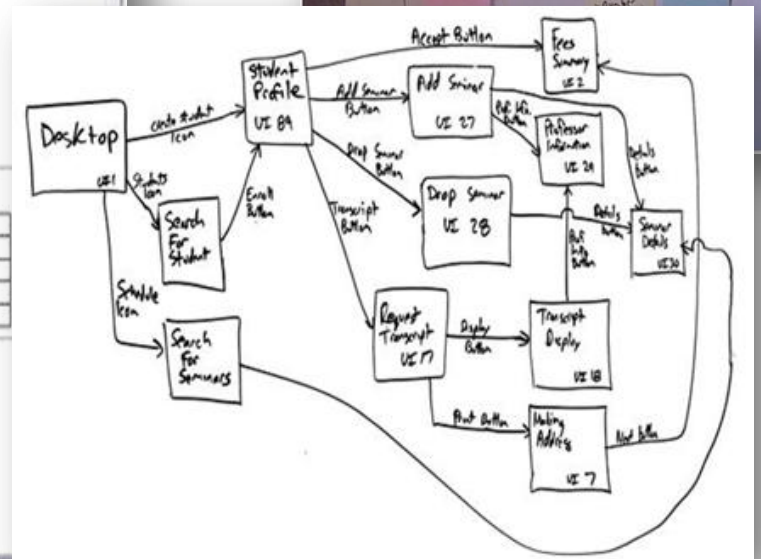
Salutation:

First enrolled: June 14 2003

Schedule:

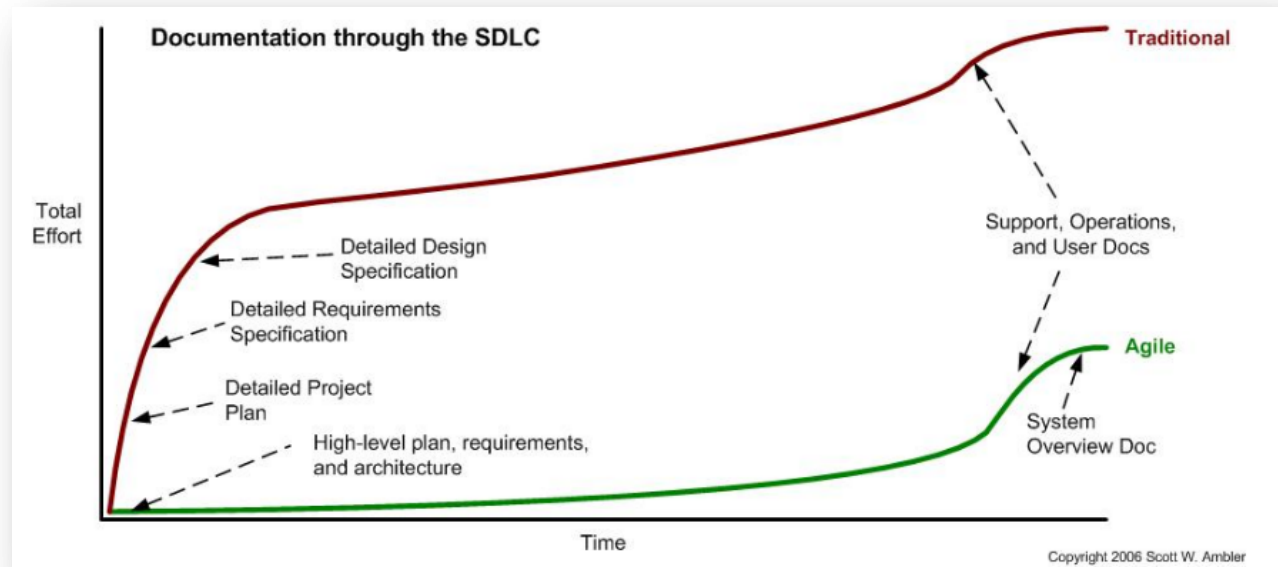
Seminar	Term	Mark	Status
CSC 100 Intro to C#	Fall 2003	A+	Passed
CSC 200 Intro to Agile Modeling	Fall 2003	B-	Passed
CSC 203 Advanced Agile Modeling	Spring 2004	-	Enrolled
CSC 220 Intro to Agile Databases	Spring 2004	-	Enrolled

Buttons:



