

MADS

S6: Extreme Programming (Bloque 2 - XP)

1996

Copyrighted Material
Kent Beck's
Guide to Better Smalltalk
A Sorted Collection

Over the last ten years Kent Beck has written dozens of technical papers for the Smalltalk community, earning himself a reputation as both a gifted writer and thinker. *Kent Beck's Guide to Better Smalltalk*, is a collection of his best work from *Object Magazine*, *The Smalltalk Report*, *Dr Dobbs Journal*, and more. Each article has a new introduction that takes a retrospective view of the writing.

Topics include:

- idioms and environments,
- methods and metamodels,
- architecture and pattern languages,
- objects, classes, inheritance,
- and all things Smalltalk!

Nowhere else can one obtain such a complete collection of Beck's writing. While demonstrating the elegance of Smalltalk and how some of its most powerful features can be exploited profitably, this collection also illuminates breakthrough concepts in object-oriented development.

This book is for Smalltalk programmers and anyone working in object-oriented software development.

About the Author



Kent Beck has been programming in Smalltalk for fifteen years. In that time, he has become one of the most important voices in the Smalltalk programming community. Not only has his name become synonymous with good Smalltalk programming, but he has also been instrumental in the popularization of such important innovations as CRC cards and software patterns.



Cover art courtesy of FPG International

Copyrighted Material

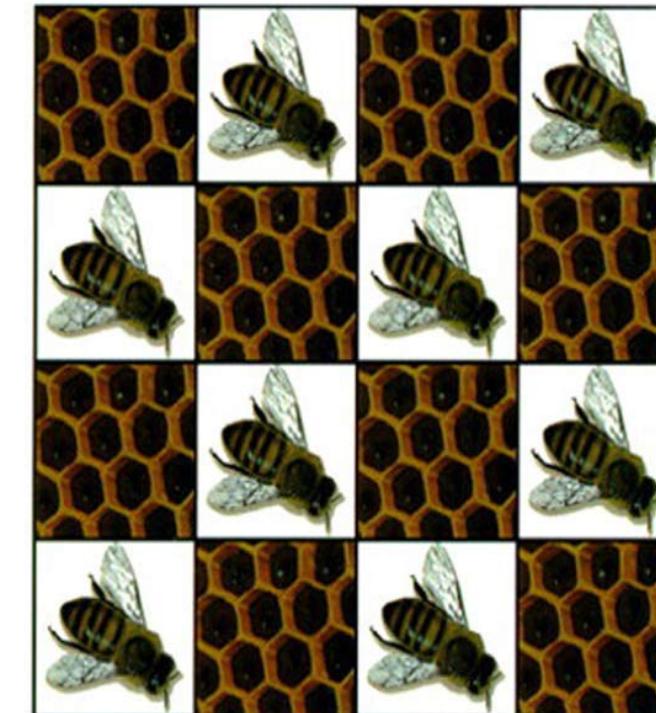
CAMBRIDGE
UNIVERSITY PRESS
www.cambridge.org

