



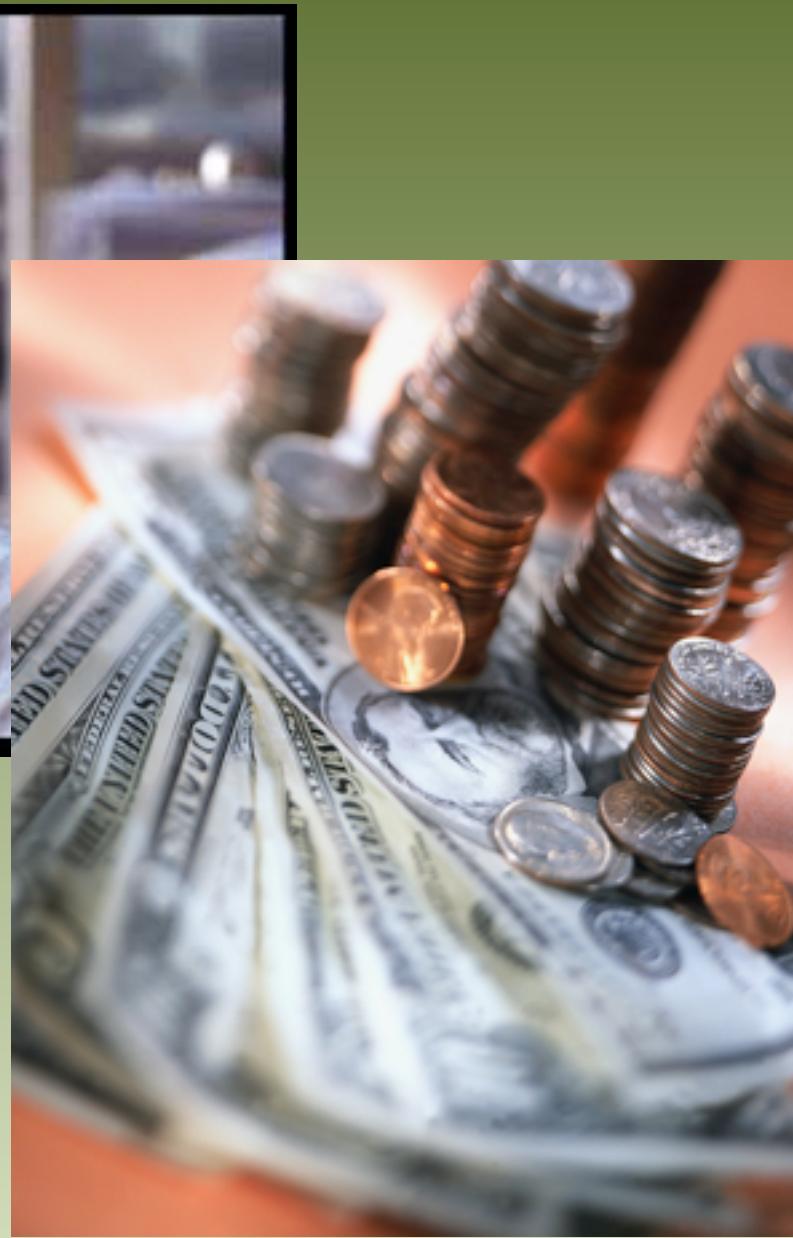
Proceso de desarrollo exitoso

Agile – IDEAL – Lean, DDD, SOLID,
Out-side-in y otros demonios

Manuel Vidaurre  manuel.vidaurre@agiltec.com.mx  @mvidaurre



Manuel Vidaurre  manuel.vidaurre@agiltec.com.mx  @mvidaurre



Manuel Vidaurre  manuel.vidaurre@agiltec.com.mx  @mvidaurre

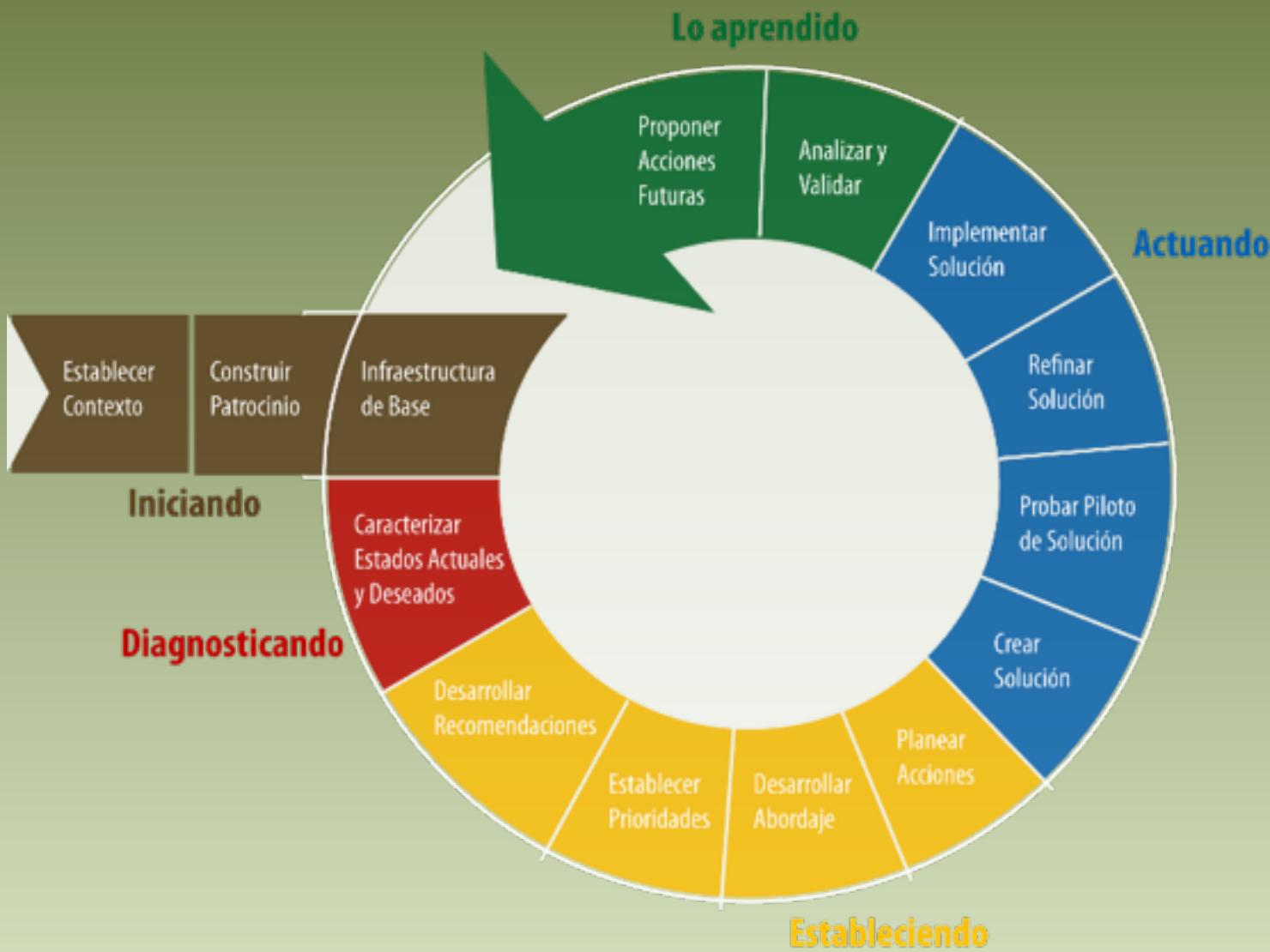


Manuel Vidaurre  manuel.vidaurre@agiltec.com.mx  @mvidaurre



Manuel Vidaurre  manuel.vidaurre@agiltec.com.mx  @mvidaurre

AgilTec usa el proceso IDEAL™ del SEI



Manuel Vidaurre manuel.vidaurre@agiltec.com.mx @mvidaurre

Este proceso nos permite caracterizar su contexto y problemática para encontrar...

El problema de	
afecta	
el impacto del cual es	
una posible solución es	

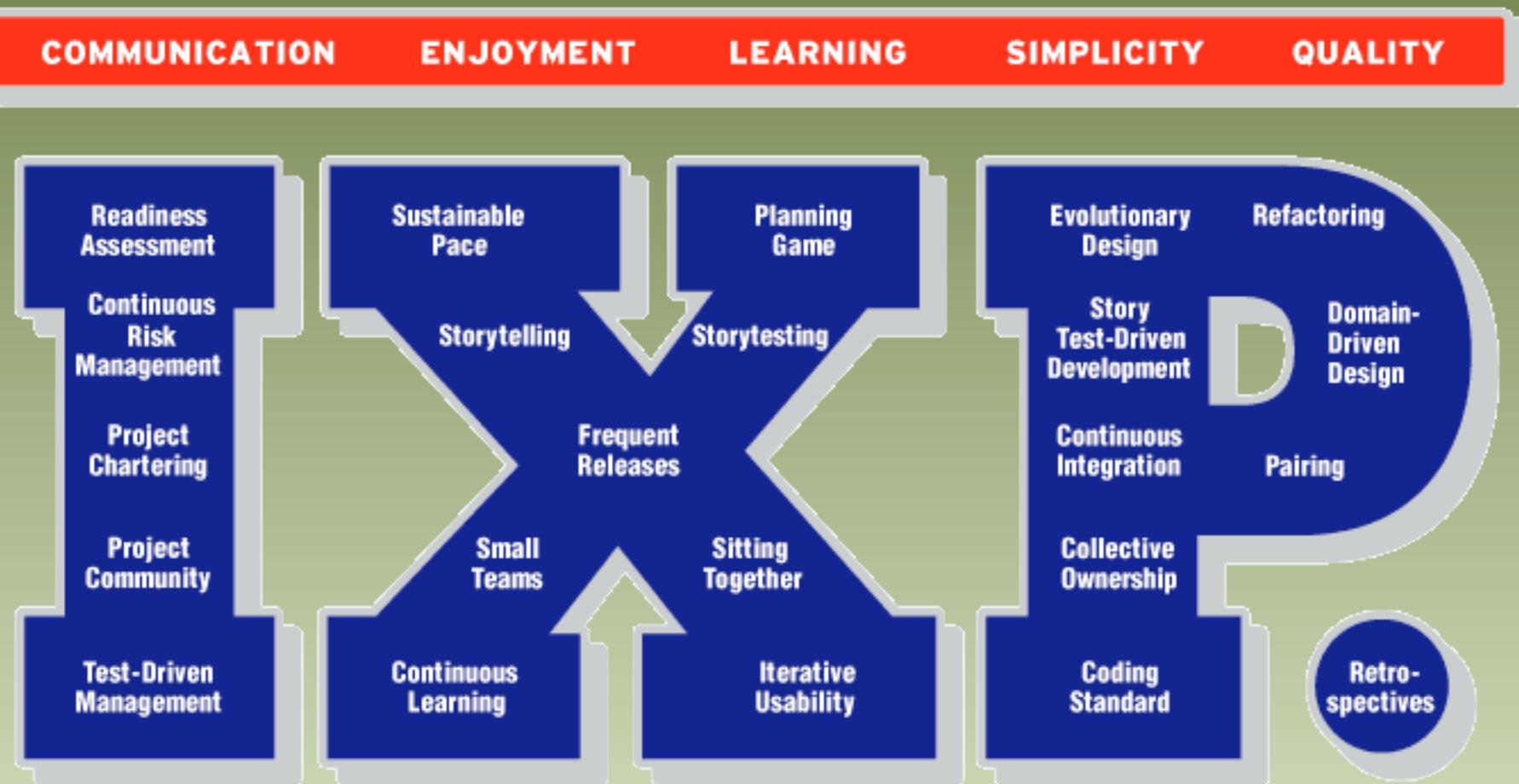
Manuel Vidaurre  manuel.vidaurre@agiltec.com.mx  @mvidaurre

... una solución idonea con el mejor
costo/beneficio del mercado

Para	
<i>La solución propuesta</i>	
que	
nuestro producto	

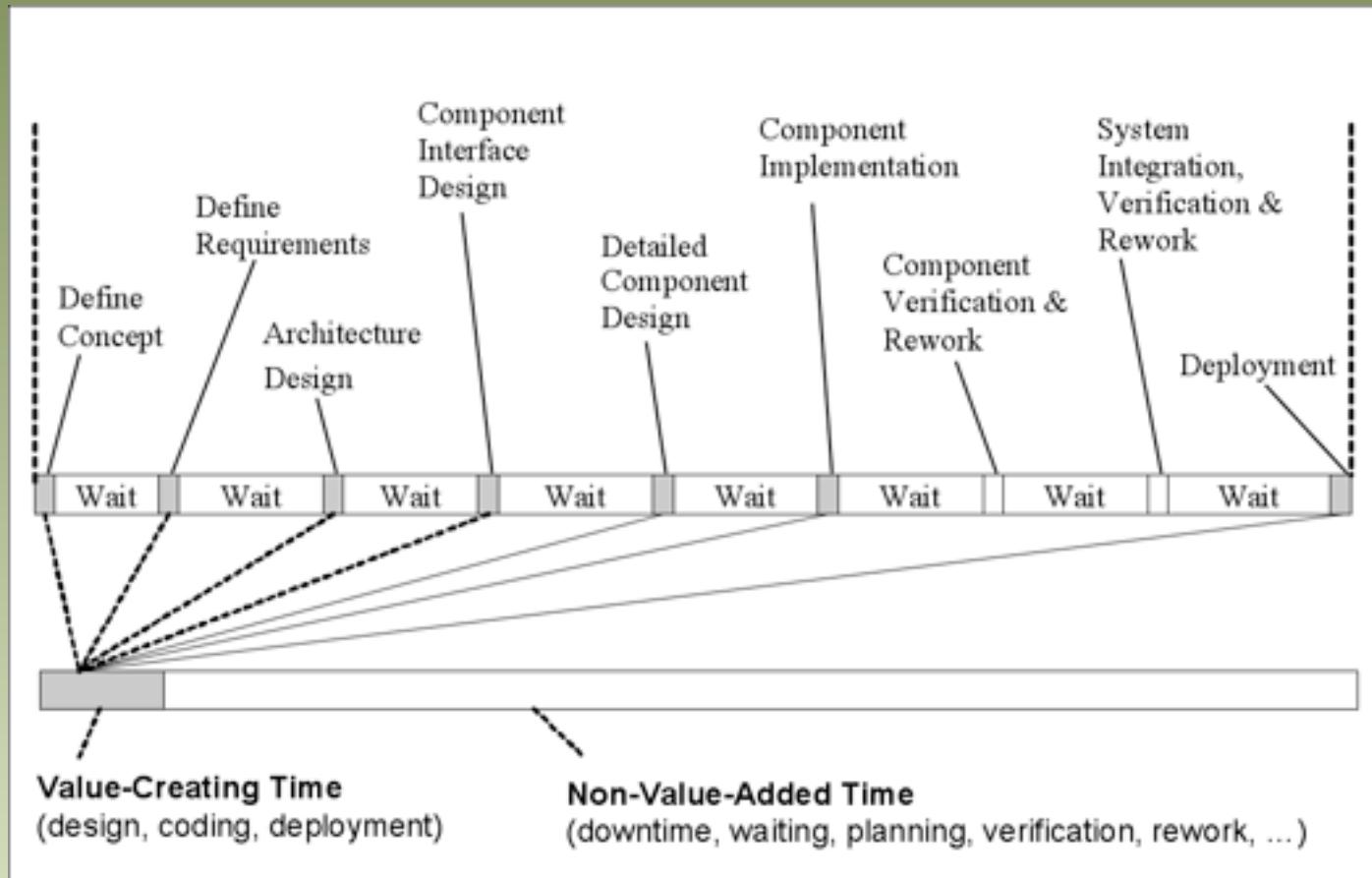
Manuel Vidaurre  manuel.vidaurre@agiltec.com.mx  @mvidaurre

Industrial XP Practices



Manuel Vidaurre manuel.vidaurre@agiltec.com.mx @mvidaurre

Lean – Value Stream Mapping



Manuel Vidaurre manuel.vidaurre@agiltec.com.mx @mvidaurre

The Goal: Value

Sustainable shortest lead time. Best quality and value (to people and society).
Most customer delight, lowest cost, high morale, safety.