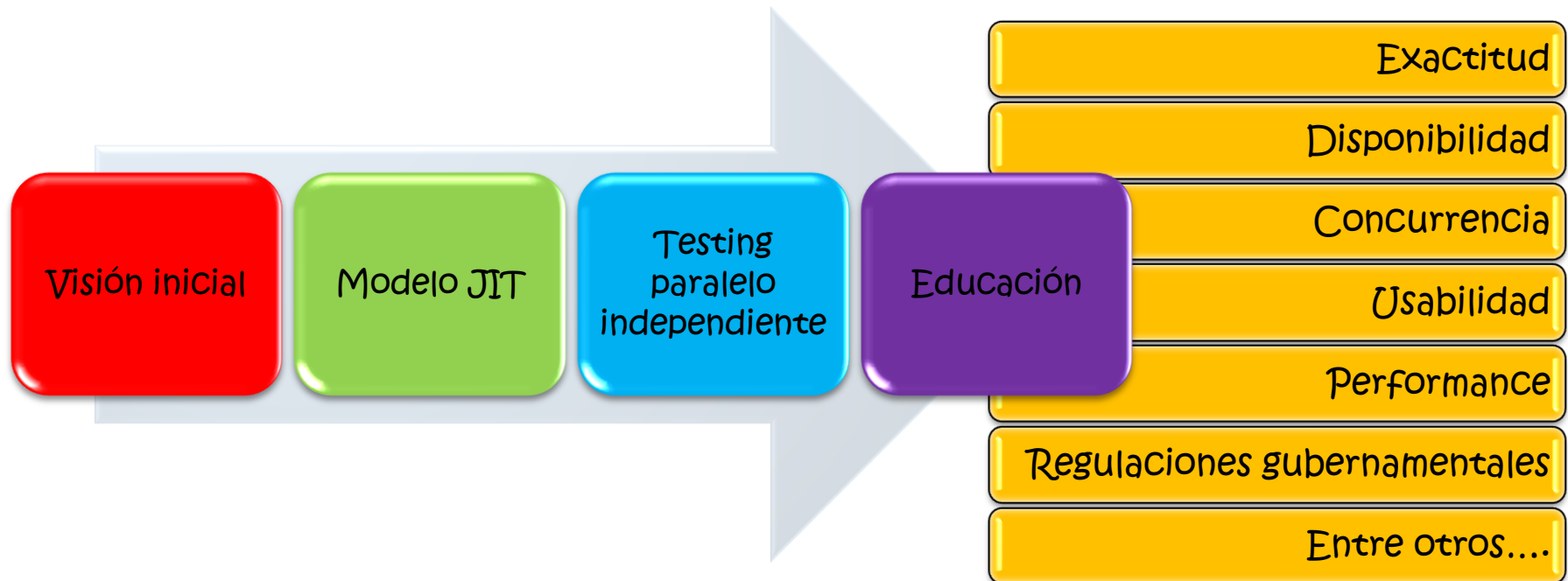
# Preferir requerimientos ejecutables a documentación estática

- Modelado detallado en forma de:
  - Especificaciones ejecutables
  - Tests de aceptación



# Requerimientos no funcionales

- Se enfocan en aspectos que típicamente “atraviesan” a los requerimientos funcionales