ISBN 978-0-521-64437-2



9 780521 644372 >

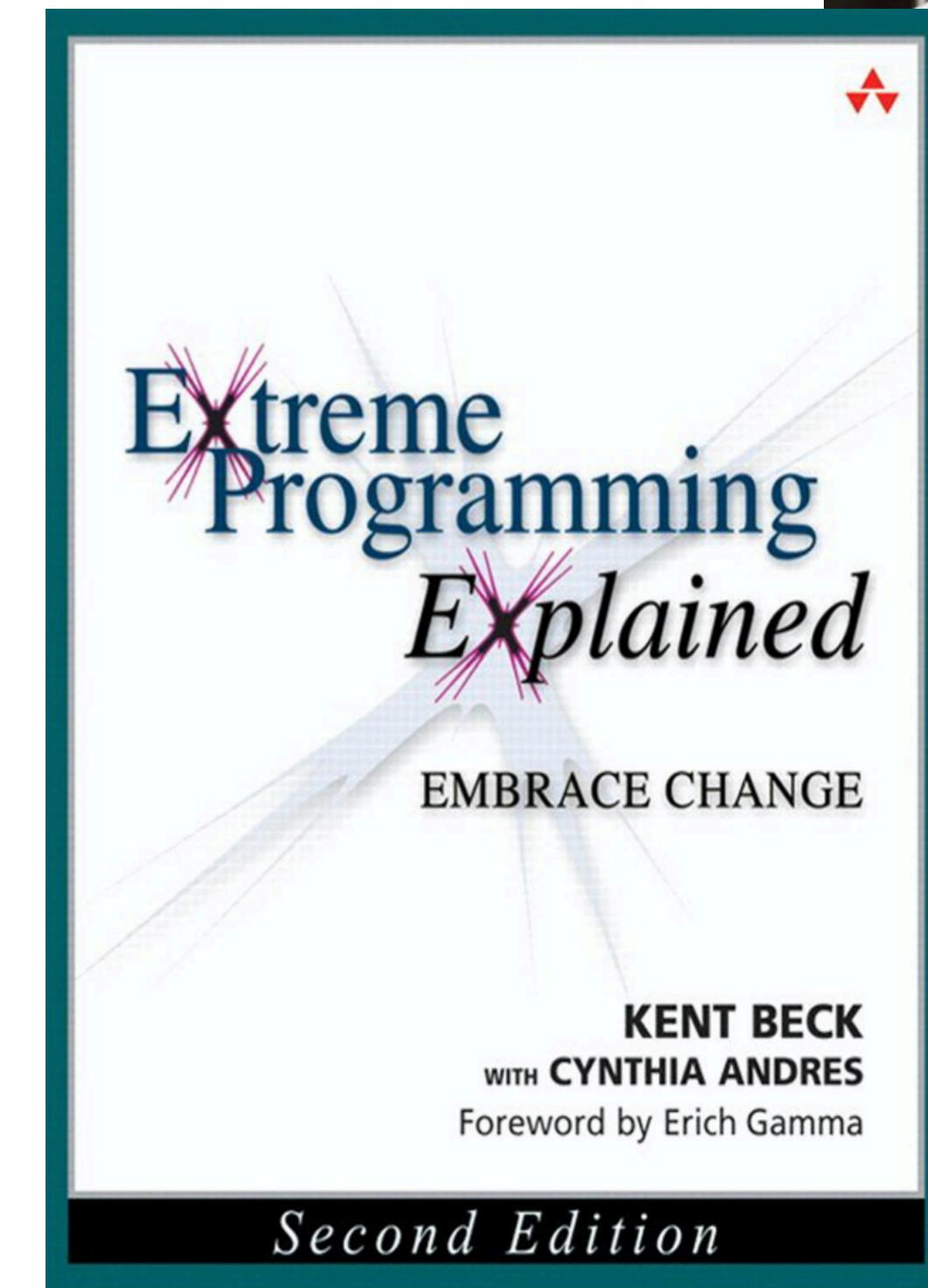
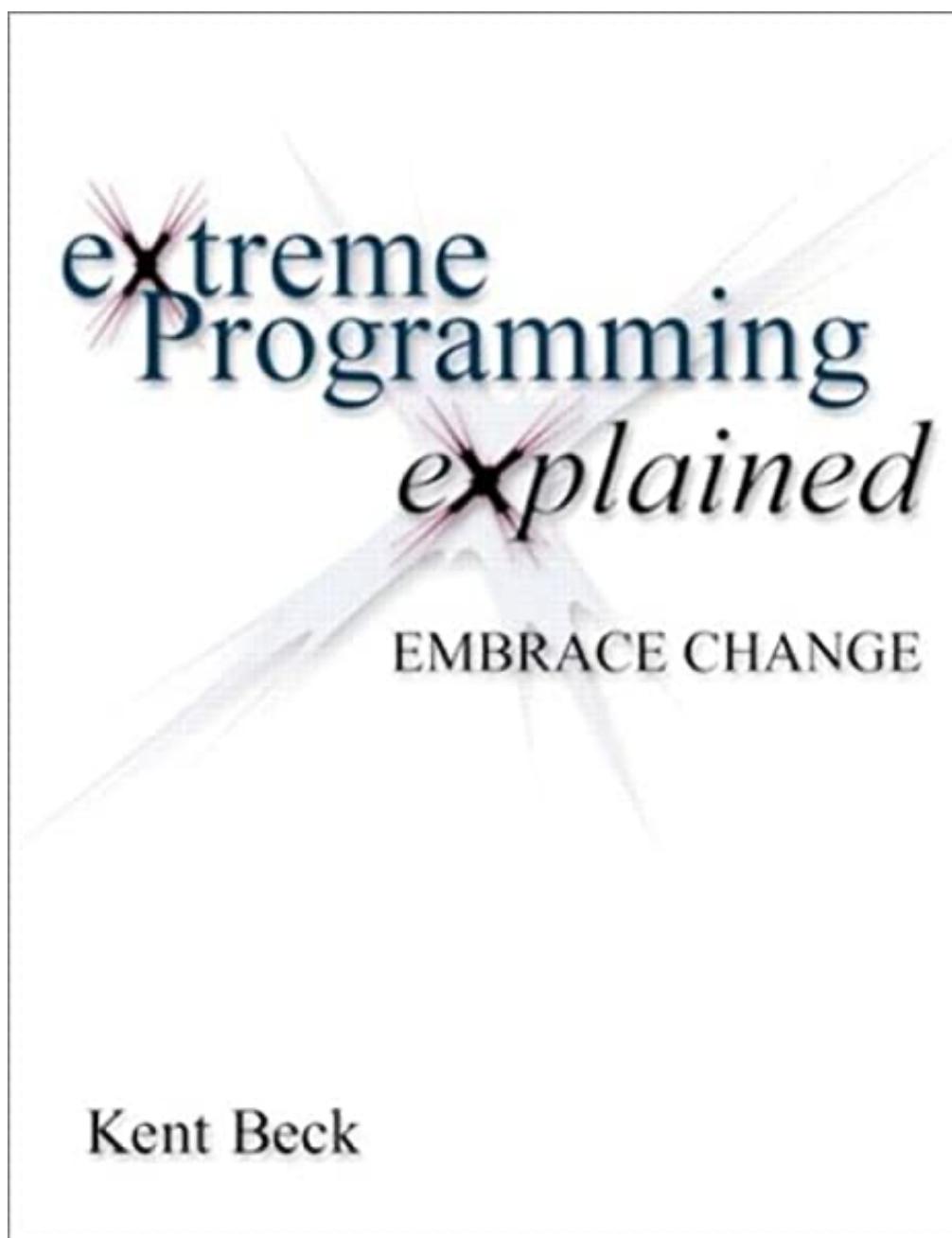
SMALLTALK BEST PRACTICE PATTERNS