Pillar 1: Respect for People

- ▶ don't trouble your customer
- ▶ develop people-then build products
- ▶ no wasteful work
- ▶ teams and individuals evolve their own practices and improvements
- ▶ build partners with stable relationships, trust and coaching lean thinking
- ▶ develop teams

Development Practices

- ▶ long-term great engineers
- ▶ mentoring mgr-eng-teacher
- ▶ cadence
- ▶ cross-functional
- ▶ team room + visual mgmt
- ▶ entrepreneurial chief/ product manager
- ▶ set based concurrent dev.
- ▶ create more knowledge

14 Lean Principles

Long-term philosophy, flow, pull, level workload, stop and fix, master norms, visual controls, tested tech, leaders-teachers from within, develop exceptional people, help partners be lean, go see, consensus and action, learning/reflection/kaizen

Pillar 2: Continuous Improvement

- ▶ Go See
- ▶ kaizen
- ▶ spread knowledge
- ▶ small, relentless,
- ▶ retrospectives
- ▶ 5 whys
- ▶ eyes for waste variability, overburden, NVA, (handoff, WIP, info scatter delay, multitasking, defects, wishful thinking...)
- ▶ perfection challenge
- ▶ Work to flow (smaller batch sizes, low cycle time)

Foundation: Management Support

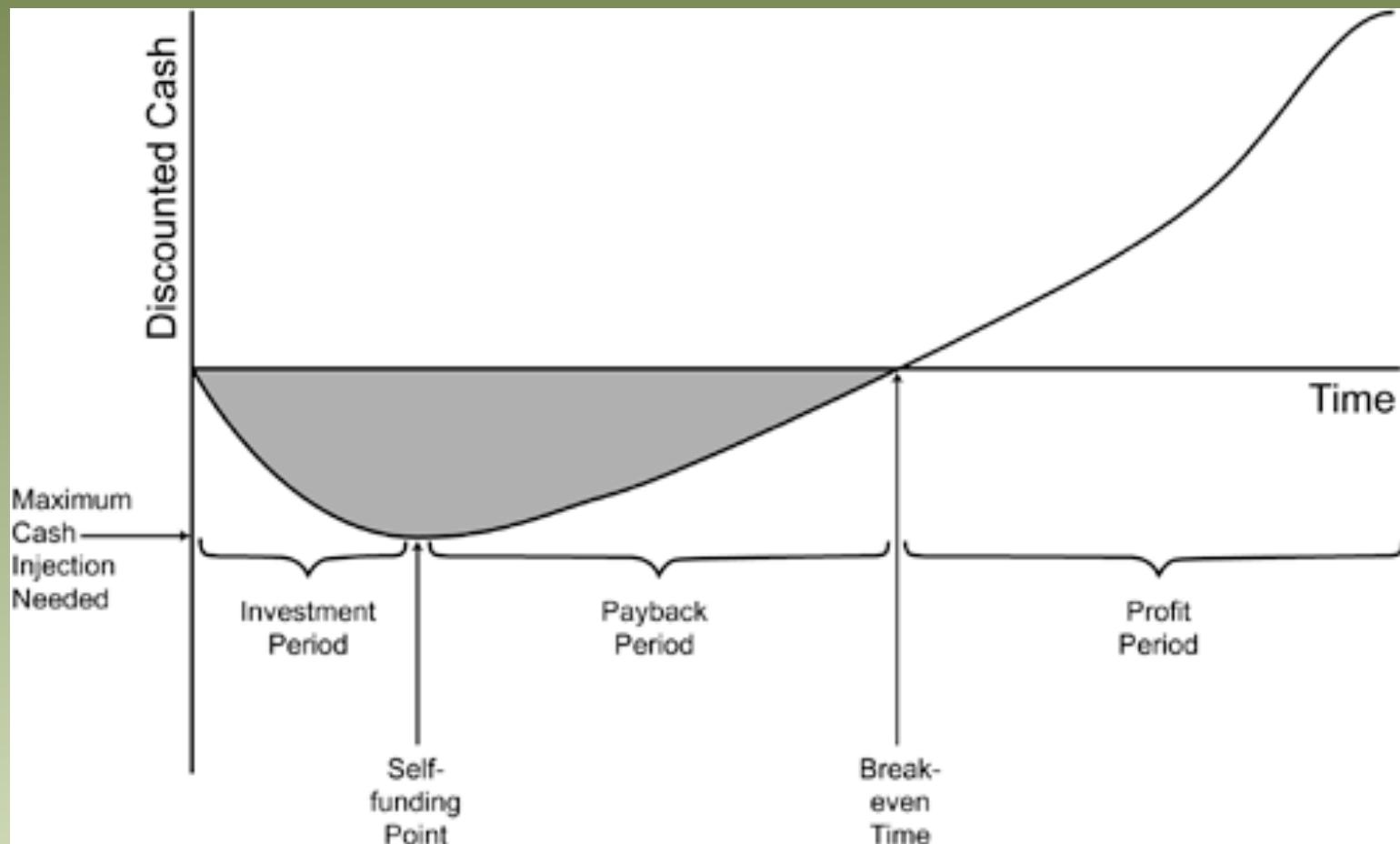
Management applies and teaches lean thinking, and bases decisions on this long-term philosophy

Derived from: Toyota Production System (2004)
Larman and Vodde (2009)

Adapted by Leffingwell, LLC. (2009)

Manuel Vidaurre  manuel.vidaurre@agiltec.com.mx  @mvidaurre

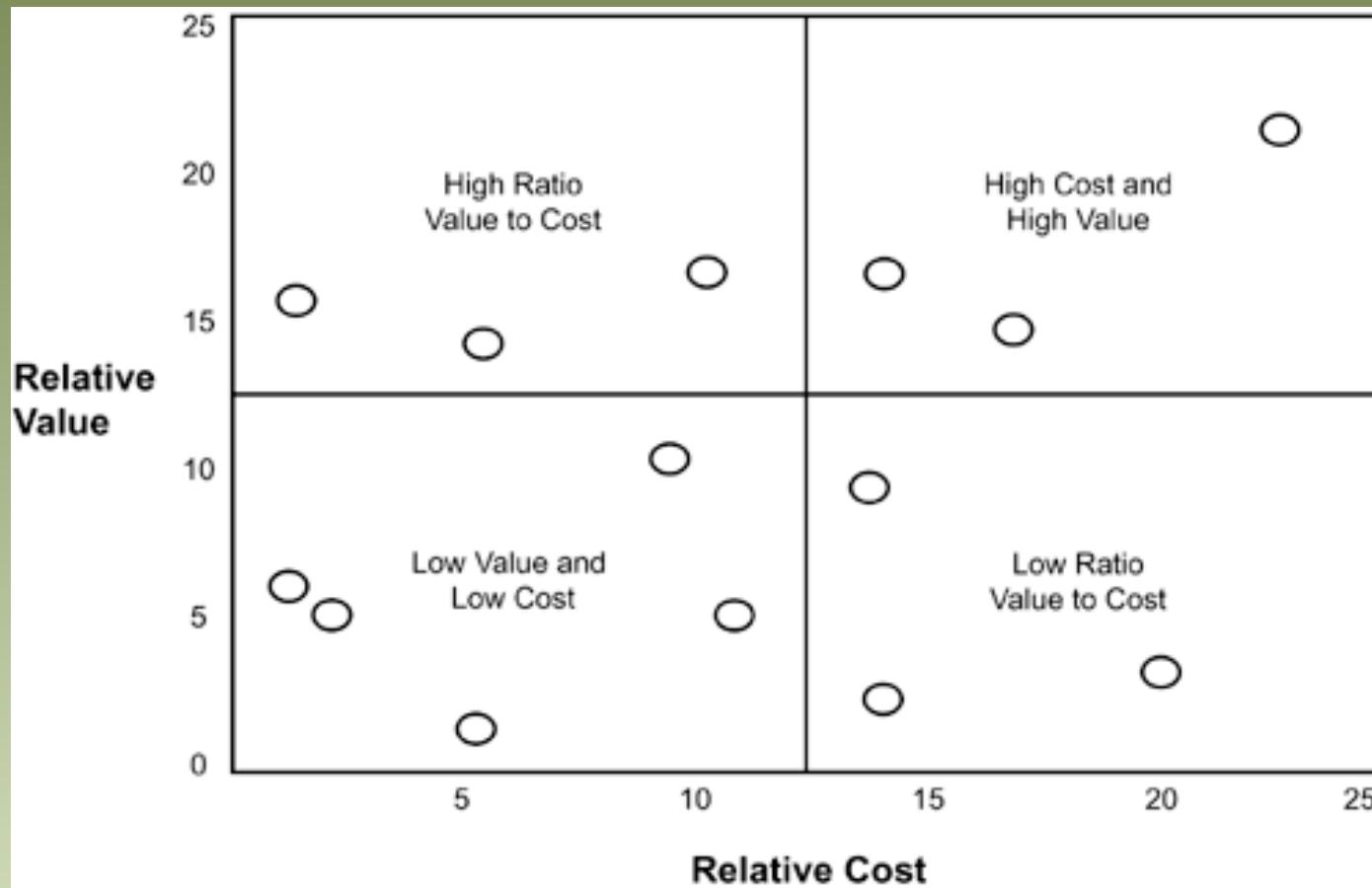
Projecto de Desarrollo Exitoso



Software by Numbers: Low-Risk, High-Return Development. By Mark Denne, Jane Cleland-Huang. Prentice Hall PTR. October 10, 2003

Manuel Vidaurre  manuel.vidaurre@agiltec.com.mx  @mvidaurre

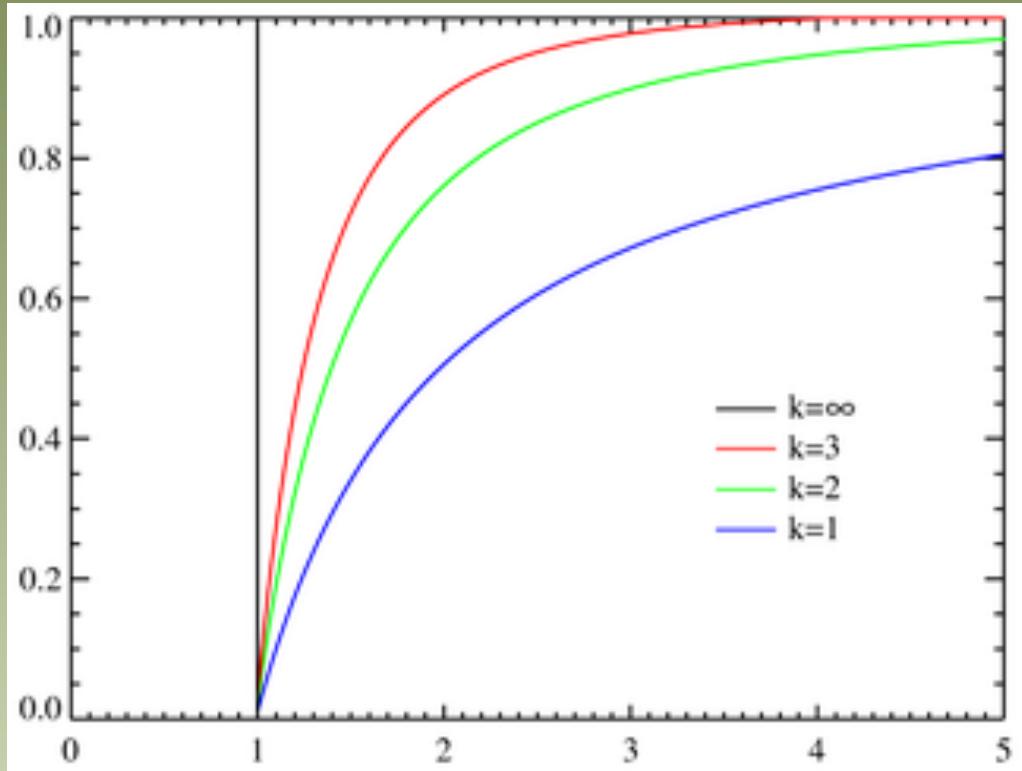
Costo vs Valor en Características mínimas negociables (MMF)