# Modelo iterativo

**Se implementan los requerimientos basados en orden de prioridades**

Selección de requerimientos del product backlog

Estimación correcta del esfuerzo por cada requerimiento

**Para poder estimar correctamente es necesario entender el trabajo requerido a ser implementado → modeling**

**Velocidad → medida de cuánto trabajo se completó en la iteración anterior**

Selección de esfuerzo en base a la velocidad anterior

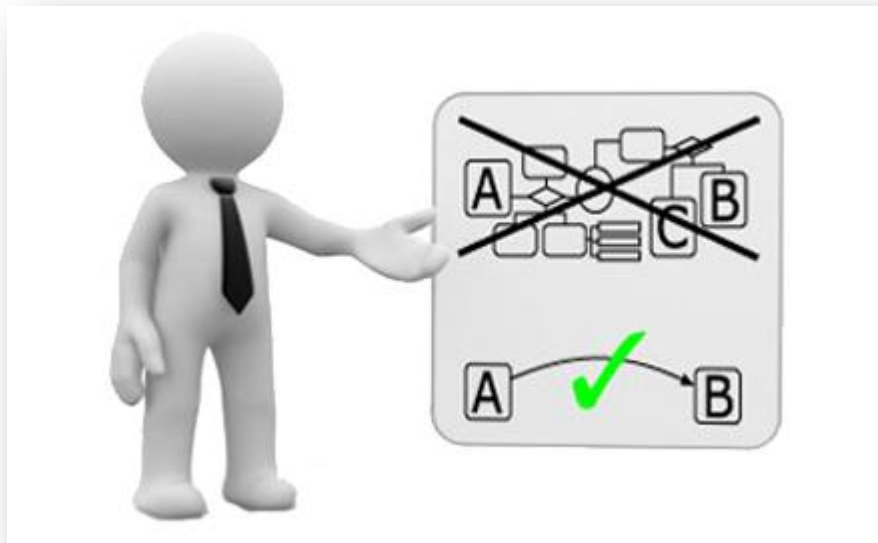
El objetivo es planear correctamente el trabajo para la iteración, identificar los items de trabajo con mayor prioridad e identificar cómo se va a desarrollar

# Modelo Justo a Tiempo (JIT)

## - Model storming -



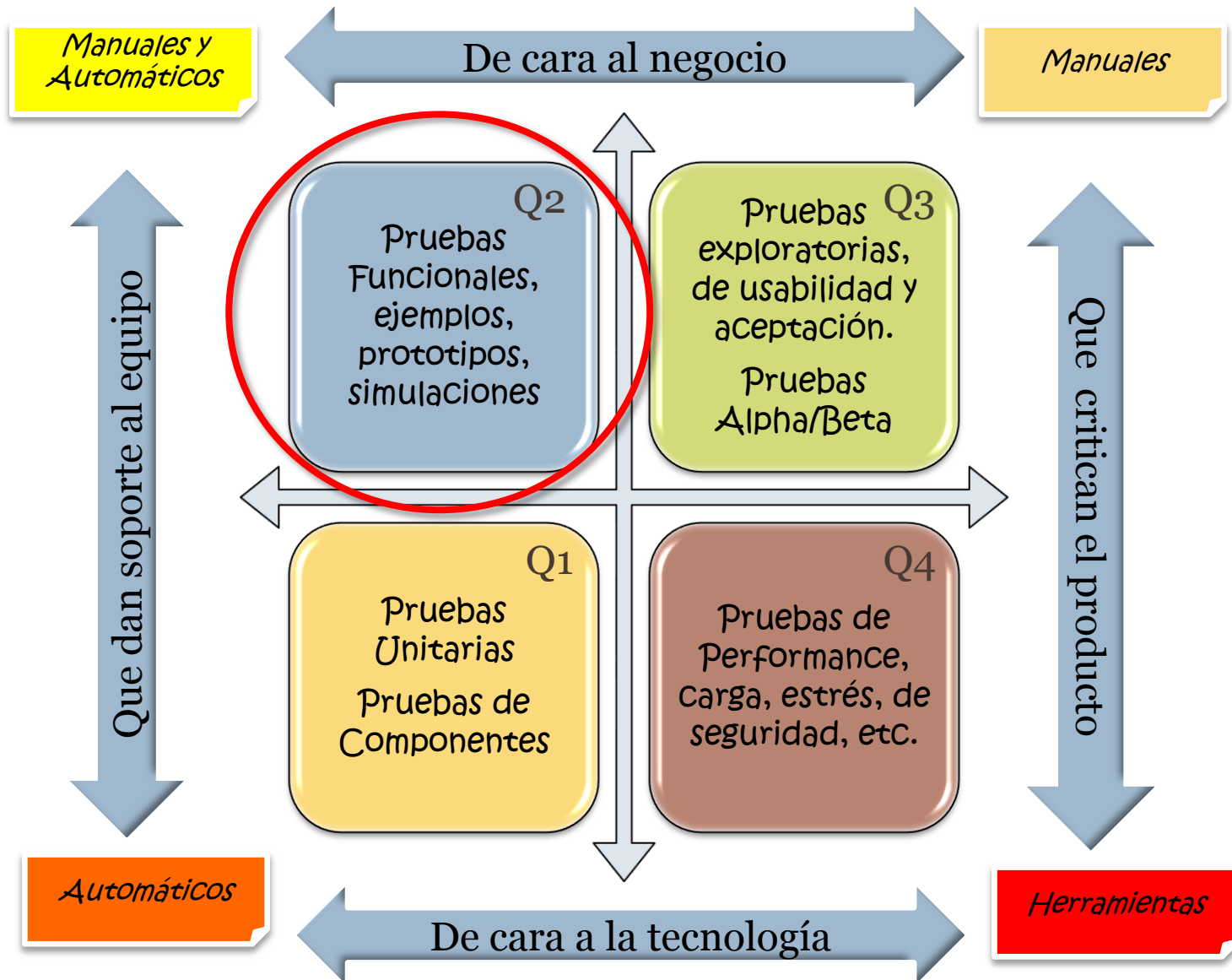
**La razón para modelar es analizar los detalles de un requerimiento**