KENT BECK

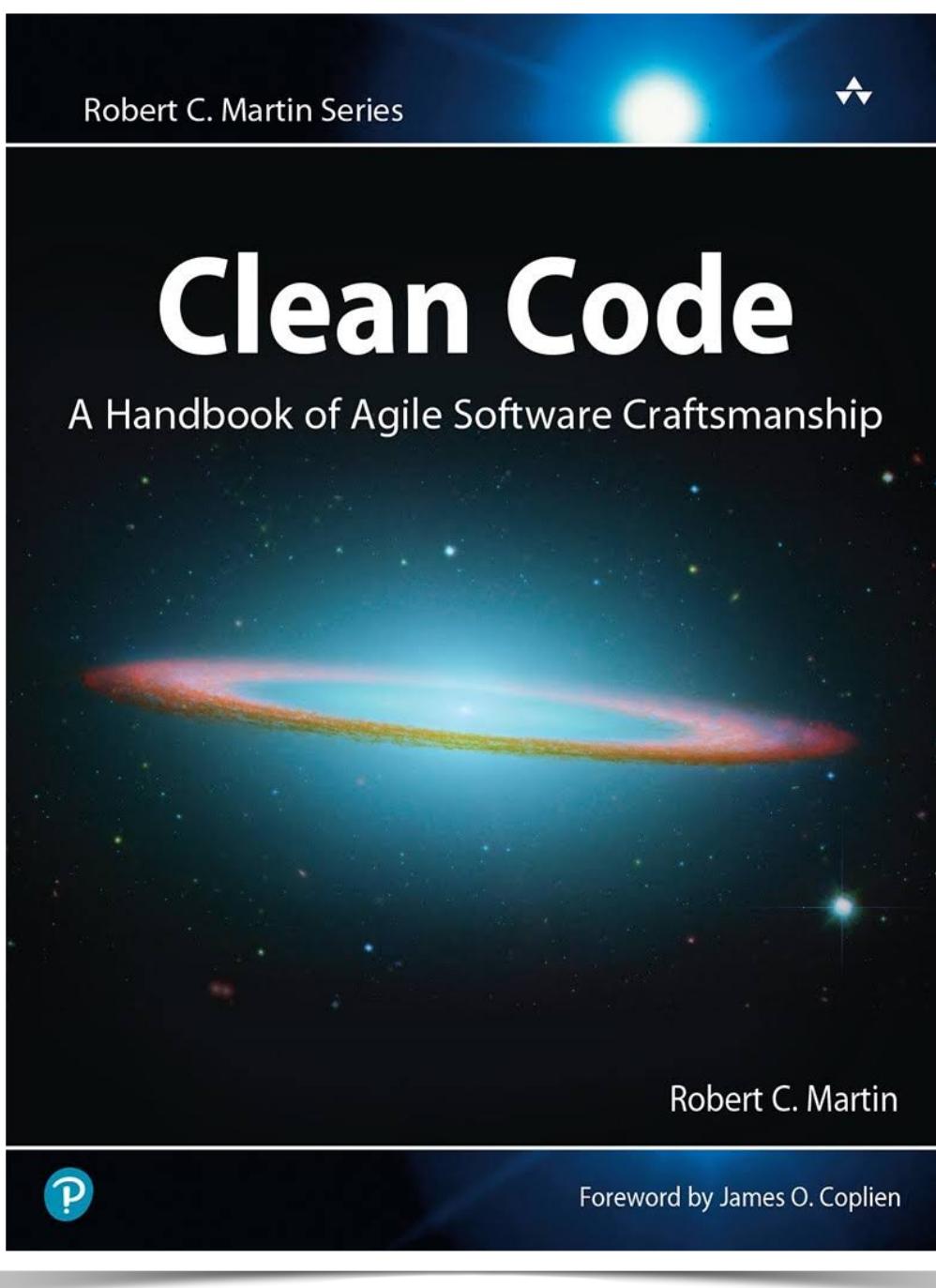
C3 was the short name of the Chrysler Comprehensive Compensation project, a payroll project at Chrysler which has since become famous as the 'birth project' of Extreme Programming.

