

Ecosistemas de desarrollo software

Líneas de automatización



Asignatura: **Desarrollo de Software Basado en Modelos y Aspectos**
Máster: **Ingeniería y Tecnologías del Software**
Organiza: **Departamento de Lenguajes y Sistemas Informáticos**
Lugar: **Escuela Técnica Superior de Ingeniería Informática**
Fecha: **14/02/2012**

Manuel Recena Soto

comencé a programar GW-Basic
con esta maravilla



este es mi proyecto personal y
profesional



y dedico gran parte de mi tiempo a

- gestionar los números de la empresa
- prototipar e implementar UI
- relación con los clientes
- limpiar la oficina
- mantener infraestructura
- evolucionar nuestro laboratorio
- desarrollar
- definición de producto
- participar en charlas, grupos de trabajo
- consultoría tecnológica
- auditoria de proyectos software

 <http://www.linkedin.com/in/recena>

 @recena

 recena@klicap.es

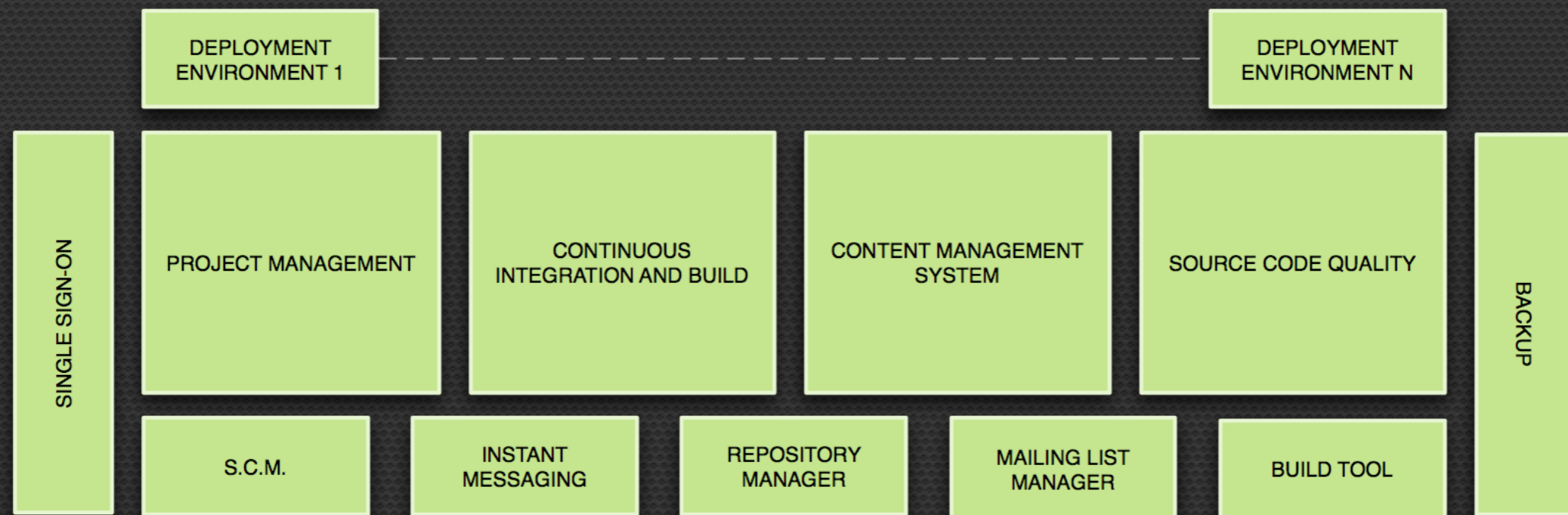
 recena@gmail.com

DE QUÉ HABLAREMOS

- Build
- Automatic Documentation Generation
- Testing
- Continuous Inspection
- Continuous Deployment

ANTES DE COMENZAR...