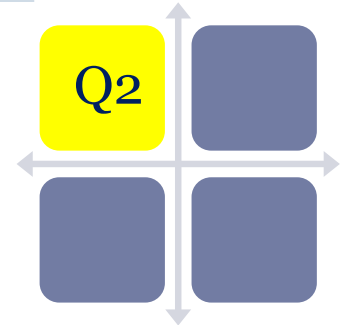
Y ahora que sabemos escribir  
correctamente los requerimientos ...

De qué otra manera podemos  
testearlos?

# Los cuadrantes del Testing Ágil

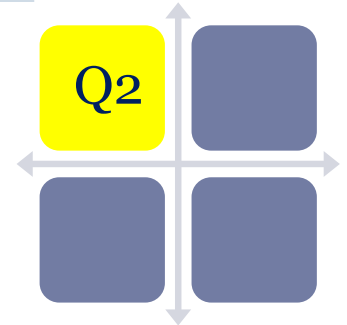


# Beneficios del Q2



- Guía al desarrollo con tests orientados al negocio
- Ayuda a obtener suficientes requerimientos para comenzar el código
- Ayuda a los clientes a alcanzar una claridad sobre el producto
- Captura ejemplos, se expresan como tests ejecutables
- Foco en la calidad externa
- Colaboración Cliente-desarrollador – tester

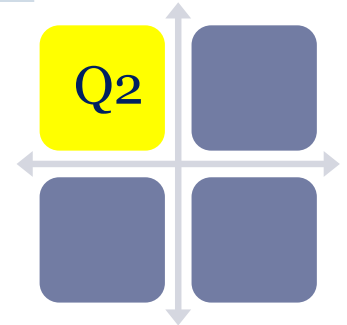
# Q2 - Test Funcional



- Pruebas basadas en el **análisis de las especificaciones** del sistema o de sus componentes
- Se puede llevar a cabo con **dos perspectivas**:
  - Basado en **requerimientos**
  - Basado en **proceso de negocio**



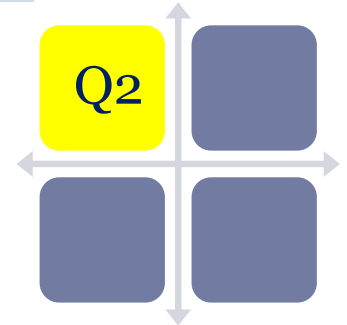
# Q2 - Test Funcional



- Basado en Requerimientos
  - Una buena forma de empezar es:
    - Usar un índice de las especificaciones de requerimientos como una prueba inicial de inventario o **lista de ítems a probar**
  - **Priorizar** los requerimientos basados en Criterio de riesgos y usar esto para priorizar las pruebas