1999-2004



Kent Beck

2010-20



A screenshot of a social media event page for 'GPUL <Labs/>'. The page shows basic information about the group, including its location (A Coruña, España), date of creation (Fundado el 4 de feb de 2016), and some statistics: 329 members, 1 review, 3 upcoming events, and 30 events. There are also links to 'Quiénes somos?' and a 'Meetups' section.

[Labs] Introducción al Software Craftsmanship

[Díselo a un amigo](#) [Compartir](#)

jueves 22 de junio de 2017
19:00

Vivero de empresas Accede Papagaio
Rúa Hospital, 19, 15002, A Coruña ([mapa](#))

Borja Fernández - Barkibu

Introducción al movimiento Software Craftsmanship, sus principios y prácticas (15-20 min).

¿En qué consiste? ¿Por qué es importante para mi como desarrollador de software? ¿Quién lo sigue? Intentaré responder estas y otras preguntas.

Y para no aburrirnos haremos una kata de desarrollo: "Bowling

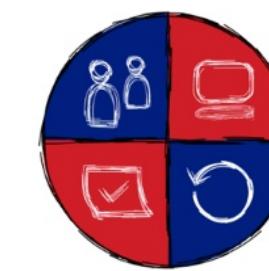
¿Quieres ir?

Únete y confirma tu asistencia

Asistirán 10

 Asociación G.
Organizador,
Anfitrión del evento

La Asociación GPUL fue fundada en el año 1998 cuya sede es la Facultad de Informática de la...