Software by Numbers: Low-Risk, High-Return Development. By Mark Denne, Jane Cleland-Huang. Prentice Hall PTR. October 10, 2003

Manuel Vidaurre manuel.vidaurre@agiltec.com.mx @mvidaurre

Pareto: regla del 80-20



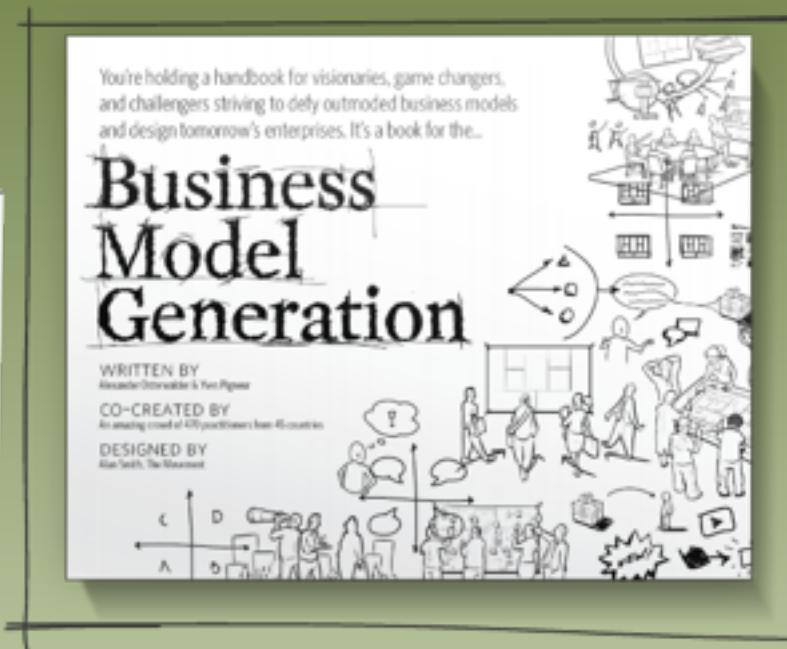
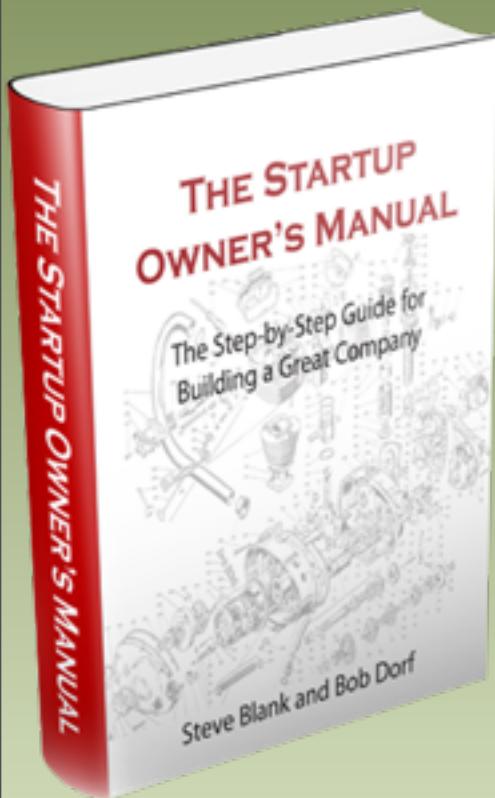
el 80% del valor de negocios radica en el 20% de los procesos

el 80% de los defectos radican en el 20% de los procesos

Software by Numbers: Low-Risk, High-Return Development. By Mark Denne, Jane Cleland-Huang. Prentice Hall PTR. October 10, 2003

Manuel Vidaurre  manuel.vidaurre@agiltec.com.mx  @mvidaurre

Customer Development & Lean Startup



Manuel Vidaurre manuel.vidaurre@agiltec.com.mx @mvidaurre

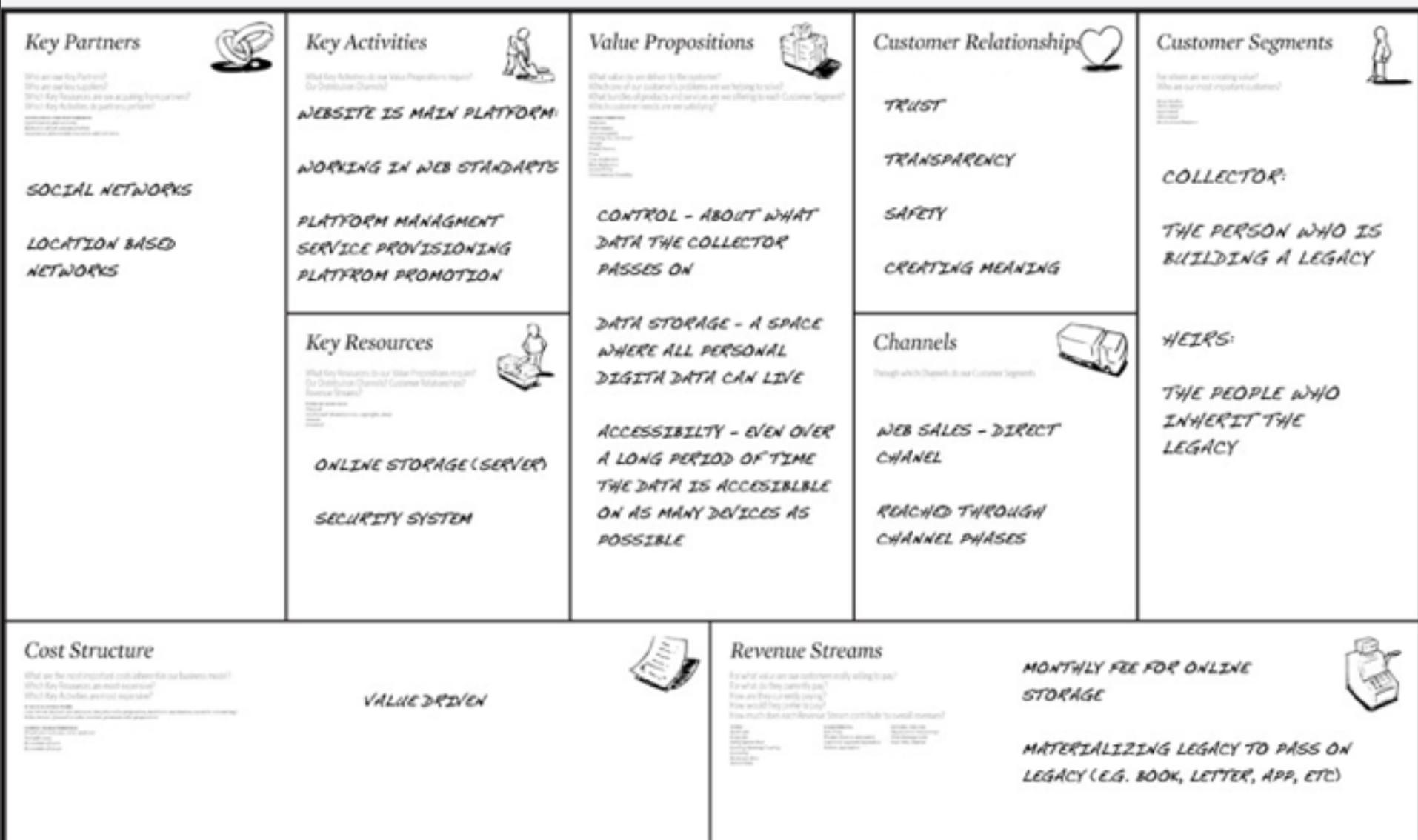
The Business Model Canvas

Designed for:

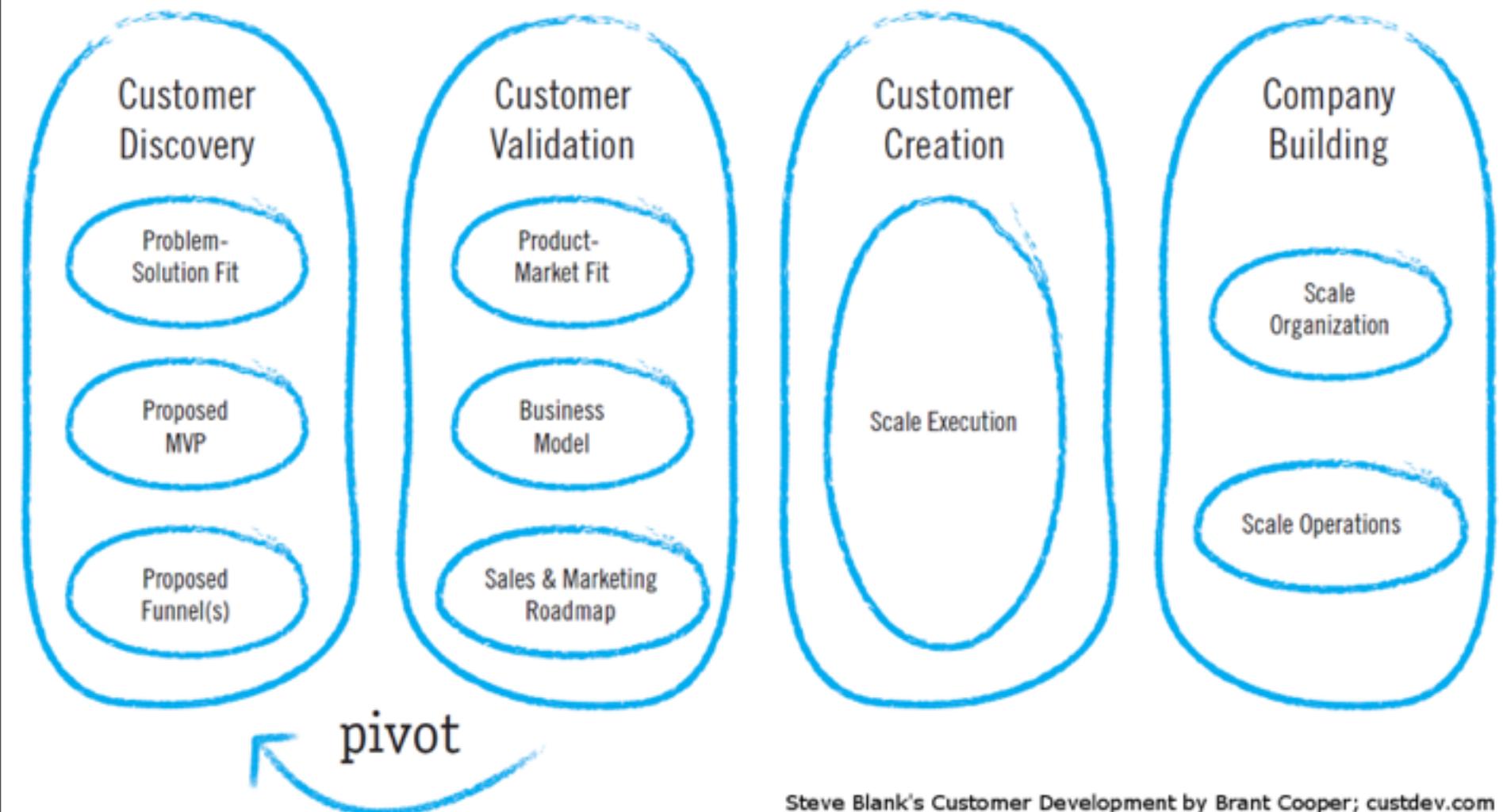
POSTGEIST

Designed by: KRISTIN GRAFE

One
Iteration



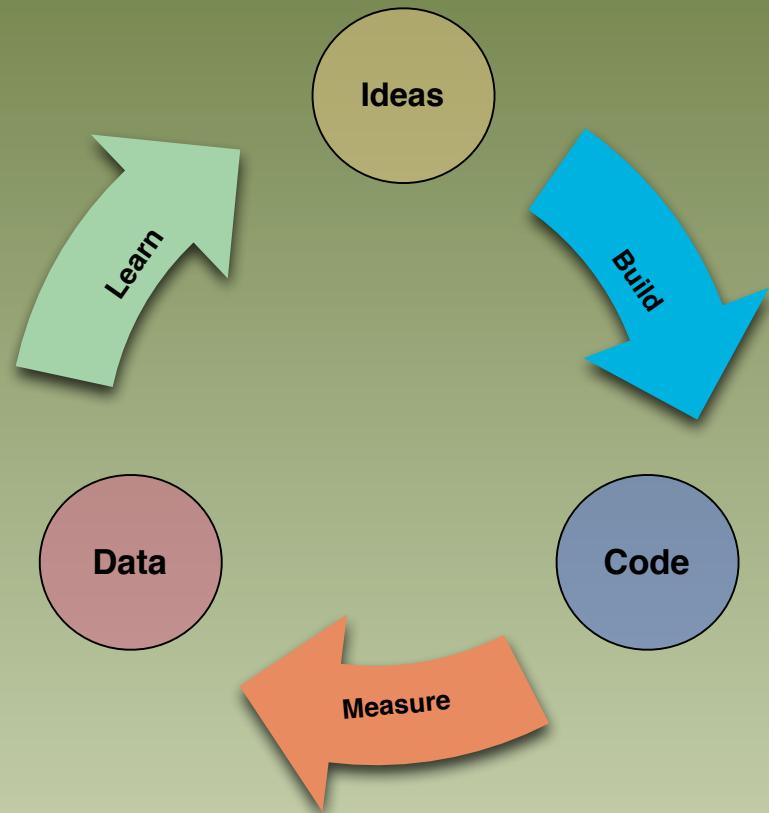
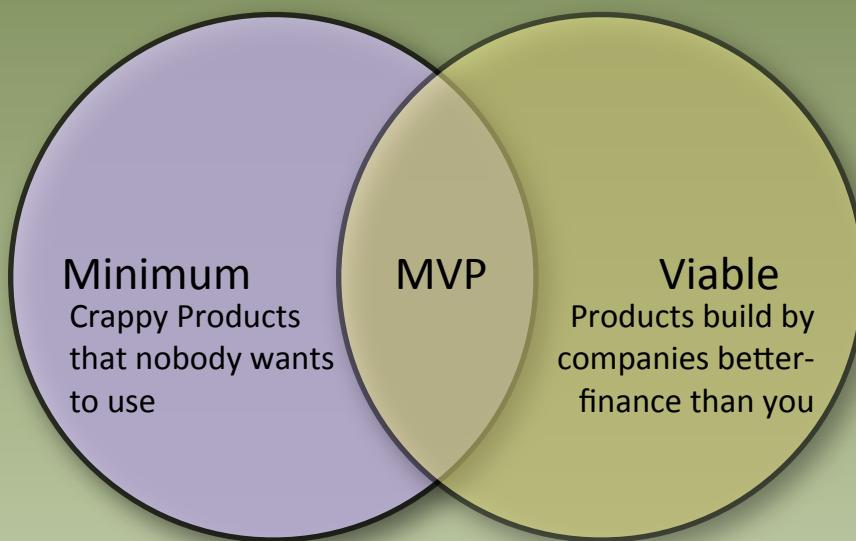
Customer Development