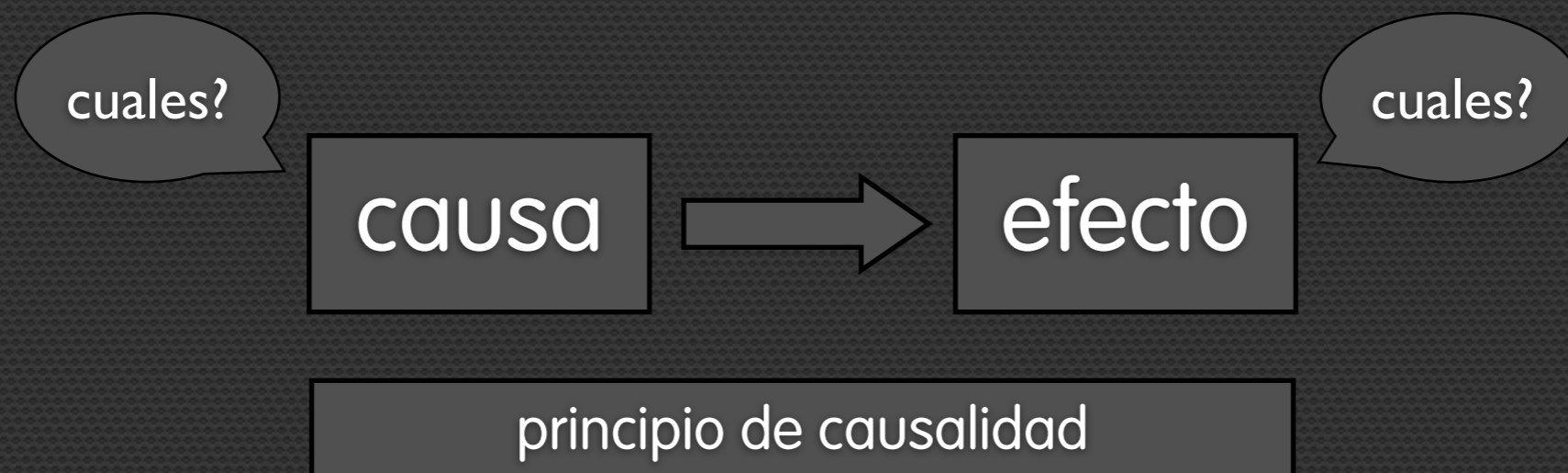
Un ecosistema de desarrollo software es un espacio de trabajo en el que **conviven** una serie de herramientas que acompañadas de unas buenas prácticas permiten a un equipo de desarrollo **modelar una metodología** de trabajo



modelo conceptual

ANTES DE COMENZAR...

La integración continua es una **práctica** en el desarrollo de software que consiste en poner en común **todos los cambios** que afecten al resultado final de nuestro proyecto de una forma **frecuente** con el objetivo de ver la **evolución de sus efectos**



Build

- Compilación
- Empaquetado
- Distribución (generación distribuibiles ¹)

¹ <http://www.manuelrecena.com/blog/archives/125>

Build

- Detectar qué cambios en el repositorio de código hacen que algo se rompa
- Distribuir rápidamente artefactos que se reutilizan
- Disponer de múltiples configuraciones de un mismo artefacto

Build

- Herramientas
 - Maven, Rake, Ant, Ivy, Buildr, Gradle, Phing, SCons, ...
- Requisitos para su ejecución

Automatic Doc Generation

- La documentación como un artefacto más que debemos mantener
- Enfoques:
 - A partir del código fuente
 - A partir de formatos ágiles

Automatic Doc Generation

- Una entrada, múltiples salidas
- Imagen corporativa
- Edición colaborativa

Automatic Doc Generation

- Herramientas

- Maven sites, doxygen, javadoc, JSDoc, phpDocumentator, docbook, Sphinx, Markdown, Textile, RDoc, Org Mode, Creole, ReStructured Text (ReST), ASCIIDoc, POD, Roff, ...

Testing

- Unitarias / Funcionales
- Integración / Sistemas
- Rendimiento

Testing

- Herramientas
 - JUnit, Selenium, The Grinder, DbUnit, BrowserStack, TestNG, ...
- Requisitos para su ejecución

Continuous Inspection

- Analizar desde un punto de vista estático
- Verificar y comprobar pautas y reglas de codificación
- Detectar posibles fallos de programación

Continuous Inspection

- Herramientas

 - Sonar

- Requisitos para su ejecución

Continuous Deployment

- Verificar que el despliegue no se rompa
- Despliegues en múltiples entornos
- Habilitar la instalación en entornos de producción

Continuous Deployment

- Despliegue en infraestructura propia o externa (Jelastic, Google App Engine, Heroku, ...)
- Virtualización como pieza clave para nuestros entornos de despliegue en local

BENEFICIOS

- Transparencia
- Agilidad
- Productividad
- Control / Trazabilidad

EL CAMINO CONTINUA



Vagrant



JUJU



Chef



GRACIAS