Introducción práctica a *Test-Driven Development*

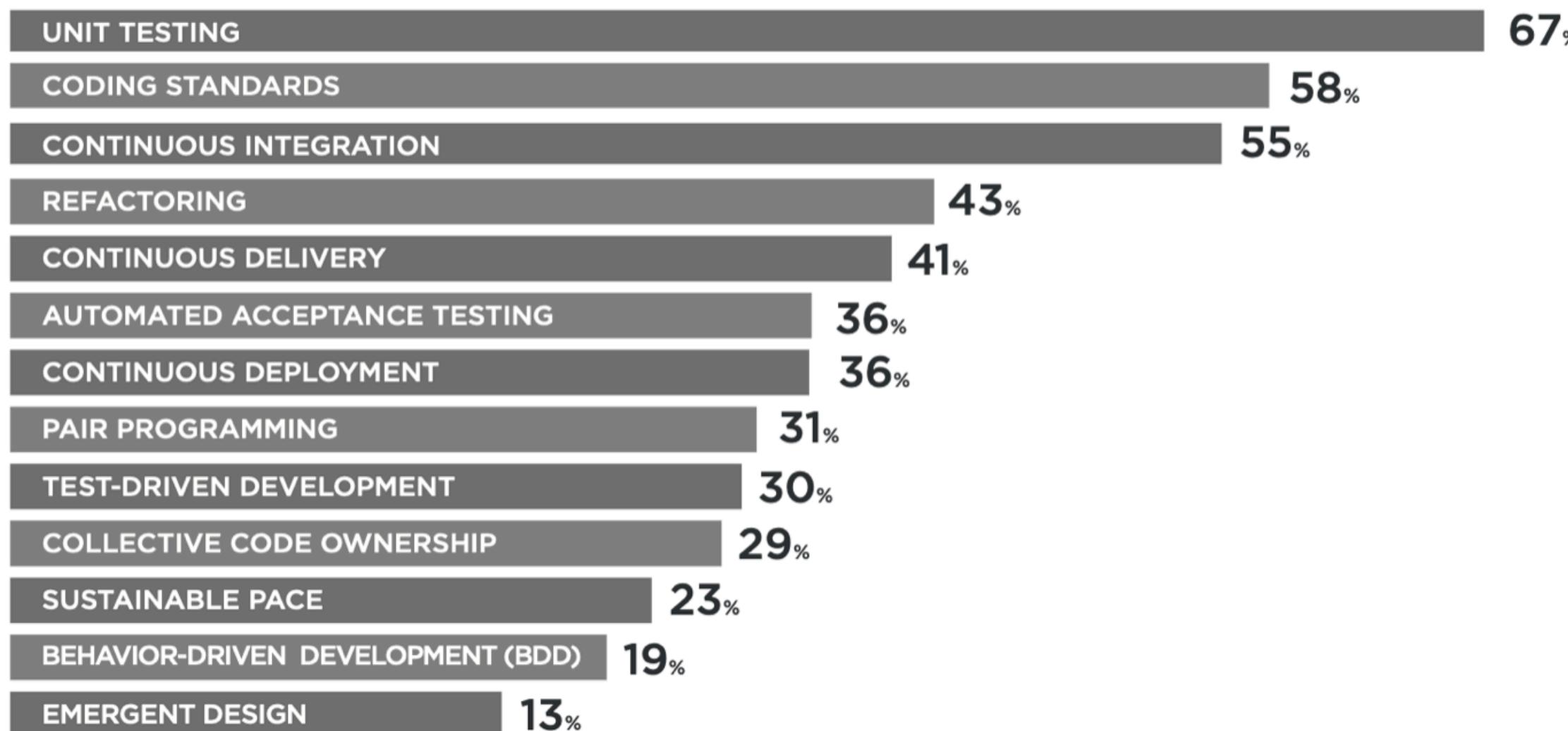
Software Craftsmanship Alicante

2020

14th annual
STATE OF
AGILE™
REPORT

ENGINEERING PRACTICES EMPLOYED

The overall rank order of engineering practices employed remained almost the same this year over last. Automated acceptance testing increased 3% while pair programming, test-driven development, and behavior-driven development each fell 3%.