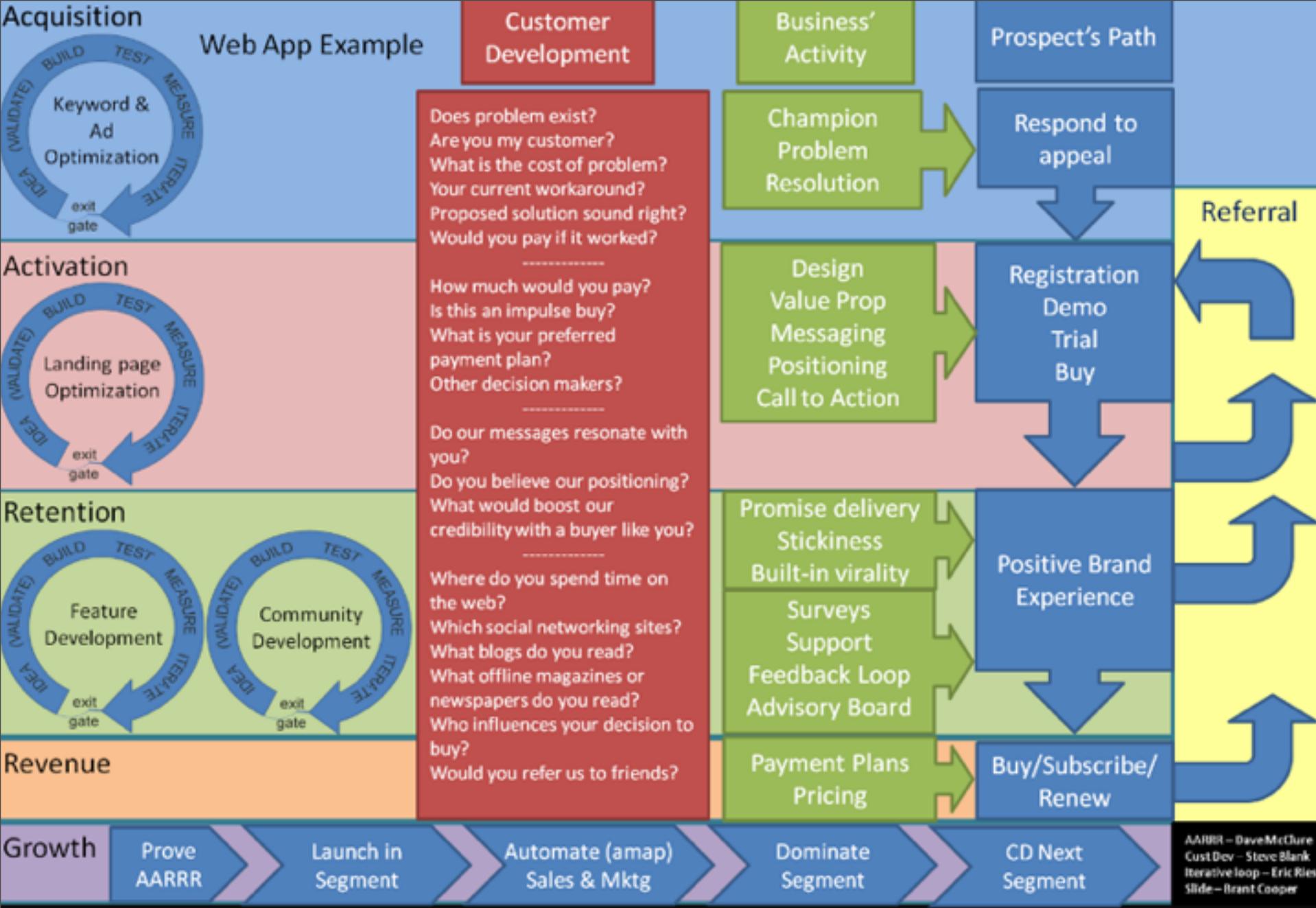
Steve Blank's Customer Development by Brant Cooper; custdev.com

Manuel Vidaurre manuel.vidaurre@agiltec.com.mx @mvidaurre

Minimum Viable Product

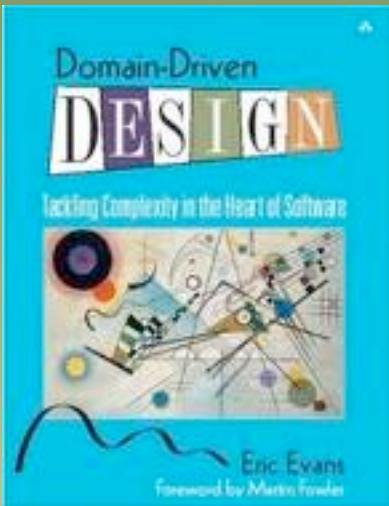


The **minimum viable product** is that version of a new product which allows a team to collect the maximum amount of validated learning about customers with the least effort.

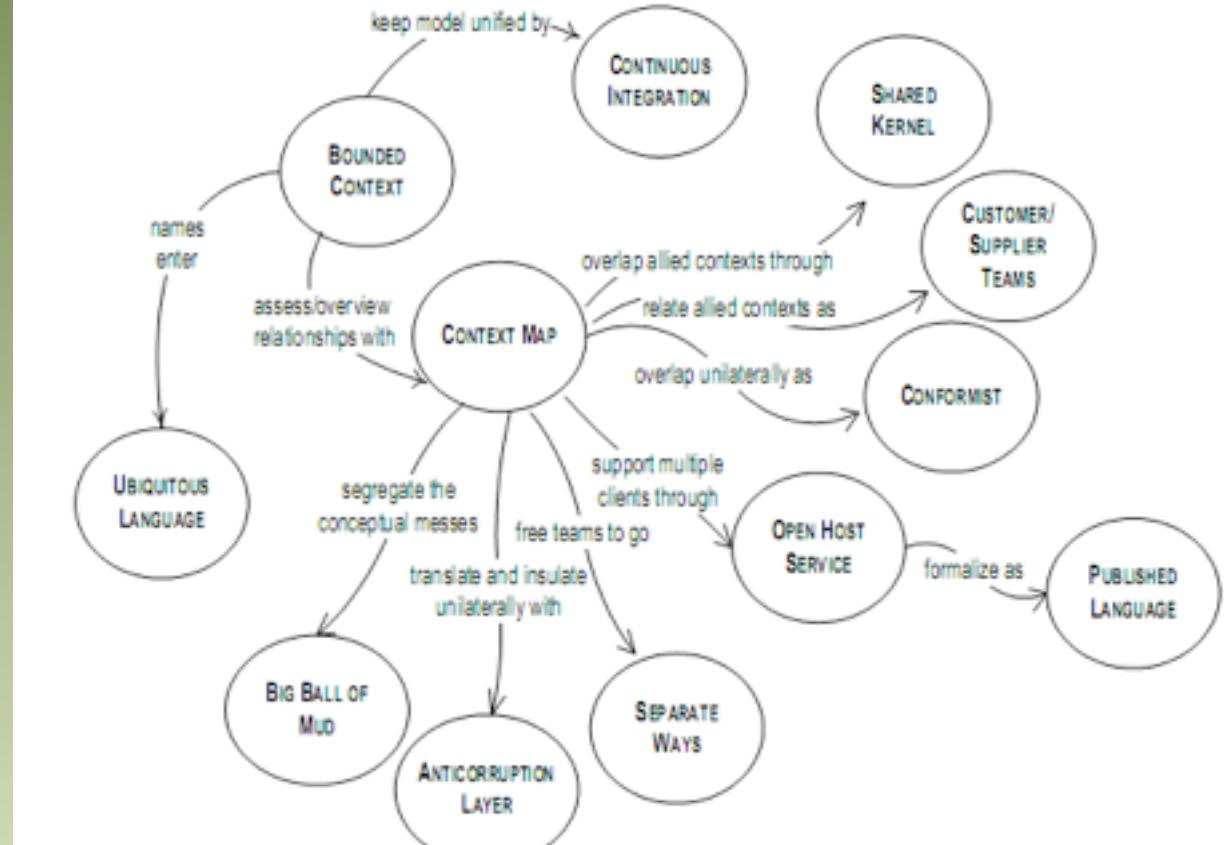


Manuel Vidaurre manuel.vidaurre@agiltec.com.mx @mvidaurre

Domain Driven Design



Maintaining Model Integrity



Manuel Vidaurre manuel.vidaurre@agiltec.com.mx @mvidaurre

Portal La Tranquila: CRM Subsistema & Inventarios

[Done](#) [Current](#) [Backlog](#) [Icebox](#) [View ▾](#) [Actions ▾](#) [Add Story](#)Velocity: 15
project default

Done displaying last 12 iterations (show all)

30	25 Jan	Pts: 0
31	8 Feb	Pts: 0
32	22 Feb	Pts: 0
33	8 Mar	Pts: 0
34	22 Mar	Pts: 0
35	5 Apr	Pts: 0
36	19 Apr	Pts: 0
37	3 May	Pts: 0
38	17 May	Pts: 0
39	31 May	Pts: 0
40	14 Jun	Pts: 0
41	28 Jun	Pts: 0

Current

- 42 12 Jul - Current Pts: 0 of 13 [Finish](#) [Edit](#)
▶ look and feel El look and feel debe de ser basado en CSS. Algunas partes no se despliegan bien en los servidores se requiere revisar (MV)
- ▶ content management FAQ: Diferentes grupos de usuarios deben de poder crear preguntas que sean respondidas por otro usuario del grupo o por él mismo, las respuestas pueden tener comentarios. Todo puede ser clasificando a través de tags (JL)
- ▶ performance Crear un conjunto de servidores para que los browser puedan requerir los assets (javascritps, css) de forma asincrona (MV)
- ▶ content management Añadir flexibilidad al infotape que se presenta abajo del menu (MV)
- ▶ content management Capacidad de WUI (Web User Interface) multi-idioma especialmente: English y Español
- ▶ Versión 16.01: CRM en toda su capacidad, manejo de roles, FAQ y manejo de archivos
- ▶ inventario Soporte para unidades de condo-hotel [Start](#) [Edit](#)

© 2010 Pivotal Labs / [Policies](#) / [Feedback](#)

Manuel Vidaurre manuel.vidaurre@agiltec.com.mx @mvidaurre

Valores de Negocios

- Proteger la **Utilidad**
- Incrementar la **Utilidad**
- Administrar el **Costo**

Técnica de los 5 *Por que?*

- Identificar el Valor de Negocios de los Requerimientos

De afuera hacia adentro (*Outside-in*)

Cliente que tiene una meta con un valor



Manuel Vidaurre  manuel.vidaurre@agiltec.com.mx  @mvidaurre



Manuel Vidaurre  manuel.vidaurre@agiltec.com.mx  @mvidaurre



RSpec

Cucumber



Manuel Vidaurre  manuel.vidaurre@agiltec.com.mx  @mvidaurre

Cucumber

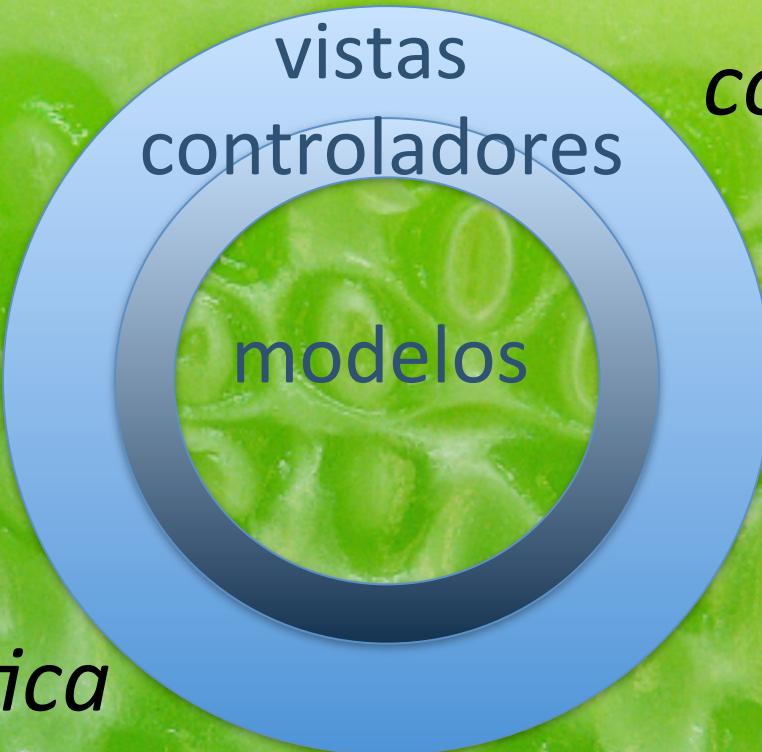
RSpec



Manuel Vidaurre manuel.vidaurre@agiltec.com.mx @mvidaurre

RSpec

“Los
componentes e
interacciones
que desearía
tener”



“La característica
que desearía tener”



Cucumber



Manuel Vidaurre  manuel.vidaurre@agiltec.com.mx  @mvidaurre

RSpec

“Los
componentes e
interacciones
que desearía
tener”

“La característica
que desearía tener”