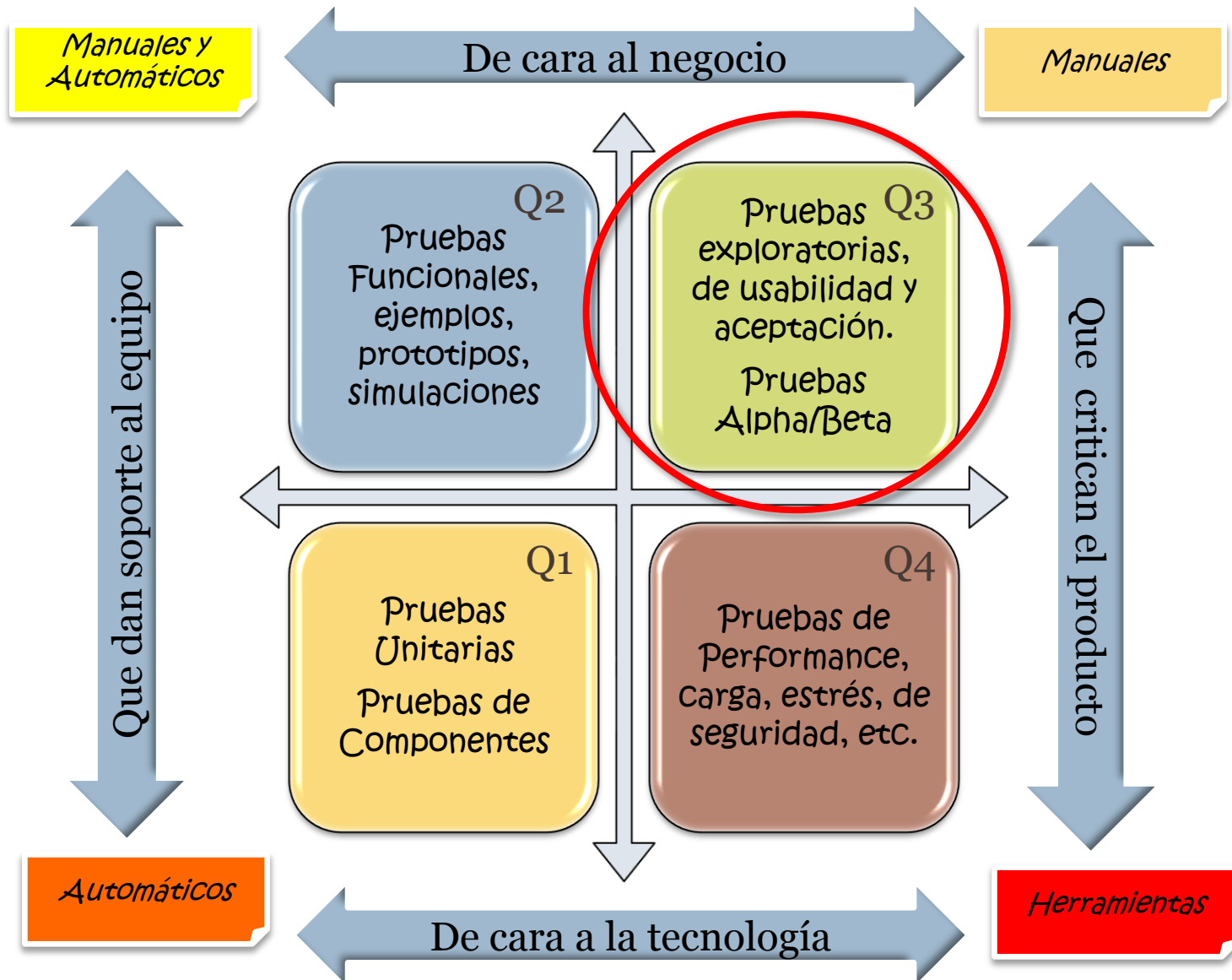


# Q2 - Test Funcional

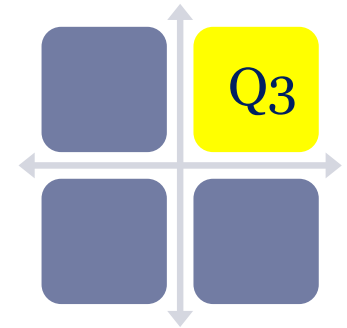


- Basado en proceso de negocio:
  - Usar conocimiento de procesos de negocio
  - Los procesos de negocio describen los escenarios en los que se involucrará el sistema
  - Los casos de uso son la base de los casos de prueba desde la perspectiva de negocio

# Los cuadrantes del Testing Ágil



# Tercer cuadrante (Q3): Testing colaborativo



Provee  
retroalimentación

- Discusiones con el cliente técnico
- Crear nuevos tests a partir de lo aprendido
- Cambiar el proceso cuando sea necesario