Cucumber

 © 2006 www.agiletecnology.com

Manuel Vidaurre  manuel.vidaurre@agiltec.com.mx  @mvidaurre

Feature: Sign in

In order to get access to protected sections of the site

As a user

I want to be able to sign in

Scenario: User is not signed up

Given no user exists with an email of "email@person.com"

When I go to the sign in page

And I sign in as "email@person.com/password"

Then I should see "Bad email/login or password"

And I should be signed out



Cucumber



Manuel Vidaurre  manuel.vidaurre@agiltec.com.mx  @mvidaurre

Feature: Sign in

In order to get access to protected sections of the site

As a user

I want to be able to sign in

Feature: [Título de la característica]

In order to [Valor de Negocios]

As a [Role]

I want [la acción que da el valor]



Manuel Vidaurre  manuel.vidaurre@agiltec.com.mx  @mvidaurre

Feature: Sign in

In order to get access to protected sections of the site

As a user

I want to be able to sign in

Feature: [Título de la característica]

In order to [Valor de Negocios]

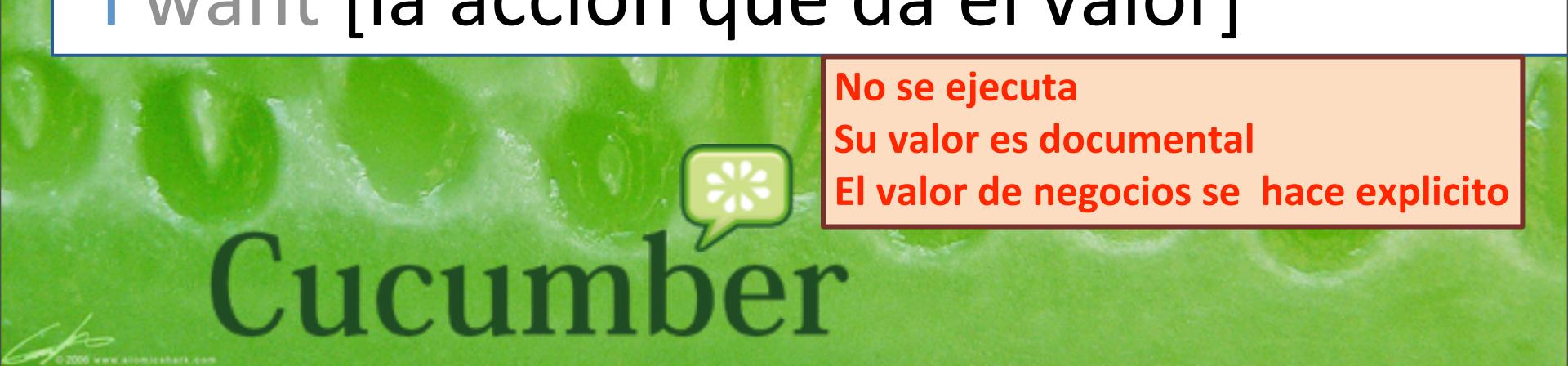
As a [Role]

I want [la acción que da el valor]

No se ejecuta

Su valor es documental

El valor de negocios se hace explícito



Manuel Vidaurre  manuel.vidaurre@agiltec.com.mx  @mvidaurre

Scenario: User is not signed up

Given no user exists with an email of "email@person.com"

When I go to the sign in page

And I sign in as "email@person.com/password"

Then I should see "Bad email/login or password"

And I should be signed out

Scenario: [Título del escenario de uso]

Given [Contexto - Precondición]

When I [do] [Acción]

Then I should [see] [Resultado]



Manuel Vidaurre manuel.vidaurre@agiltec.com.mx @mvidaurre

Scenario: User is not signed up

Given no user exists with an email of "email@person.com"

When I go to the sign in page

And I sign in as "email@person.com/password"

Then I should see "Bad email/login or password"

And I should be signed out

Scenario: [Título del escenario de uso]

Given [Contexto - Precondición]

When I [do] [Acción]

Then I should [see] [Resultado]

Ejecutable mediante los *steps*
Valor de diseño
El resultado esperado es explícito

Cucumber



Manuel Vidaurre manuel.vidaurre@agiltec.com.mx @mvidaurre

RSpec

```
nature: Sign in-  
In order to get access to protected sections of the site-  
As a user-  
I want to be able to sign in-  
  
Scenario: User is not signed up-  
Given no user exists with an email of "email@person.com"-  
When I go to the sign in page-  
And I sign in as "email@person.com/password"-  
Then I should see "Bad email/login or password"-  
And I should be signed out-
```



Cucumber

 © 2006 www.kimjishark.com

Manuel Vidaurre  manuel.vidaurre@agiltec.com.mx  @mvidaurre

Feature: Sign in~
In order to get access to **protected** sections of the site~
As a user~
I want to be able to sign **in~**

RSpec

Scenario: User is not signed up

Given no user exists with an email of "email@person.com"-

When I go to the sign in page-

And I sign in as "email@person.com/password"-

Then I should see "Bad email/login or password"

And I should be signed out-

Step Definition

```

Given /no user exists with an email of "(.*)"$/ do |email|
  assert_nil User.find_by_email(email)
end

When /I sign in( with "remember me")? as "(.*)(.*)"/$ do |remember, email, password|
  When %{I go to the sign in page}-
  And %{I fill in "session_email" with "#{email}"}-
  And %{I fill in "session_password" with "#{password}"}-
  And %{I check "Remember me"} if remember-
  And %{I press "Sign In"}-
end

Then /I should be signed out$/ do-
  assert ! controller.signed_in?
end

```

```
assert ! controller.signed_in?  
end
```

Manuel Vidaurre  manuel.vidaurre@agiltec.com.mx  @mvidaurre

feature: Sign in
In order to get access to **protected** sections of the site
As a user
I want to be able to sign **in**

Scenario: User is **not** signed up

Given no user exists with an email of "email@person.com"

And I sign **in** as "email@person.com/password"
Then I should see "Bad email/login or password"
And I should be signed out

Step Definition

Given /no user exists with an email of "(.*)"\$/ do Iemaili
assert_nil User.find_by_email(email)
end

And %I fill in session_email with "#email"
And %I fill in "session_password" with "#[password]"
And %I check "Remember me" if remember
And %I press "Sign In"
end

Then /I should be signed out\$/ do
assert ! controller.signed_in?
end

Cucumber



Manuel Vidaurre manuel.vidaurre@agiltec.com.mx @mvidaurre

```
nature: Sign in-
In order to get access to protected sections of the site-
As a user-
I want to be able to sign in-

Scenario: User is not signed up-
Given no user exists with an email of "email@person.com"
When I go to the sign in page-
And I sign in
Then I should see "Sign in successful"
And I should see "require File.dirname
```

```
when User exist
insert nil User Fix
d

in API sign inC with
When REd go to the
And REd full in "se
And REd full in "se
And REd check "Name
And REd press "Sign
d

in API should be al
assert : controller
o

describe User do -
  context "A new User" do-
    it "should require email" do-
      lambda{ -
        u = create_user(:email => nil)-
        u.should have_at_least(1).errors_on(:email)-
      }.should_not change(User,:count)-
    end-
    it "should find the new user" do-
      u = create_user(:email => "valid@email.com")-
      u.email.should = User.find_by_email("valid@email.com")-
    end-
  end-
end-
```



Manuel Vidaurre  manuel.vidaurre@agiltec.com.mx  [@mvidaurre](https://www.linkedin.com/in/mvdaurre)

```
require File.dirname(__FILE__) + '/../spec_helper'
```

```
describe User do
  context "A new User" do
    it "should require email" do
```

```
describe [Class] do
  context [Contexto] do
    it [ejemplo de especificación]
      <code>
    end
  end
end
```



Manuel Vidaurre  manuel.vidaurre@agiltec.com.mx  [@mvidaurre](#)

nature: Sign in-
In order to get access to protected sections of the site-
As a user-
I want to be able to sign in-

Scenario: User is not signed up-
Given no user exists with an email of "email@person.com"-
When I go to the sign in page-
And I sign in as "email@person.com/password"-
Then I should see "Bad email/login or password"-
And I should be signed out-

```
on /No user exists with an email of "Cucumber" do
  User.create!(email: "Cucumber")
end

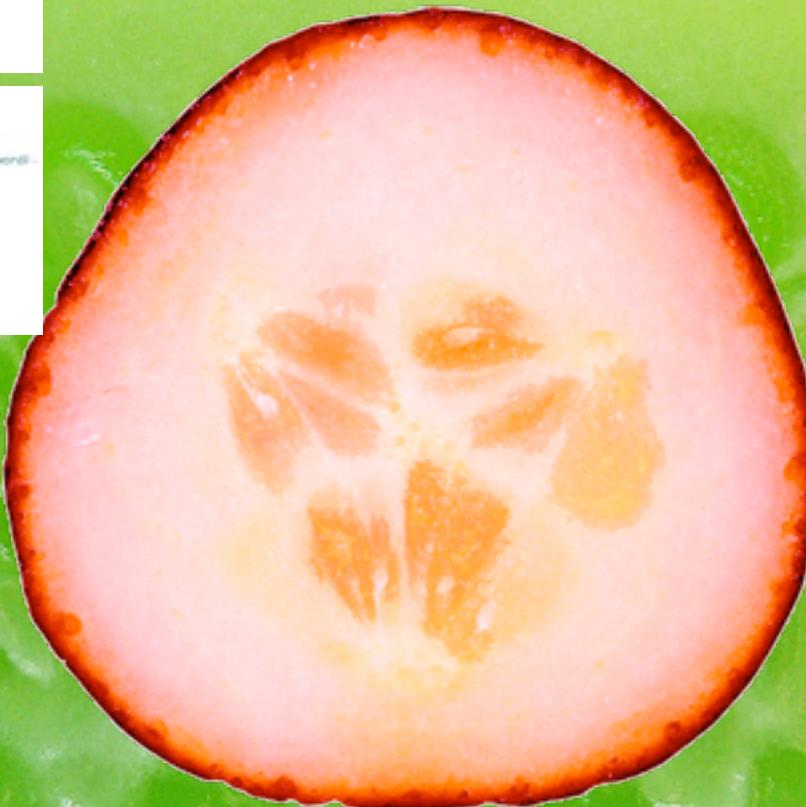
on /Sign in with "remember me"/ do
  visit "/sign_in"
  fill_in "Email", with: "Cucumber"
  fill_in "Password", with: "password"
  check "Remember me"
  click_button "Sign in"
end

on /User should be signed out/ do
  assert !controller.signed_in?
end
```

RSpec

```
require File.dirname(__FILE__) + '/../spec_helper'

describe User do
  context "A new User" do
    it "should require email" do
      lambda {
        u = create_user(:email => nil)
        u.should have_at_least(1).errors_on(:email)
      }.should_not change(user, :count)
    end
    it "should find the new user" do
      u = create_user(:email => "valid@email.com")
      u.email.should == User.find_by_email("valid@email.com")
    end
  end
end
```



 Cucumber

 © 2006 www.atomiqshark.com

Manuel Vidaurre  manuel.vidaurre@agiltec.com.mx  @mvidaurre

nature: Sign in-
In order to get access to protected sections of the site-
As a user-
I want to be able to sign in-

Scenario: User is not signed up-
Given no user exists with an email of "email@person.com"-
When I go to the sign in page-
And I sign in as "email@person.com/password"-
Then I should see "Bad email/login or password"-
And I should be signed out-

```
on /no user exists with an email of "C:\Users\do\email1"-  
assert_nil User.find_by_email(email1)-  
  
on /I sign inC with "remember me" do "C:\Users\do\remember_email, email, password"-  
When I go to the sign in page-  
And I fill in "session_email" with "#{email1}"-  
And I fill in "session_password" with "#{password}"-  
And I check "Remember me" if remember-  
And I press "Sign In"-  
  
on /I should be signed out/ do-  
assert ! controller.signed_in-
```

RSpec

```
require File.dirname(__FILE__) + '/../spec_helper'-  
  
describe User do-  
  context "A new User" do-  
    it "should require email" do-  
      lambda{  
        u = create_user(:email => nil)-  
        u.should have_at_least(1).errors_on(:email)-  
        .should_not change(user,:count)-  
      end-  
    it "should find the new user" do-  
      u = create_user(:email => "valideEmail.com")-  
      u.email.should == User.find_by_email("valideEmail.com")-  
    end-  
  end-  
end-
```

```
class User < ActiveRecord::Base-  
  include Clearance::App::Models::User-  
end-
```



Cucumber



Manuel Vidaurre  manuel.vidaurre@agiltec.com.mx  @mvidaurre

nature: Sign in-
In order to get access to protected sections of the site-
As a user-
I want to be able to sign in-

Scenario: User is not signed up-

Given no user exists with an email of "email@person.com"-
When I go to the sign in page-
And I sign in as "email@person.com/password"-
Then I should see "Bad email/login or password"-
And I should be signed out-

Given there exists with an email of "Cucumber do Test"-

assert_equal User.find_by_email(email).nil

end

When I fill in "remember me" as "Cucumber do Test" do remember, email, password-
And I fill in "the sign in page"-
And I fill in "session_email" with "#{email}"-
And I fill in "session_password" with "#{password}"-
And I check "Remember me" if remember-
And I press "Sign In"-

end

Then I should be signed out- do

assert ! controller.signed_in?

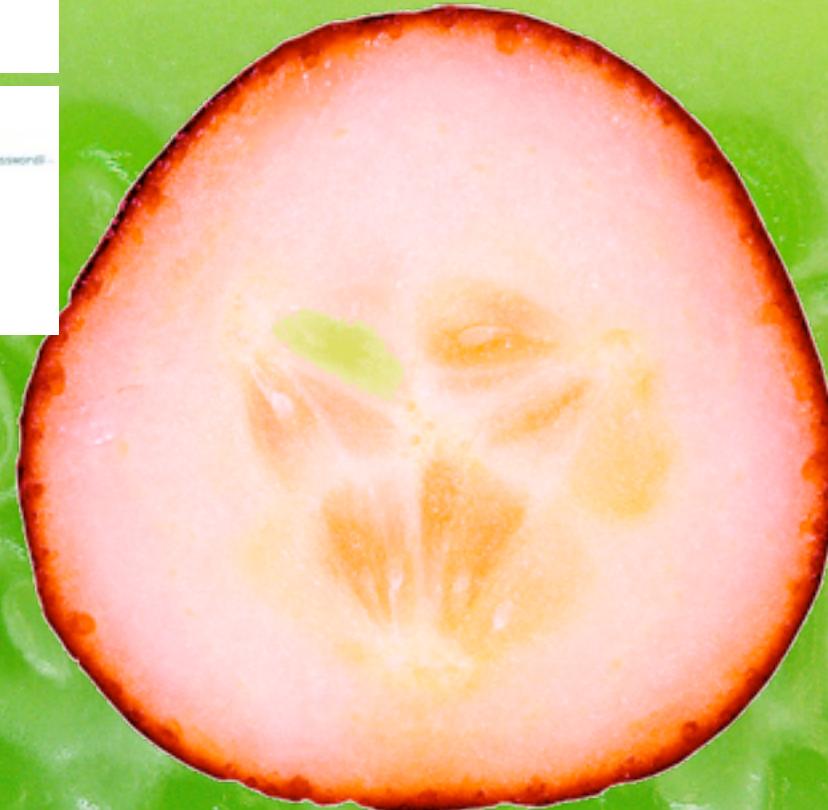
end

RSpec

```
require File.dirname(__FILE__) + '/../spec_helper'
```

```
describe User do
  context "A new User" do
    it "should require email" do
      lambda{
        u = create_user(:email => nil)
        u.should have_at_least(1).errors_on(:email)
      }.should_not change(user,:count)
    end
    it "should find the new user" do
      u = create_user(:email => "valideEmail.com")
      u.email.should == User.find_by_email("valideEmail.com")
    end
  end
end
```

```
class User < ActiveRecord::Base
  include Clearance::App::Models::User
end
```



Cucumber

 © 2006 www.atmosphereshark.com

Manuel Vidaurre  manuel.vidaurre@agiltec.com.mx  @mvidaurre

nature: Sign in-
In order to get access to protected sections of the site-
As a user-
I want to be able to sign in-

Scenario: User is not signed up-

Given no user exists with an email of "email@person.com"-
When I go to the sign in page-
And I sign in as "email@person.com/password"-
Then I should see "Bad email/login or password"-
And I should be signed out-

Given there exists with an email of "Cucumber" do |email|-

insert(:user, :email => email)-

-

When I fill in "remember me" as "Cucumber" do |remember_email, password|-
When I go to the sign in page-
And I fill in "session_email" with "#{email}"-
And I fill in "session_password" with "#{\$password}"-
And I check "Remember me" if remember-
And I press "Sign In"-

Then I should be signed out-
do
assert ! controller.signed_in?

-



Cucumber



Manuel Vidaurre  manuel.vidaurre@agiltec.com.mx  @mvidaurre

RSpec

```
describe User, "A new user" do
  it "should be created" do
    lambda{
      user = create_user -
      user.should_not_be_new_record -
    }.should_change(User,:count).by(1)-
  end-
  it "should require login" do
    lambda{
      u = create_user(:login => nil)-
      u.should have_at_least(1).errors_on(:login)-
    }.should_not_change(user,:count)-
  end-
  it "should require password" do
    lambda{
      u = create_user(:password => nil)-
      u.should have_at_least(1).errors_on(:password)-
    }.should_not_change(user,:count)-
  end-
  it "should require password confirmation" do
    lambda{
      u = create_user(:password_confirmation => nil)-
      u.should have_at_least(1).errors_on(:password_confirmation)-
    }.should_not_change(user,:count)-
  end-

```

```
class User < ActiveRecord::Base
  include Clearance::App::Models::User-
  is_gravatar # to support gravatar associated to the email for the user-
  has_and_belongs_to_many :roles-
  has_one :vendor-
  belongs_to :creator, :class_name => "User", :foreign_key => "created_by"-
  belongs_to :updater, :class_name => "User", :foreign_key => "updated_by"-
  has_many :roles, :class_name => "Role", :through => :related_users-
  # Virtual attribute for the unencrypted password-
  attr_accessor :password-
  validates_presence_of :login-
  validates_length_of :login, :within => 3..100-
  validates_length_of :email, :within => 7..100-
  validates_uniqueness_of :login, :email, :case_sensitive => false-
  attr_accessible :login, :email, :password, :password_confirmation, :authorized_access-

```

nature: Sign in-
In order to get access to protected sections of the site-
As a user-
I want to be able to sign in-

Scenario: User is not signed up-

Given no user exists with an email of "email@person.com"-
When I go to the sign in page-
And I sign in as "email@person.com/password"-
Then I should see "Bad email/login or password"-
And I should be signed out-

Given no user exists with an email of "Cucumber do Test"-

insert(:id, User, :find_by_email(email))

;

When I fill in "remember me" with "Cucumber do Test"-
And I fill in "session_email" with "#(email)"-
And I fill in "session_password" with "#(password)"-
And I check "Remember me" if remember-
And I press "Sign In"-

;

Then I should be signed out-
do

assert ! controller.signed_in?

;

Given no user exists with an email of "Cucumber do Test"-

insert(:id, User, :find_by_email(email))

;

When I fill in "remember me" with "Cucumber do Test"-
And I fill in "session_email" with "#(email)"-
And I fill in "session_password" with "#(password)"-
And I check "Remember me" if remember-
And I press "Sign In"-

;

Then I should be signed out-
do

assert ! controller.signed_in?

;

Given no user exists with an email of "Cucumber do Test"-

insert(:id, User, :find_by_email(email))

;

When I fill in "remember me" with "Cucumber do Test"-
And I fill in "session_email" with "#(email)"-
And I fill in "session_password" with "#(password)"-
And I check "Remember me" if remember-
And I press "Sign In"-

;

Then I should be signed out-
do

assert ! controller.signed_in?

;

Given no user exists with an email of "Cucumber do Test"-

insert(:id, User, :find_by_email(email))

;

When I fill in "remember me" with "Cucumber do Test"-
And I fill in "session_email" with "#(email)"-
And I fill in "session_password" with "#(password)"-
And I check "Remember me" if remember-
And I press "Sign In"-

;

Then I should be signed out-
do

assert ! controller.signed_in?

;

Given no user exists with an email of "Cucumber do Test"-

insert(:id, User, :find_by_email(email))

;

When I fill in "remember me" with "Cucumber do Test"-
And I fill in "session_email" with "#(email)"-
And I fill in "session_password" with "#(password)"-
And I check "Remember me" if remember-
And I press "Sign In"-

;

Then I should be signed out-
do

assert ! controller.signed_in?

;

Given no user exists with an email of "Cucumber do Test"-

insert(:id, User, :find_by_email(email))

;

When I fill in "remember me" with "Cucumber do Test"-
And I fill in "session_email" with "#(email)"-
And I fill in "session_password" with "#(password)"-
And I check "Remember me" if remember-
And I press "Sign In"-

;

Then I should be signed out-
do

assert ! controller.signed_in?

;

Thursday, October 24, 13

RSpec

"Los componentes e interacciones que desearía tener"

```
describe User, "A new user" do
  it "should be created" do
    lambda{
      user = create_user
      user.should_not_be_new_record
    }.should_change(User,:count).by(1)
  end

  it "should require login" do
    lambda{
      u = create_user(:login => nil)
      u.should have_at_least(1).errors.on(:login)
    }.should_not_change(User,:count)
  end

  it "should require password" do
    lambda{
      u = create_user(:password => nil)
      u.should have_at_least(1).errors.on(:password)
    }.should_not_change(User,:count)
  end

  it "should require password confirmation" do
    lambda{
      u = create_user(:password_confirmation => nil)
      u.should have_at_least(1).errors.on(:password_confirmation)
    }.should_not_change(User,:count)
  end
```

```
class User < ActiveRecord::Base
  include Clearance::App::Models::User
  # Gravator # to support gravator associated to the email for the user
  has_and_belongs_to_many :roles
  has_one :vendor
  belongs_to :creator, :class_name => "User", :foreign_key => "created_by"
  belongs_to :updater, :class_name => "User", :foreign_key => "updated_by"
  has_many :roles, :class_name => "Role", :through => :related_users

  # Virtual attribute for the unencrypted password
  attr_accessor :password

  validates_presence_of :login
  validates_length_of :login, :within => 3..100
  validates_length_of :email, :within => 7..100
  validates_uniqueness_of :login, :email, :case_sensitive => false
  attr_accessible :login, :email, :password, :password_confirmation, :authorized_access
```

Cucumber



Manuel Vidaurre manuel.vidaurre@agiltec.com.mx @mvidaurre

© 2006 www.linkedin.com

PERO TODO ESTO SIRVE SIN UN BUEN DISEÑO?

Manuel Vidaurre  manuel.vidaurre@agiltec.com.mx  @mvidaurre



Manuel Vidaurre  manuel.vidaurre@agiltec.com.mx  @mvidaurre



Manuel Vidaurre  manuel.vidaurre@agiltec.com.mx  @mvidaurre



(c) Robert Koss. All rights reserved.

SRP: Single Responsibility Principle

OCP: Open Closed Principle

LSP: Liskov Substitution Principle

ISP: Interface Segregation Principle

DIP: Dependency Inversion Principle



Connascence

Cuando dos cosas nacen y crecen
Juntas.

Cuando hay conocimiento (*connascence*)
entre A y B Todo cambio en A requiere
un cambio en B para preservar lo correcto
y viceversa



SRP: *Single Responsibility Principle*

Principio de una Sola Responsabilidad

No debería nunca haber más de una razón para que una clase cambie

OCP: *Open Closed Principle*

Principio de abierto cerrado

Software de entidades (clases, módulos, funciones, etc.) Deben estar abiertos a la extensión pero cerrado por modificación



LSP: *Liskov Substitution Principle*

Principio de substitución de Liskov

Funciones que usen ... referencias a una clase base
deben ser capaz de utilizar objetos de clases
derivadas de la base sin saberlo

ISP: *Interface Segregation Principle*

Principio de segregación de interfaces

los clientes no se vean obligados a depender de
interfaces que no utilizan



DIP: *Dependency Inversion Principle*

Principio de inversión de dependencia

- A. Módulos de alto nivel no deben depender de módulos de bajo. Ambos deben depender de abstracciones
- B. Abstracciones no deberen depender de detalles. Detalles deben de depender abstracciones



Manuel Vidaurre  manuel.vidaurre@agiltec.com.mx  @mvidaurre

"Si

egador n

tes e
i que
ner"

Característica: adición

Para evitar hacer errores tontos

Como un matemático idiota

Quiero saber la suma de los números

Esquema del escenario: Sumar dos números

Dado que he introducido <entrada_1> en la calculadora

Y que he introducido <entrada_2> en la calculadora

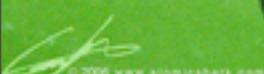
Cuando oprimo el <botón>

Entonces el resultado debe ser <resultado> en la pantalla

Ejemplos:

entrada_1	entrada_2	botón	resultado
20	30	add	50
2	5	add	7
0	40	add	40

Cucumber



Manuel Vidaurre manuel.vidaurre@agiltec.com.mx @mvidaurre



REFACTORING

RUBY EDITION

JAY FIELDS ■ SHANE HARVIE ■ MARTIN FOWLER
with KENT BECK



Extract Method

- This allows you to define a new method based on a selection of code statements.

Encapsulate Field

- Turns a public field into a private field encapsulated by a .NET property.

Extract Interface

- Defines a new interface type based on a set of existing type members.

Reorder Parameters

- Provides a way to reorder member arguments.

Remove Parameters

- As you would expect, this refactoring removes a given argument from the current list of parameters.

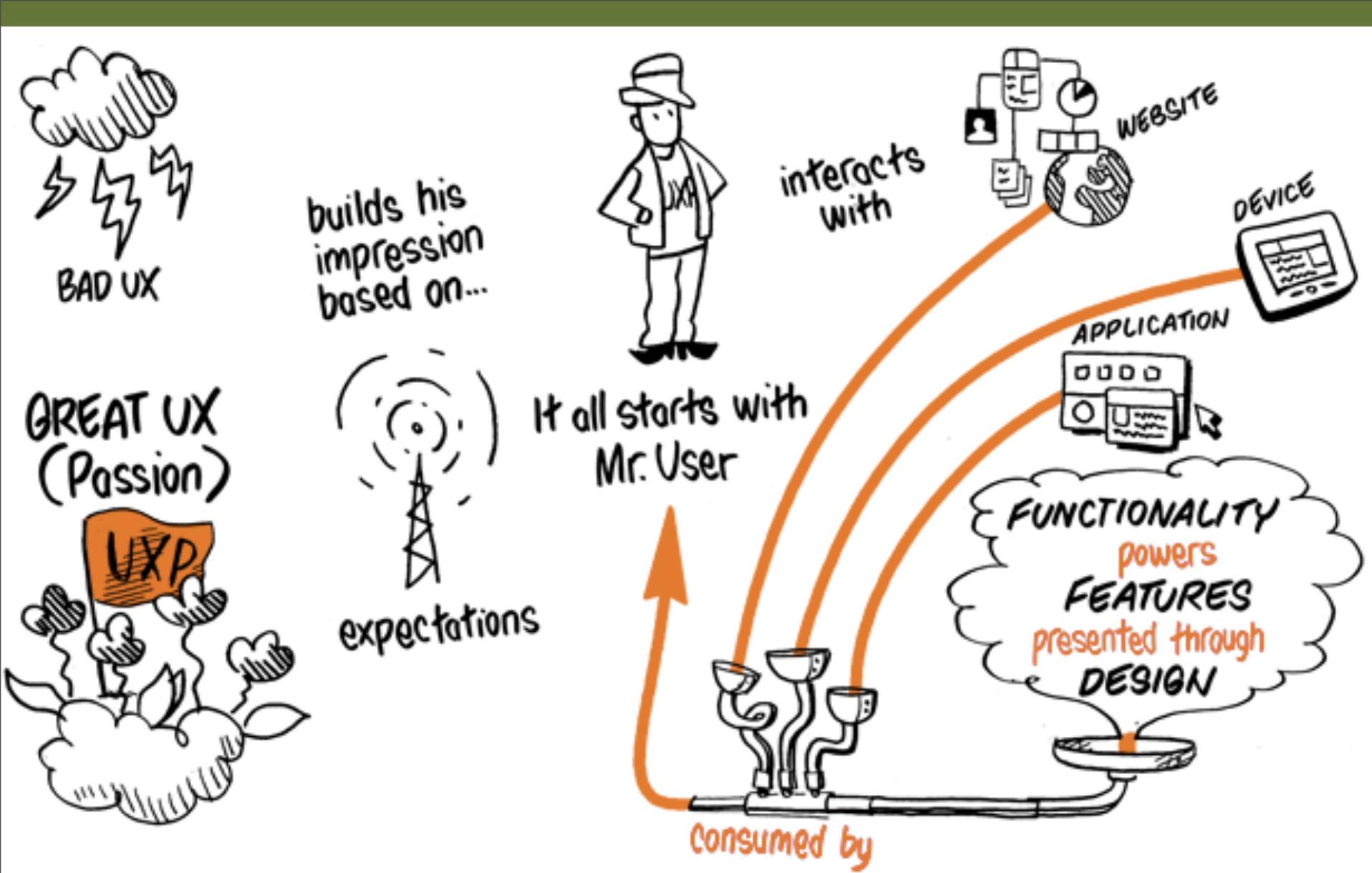
Rename

- This allows you to rename a code token (method name, field, local variable, and so on) throughout a project.