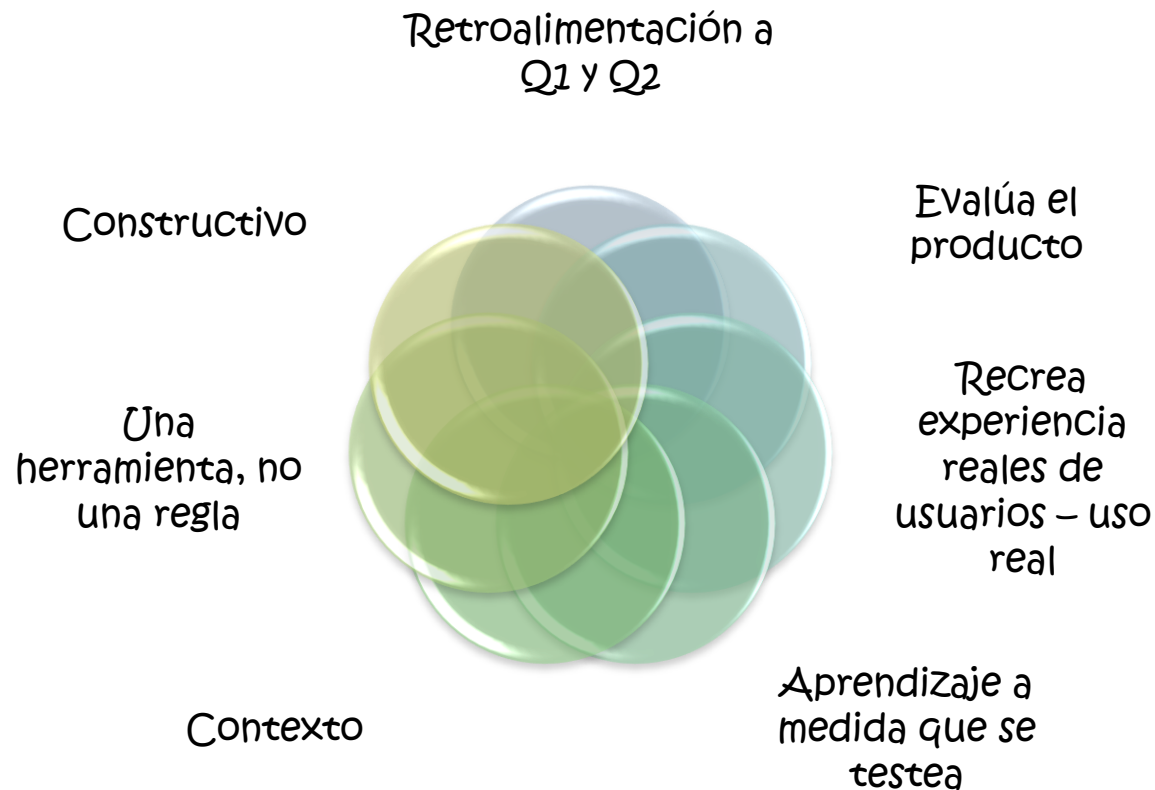
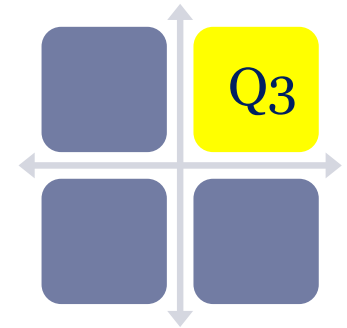
Revisiones de  
iteración

- Confianza en los Builds
- Rápida bucle de retroalimentación

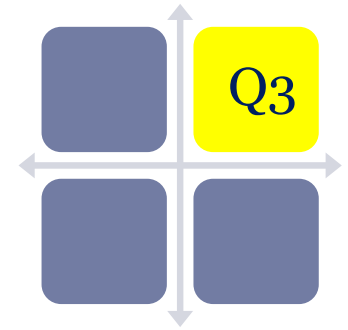
Demos informales

- Testing exploratorio en conjunto con el cliente
- Incluso en código no terminado

# Beneficios del Q3



# Q3 - Testing de Usabilidad



## Persona Testing:

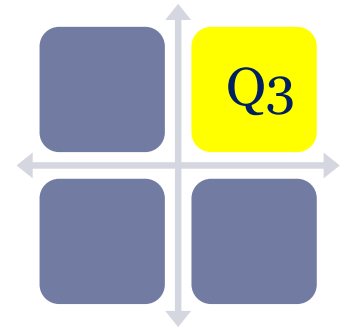
- Crear diferentes tipos de usuarios



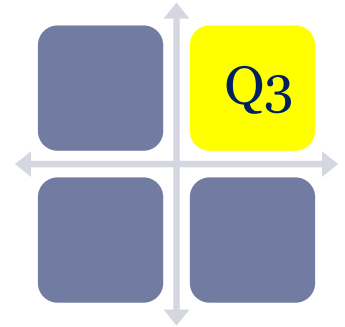
# Q3 - Testing de Usabilidad

Persona Testing:

- Usar personajes



# Q3 - Pruebas de aceptación

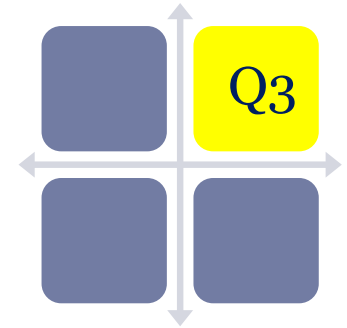


**Objetivo → establecer  
confianza en el sistema**

El ambiente debería ser una representación  
del ambiente de producción



# Q3 - Pruebas de aceptación



## ► Pruebas operacionales

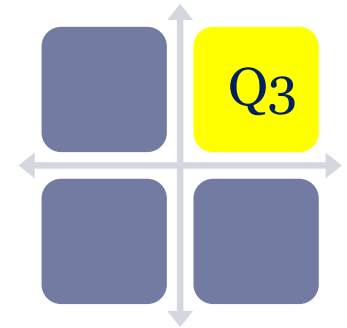
- Valida si el sistema cumple con los requerimientos de operación.

Puede incluir pruebas de:

- Respaldo/restauración
- Recuperación de desastre.
- Tareas de mantenimiento
- Controles periódicos de vulnerabilidades de seguridad



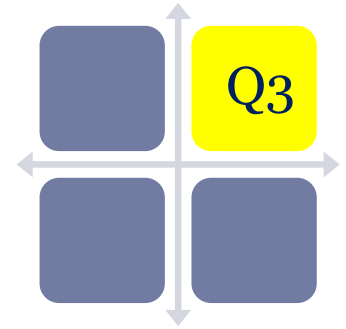
# Q3 - Pruebas de aceptación



## ► Aceptación de usuario

- Se enfoca principalmente en la **funcionalidad y usabilidad** del sistema
- Es **llevada a cabo por usuarios**, gerentes de la aplicación
- Muchas veces hay conexión entre prueba de aceptación y de sistema

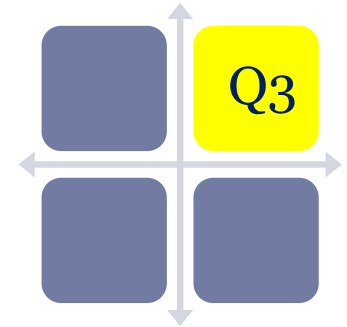
# Q3 - Pruebas de aceptación



## ► Pruebas Alfa

- Se lleva a cabo en el ambiente donde se desarrolló el sistema en un **ambiente controlado**
- Las pruebas **las realizan los usuarios** y desarrolladores del sistema
- Los desarrolladores toman nota de los problemas de los usuarios

# Q3 - Pruebas de aceptación



## ► Pruebas Beta

- Las lleva a cabo el cliente en el ambiente del cliente, **bajo condiciones reales**.
- El usuario envía registros de los incidentes con el sistema.

# Bibliografía

- Ken Schwaber; Scrum Development Process; 1995
- Ken Schwaber and Jeff Sutherland; Scrum Guide; Scrum Alliance; 2010
- Kent Beck; Embracing Change with Extreme Programming; IEEE; 1999
- Brent Barton et al.; Reporting Scrum Project Progress to Executive Management through Metrics; Scrum Alliance; 2005
- Victory Szalvay et al; Agile Transformation Strategy; Danube; 2005
- Jeff Sutherland et al.; Scrum and CMMI Level 5: The Magic Potion for Code Warriors; 2007;
- Mike Cohn; Agile Estimating and Planning ; Prentice Hall; 2006; 0-13-147941-5
- Mary and Tom Poppendieck; Lean Software Development: An Agile Toolkit; Addison-Wesley; 2003; 0-321-15078-3

# Bibliografía

- The Certified Software Quality Engineer Handbook - Linda Westfall
- Foundations of Software Testing: ISTQB Certification - Dorothy Graham, Erik Van Veenendaal, Isabel Evans and Rex Black
- <http://janetgregory.ca/documents/Edmonton-Quadrants.pdf>

# Versión

Versión	Fecha	Comentarios	Autor
1.0.0_Draft_A	17-sep-2012	Versión inicial	Natalia Andriano
1.0.0	28-sep-2012	Baseline	Natalia Andriano
1.0.1	28-sep-2012	Cambios cosméticos a la filmina 8	Natalia Andriano