Promote Local Variable to Parameter

- Moves a local variable to the parameter set of the defining method.

Manuel Vidaurre manuel.vidaurre@agiltec.com.mx @mvidaurre



Manuel Vidaurre manuel.vidaurre@agiltec.com.mx @mvidaurre



www.projectcartoon.com

Como lo explico el cliente



www.projectcartoon.com

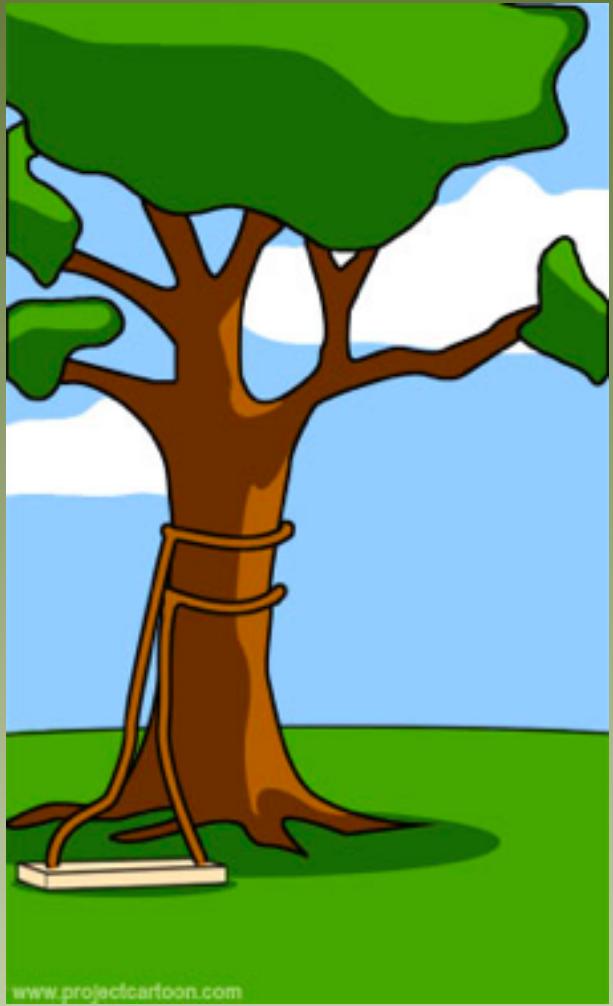
Como lo entendió el líder
de proyecto



www.projectcartoon.com

Como lo diseño el analista

Manuel Vidaurre  manuel.vidaurre@agiltec.com.mx  @mvidaurre



www.projectcartoon.com

Como lo escribió el
programador



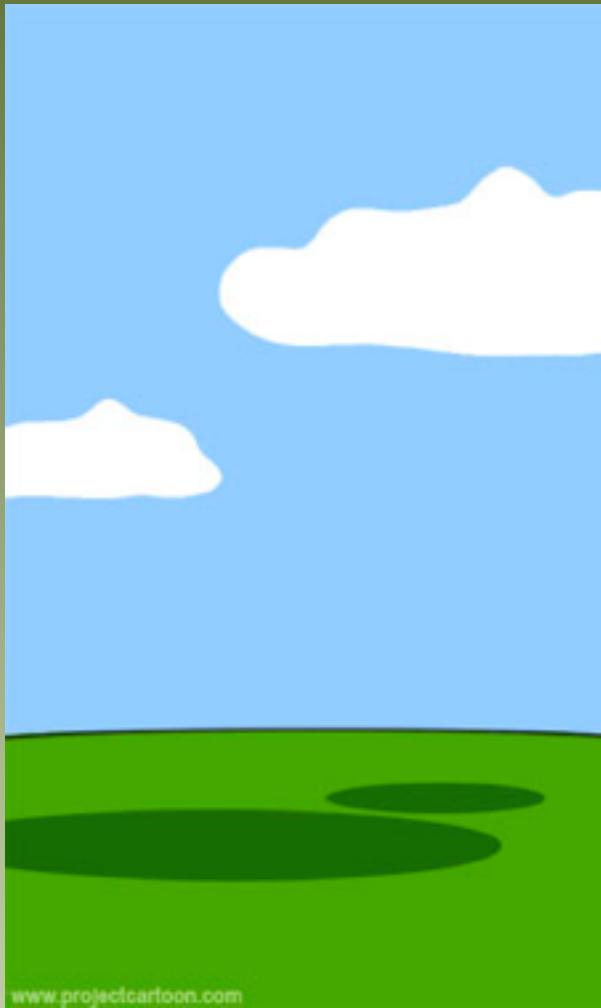
www.projectcartoon.com

Lo que recibieron los
probadores beta

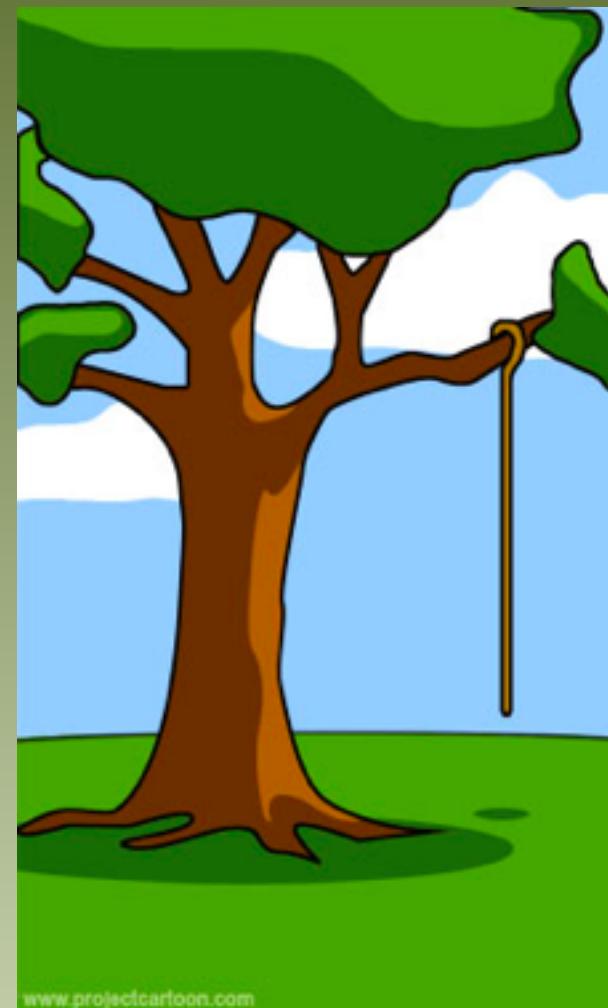


www.projectcartoon.com

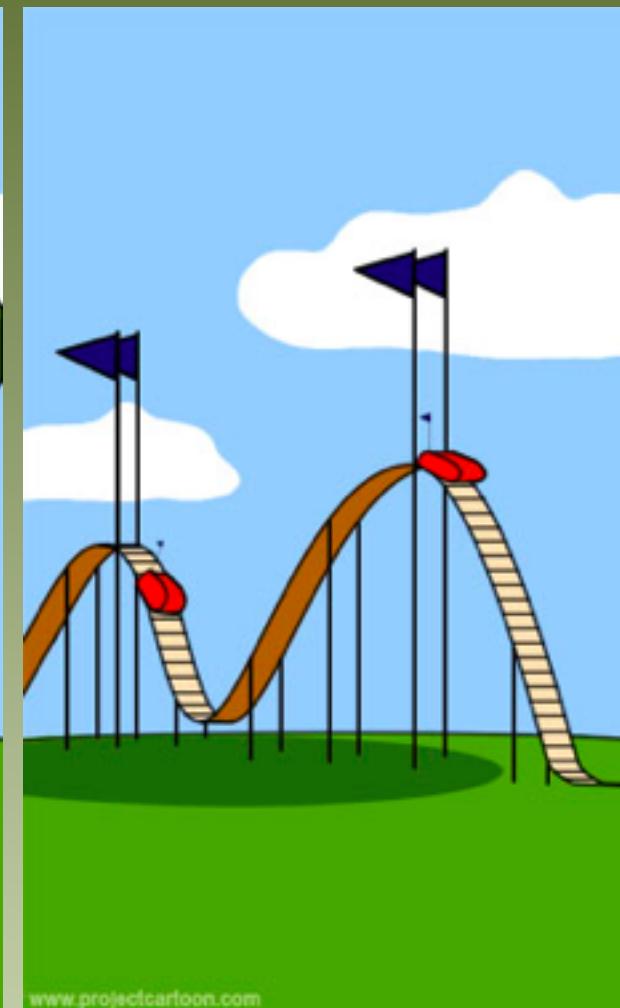
Como lo describió el
consultor de negocios



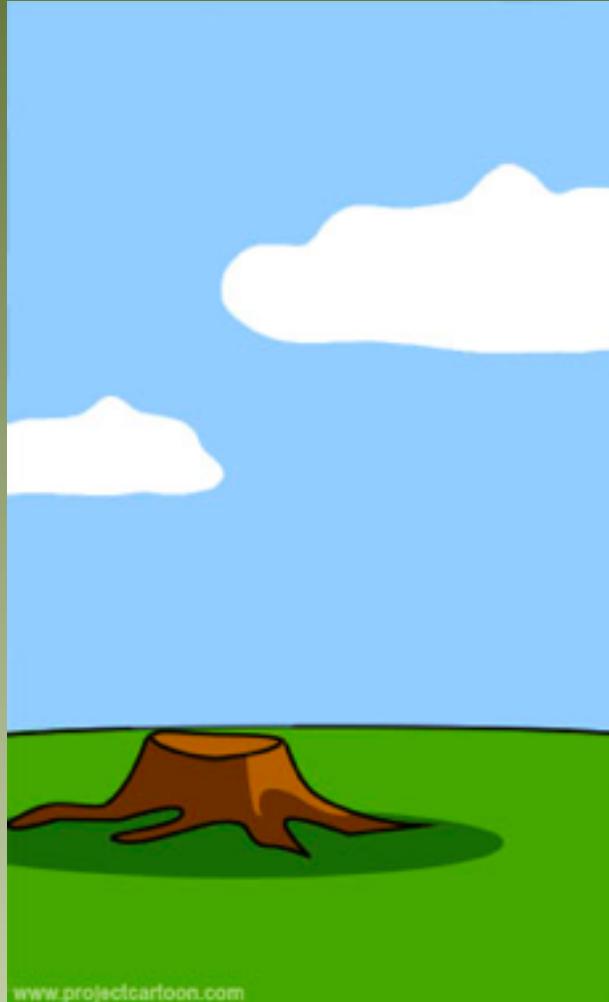
Como se documentó el
proyecto



Las operaciones
instaladas



Lo que se le cobró al
cliente



El soporte que se le dio



Como se comercializo



Lo que los clientes
realmente necesitaban

Nuestras Herramientas



Ruby

A Programmer's Best Friend

AJAX



HTML



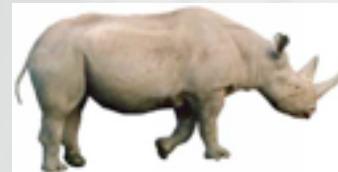
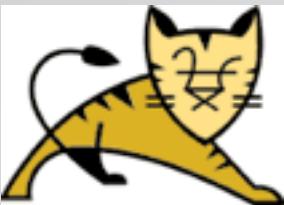
python™

*script.aculo.us
it's about the user interface baby!*

php



C#



CoffeeScript

github
SOCIAL CODING



Cucumber

behaviour driven development
with elegance and joy



Handlebars



Nuestros clientes



ActivaONE



CYTEC:



Servicios de Salud
de Yucatán

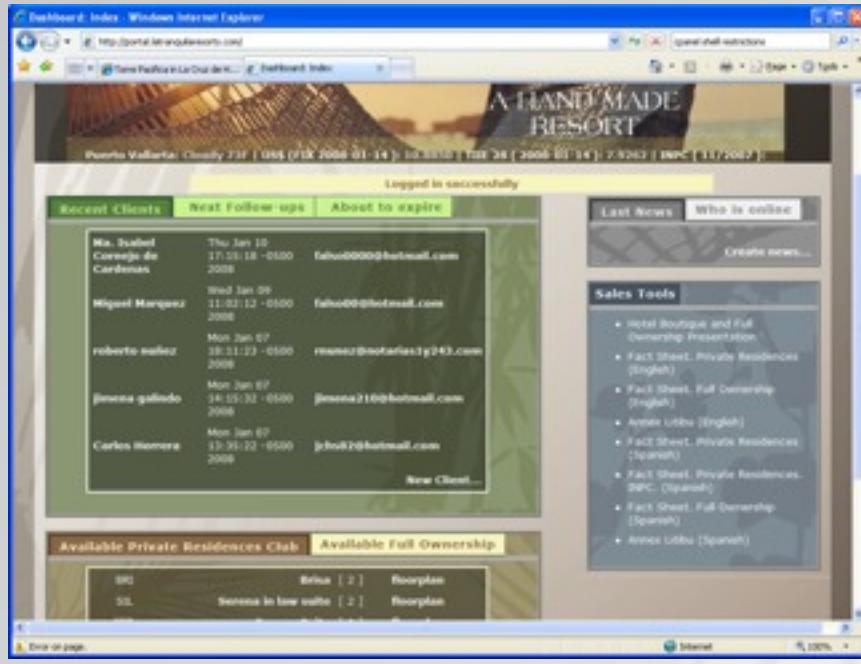


Centro Universitario de Ciencias Económico Administrativas



Web-based Systems

Portal La Tranquila

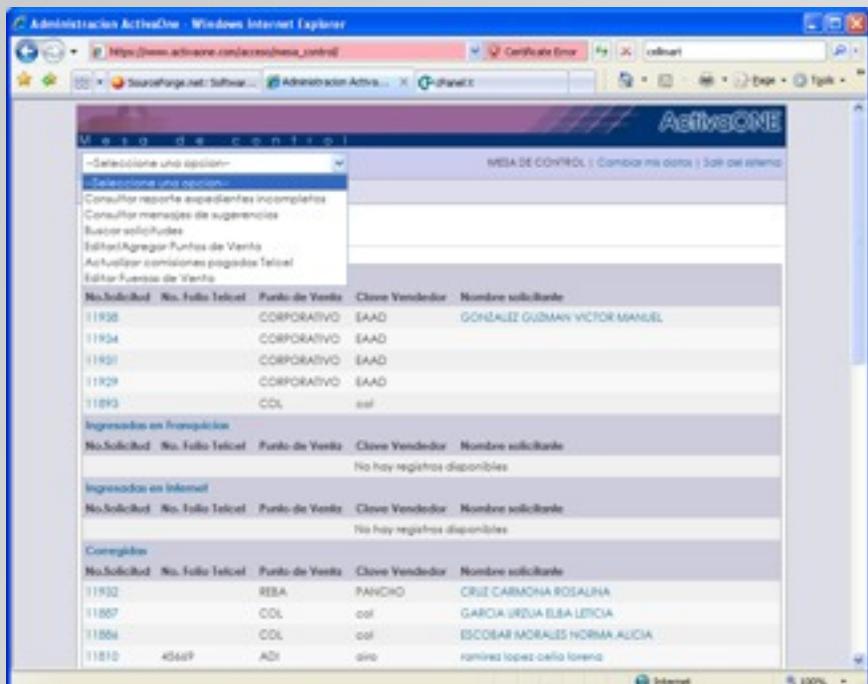


Portal Corporativo y Sistema de Información basado en Web para el control de los clientes, vendedores inmobiliarias, seguimientos, proceso de venta y pagos de propiedades para La Tranquila Resorts.

Desarrollado en Ruby on Rails 2.2 utilizando Web 2.0 (AJAX), XHTML, CSS, ECMAScript. RDBMS: MySQL

Web-based Systems

Sistema de mesa de control para CelularOne/ActivaOne



Sistema de Información basado en Web para el control del proceso de negocios de contratación, activación, altas, pago de comisiones para CelularOne distribuidor autorizado Telcel.
Desarrollado en PHP utilizando XHTML/CSS
RDBMS: MySQL

HTML

php



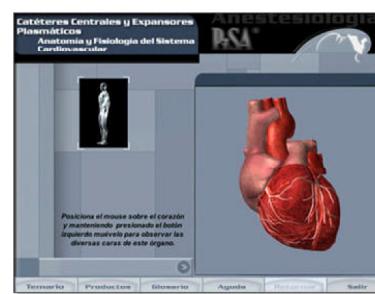
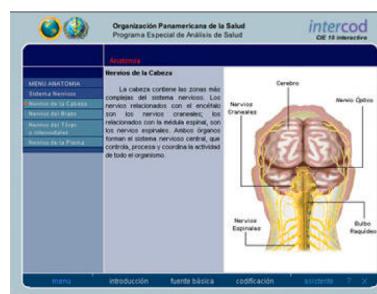
Web-based Systems

OPS/OMS Sistema Regional de Datos Básicos en Salud

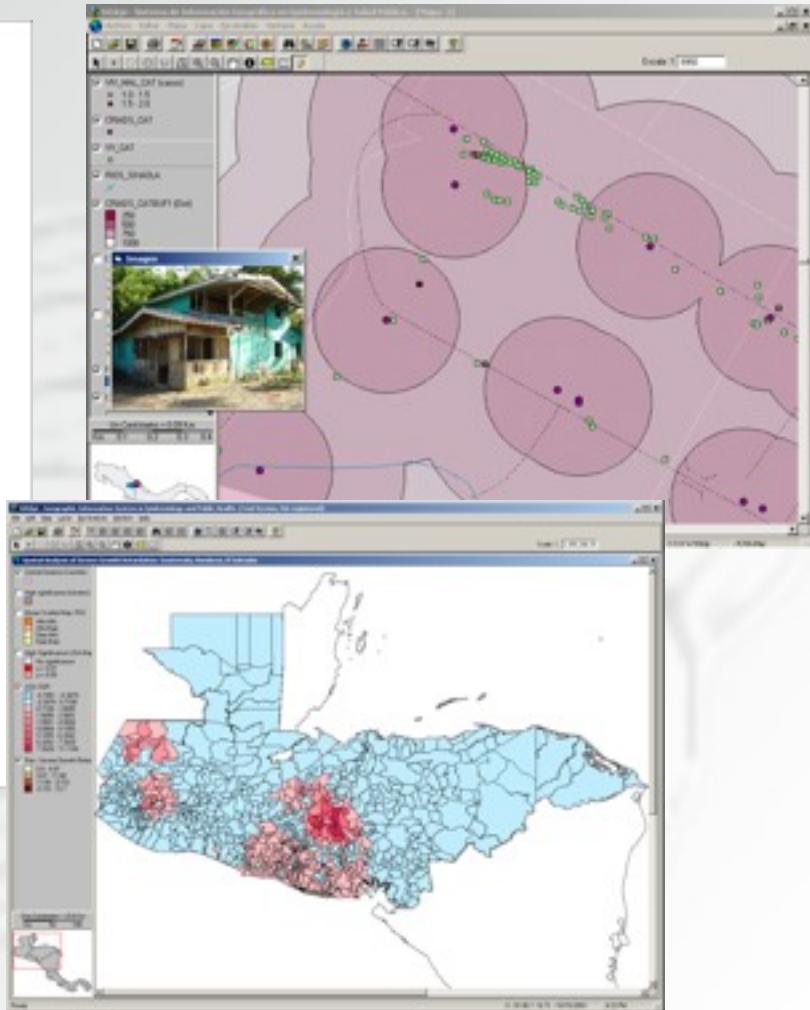
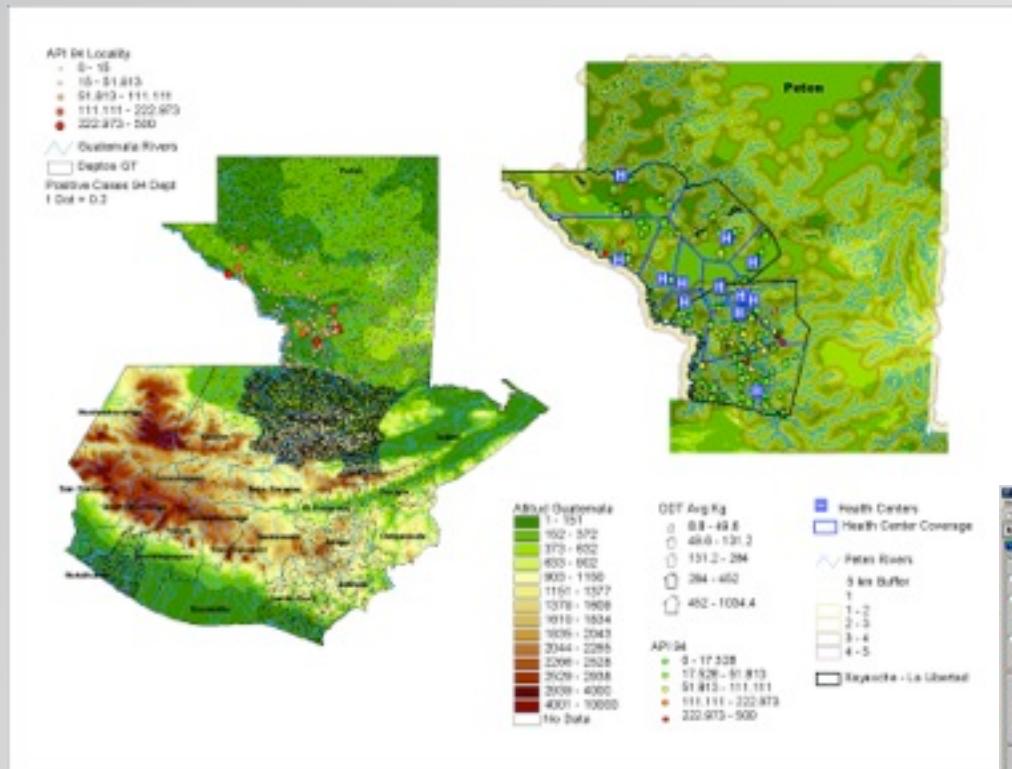
The screenshot shows a web-based application titled "Regional Core Health Data System - Table Generator". The URL is <http://www.paho.org/spanish/SMA/coredata/tabulator/newTabulator.htm>. The page header includes the Pan American Health Organization logo and links for "Hospitales seguros frente a los desastres" and "CAMPANERIA DE REDACCION DE DESATRES 2011 - 2015". The main content area is titled "Iniciativa Regional de Datos Básicos en Salud. Sistema Generador de Tablas". It features three dropdown menus: "Año(s)", "Indicador(es)", and "País(es)". The "Año(s)" menu lists years from 2001 to 2011. The "Indicador(es)" menu lists "Demográficos", "Socioeconómicos", "Mortalidad", and "Selección de Indicadores". The "País(es)" menu has a single entry. To the right of the menu is a search bar and a sidebar with various icons and links.

Sistema de Información basado en Web para indicadores básicos que reflejen la situación de salud de la región, se desarrollo una plataforma integral entre personas y organizaciones en los países, y organismos internacionales como el Banco Mundial, Banco Interamericano y Naciones Unidas Desarrollado en ASP/ECMAScript utilizando XHTML/CSS RDBMS: MS-SQL Server

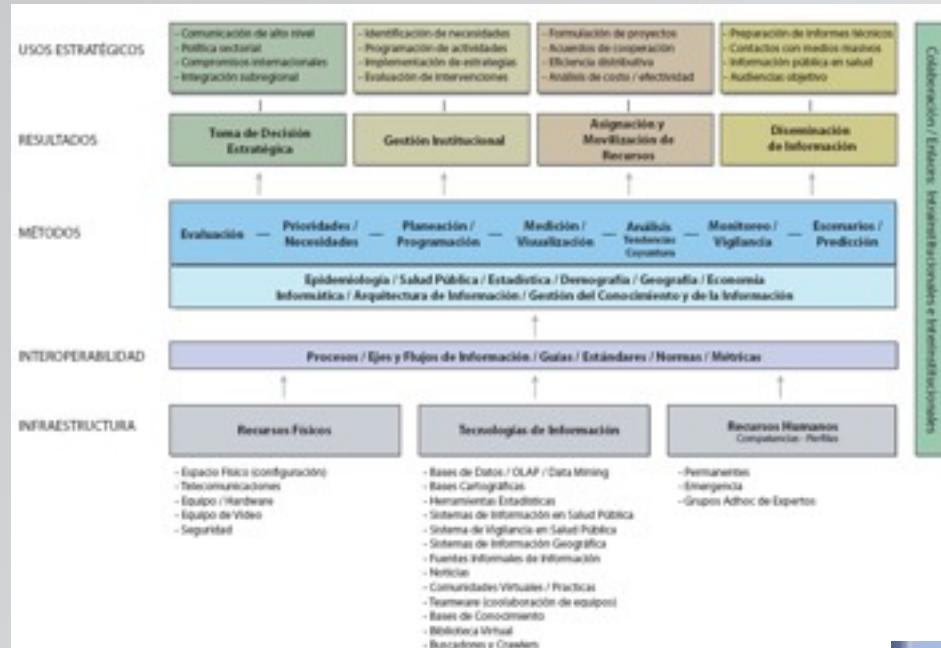
Arquitectura de Información, diseño instruccional o de interacciones



Sistemas de Información Geográfica



C4I2 - Control, Comando, Comunicaciones, Computo, Inteligencia e Interoperabilidad



Supercomputo y visualización 3D

CUDA ZONE

USA - United States | Search NVIDIA

DOWNLOAD CUDA WHAT IS CUDA CUDA U DEVELOPING WITH CUDA FORUMS NEWS AND EVENTS

LATEST CUDA NEWS New! CUDA 2.1 – Download the Complete Tool Suite Now

The CUDA Zone website features a grid of 15 project cards. Some examples include:

- Some Issues in Dense Linear (2x)
- Accelerating Lattice Boltzmann Fluid Flow Simulations Using Graphi (28x)
- National Instruments LabVIEW HIL application in control of extremely large (28x)
- Neurocuda - CUDA-accelerated neurodynamics (2x)
- MAGMA (Matrix Algebra on GPU and Multicore Architectures) (2x)
- A Note on Auto-tuning GEMM for GPUs
- GPU Particle Tracking and Multi-Fluid Simulations with Gmres Enhanced (60x)
- Flame Fractals (50x)
- GPU-Quicksort (50x)
- CG, GMRES, FGMRES, BiCGStab (50x)
- Fast Iterative Linear System Solvers (50x)
- ConAVC Professional
- Multi-GPU Incompressible Navier-Stokes Solver (100x)
- Multigrid on GPU: Tackling Power Grid Analysis on Parallel SIMD Platforms
- Enhancing the Performance of Dense Linear Algebra Solvers on GPUs (2x)
- LAPACK → MAGMA (Matrix Algebra on GPU and Multicore Architectures)



Mobile Apps and API Servers





Por todo esto
AgilTec es tu mejor
aliado para que
alcances tus metas
y le des mayor
valor a tu
organización
mediante
Tecnología de
Información