*Respondents were able to make multiple selections

Intro rápida a XP - prácticas

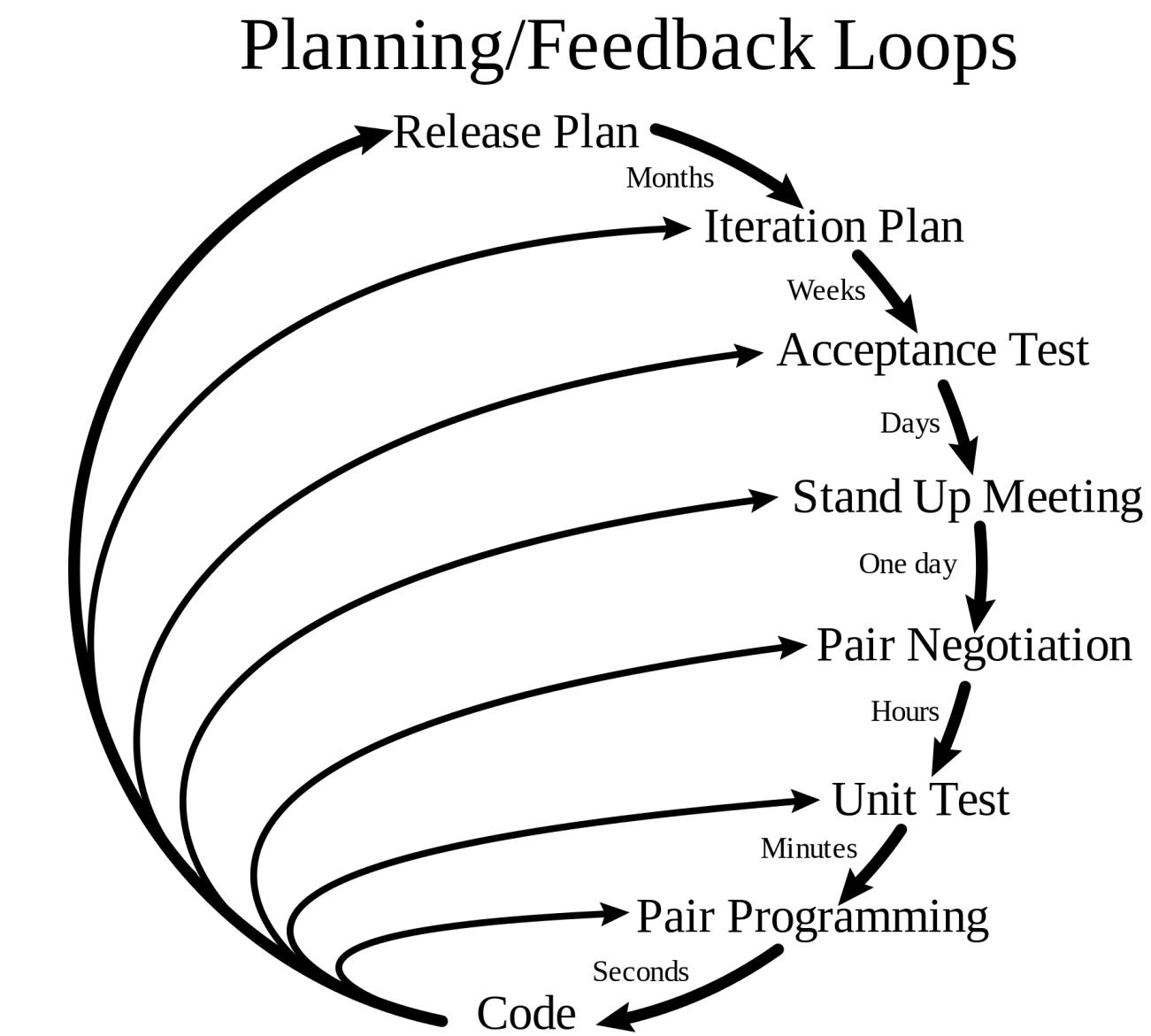
- Un desarrollador trabajando con XP participará en un equipo que se siente unido, que trabaja en un espacio común con paredes llenas de esquemas, fichas, y elementos informativos y que desarrolla su faena a pleno rendimiento, trabajando de forma sostenida y sin presión ni agobios.
- El equipo diseña el software escribiendo en tarjetas (denominadas *historias*) las nuevas funcionalidades a implementar usando un lenguaje de negocio común. El equipo construye la aplicación usando un enfoque iterativo y evolutivo, basado en un diseño incremental.

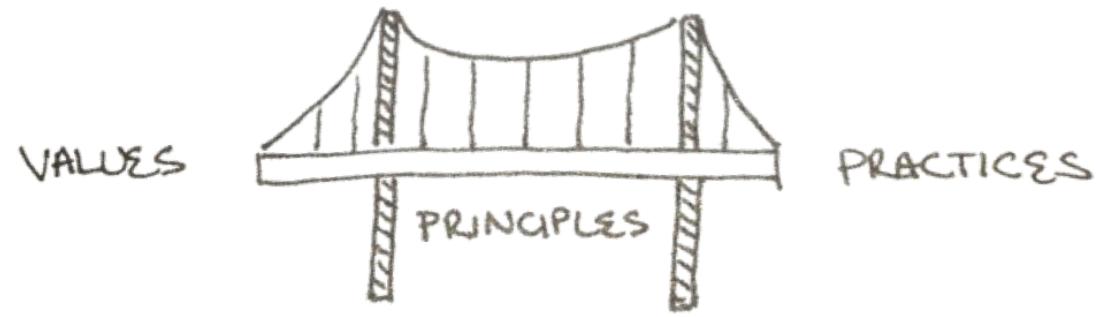
Intro rápida a XP - prácticas

- Semanalmente los clientes definen las historias y se añaden a la aplicación, definiendo primero los tests que deben cumplir y trabajando en parejas para escribir el código que hace pasar esos tests. Continuamente se compila la aplicación, se integra y se pasan todos los tests, en un proceso que debería ser muy rápido (menos de 10 minutos).
- Cada tres meses se reflexiona sobre la dirección global del proyecto, estableciendo nuevas prioridades y analizando posibles mejoras en el proceso.

Valores de XP

- Comunicación
- Simplicidad
- Retroalimentación
- Valentía
- Respeto

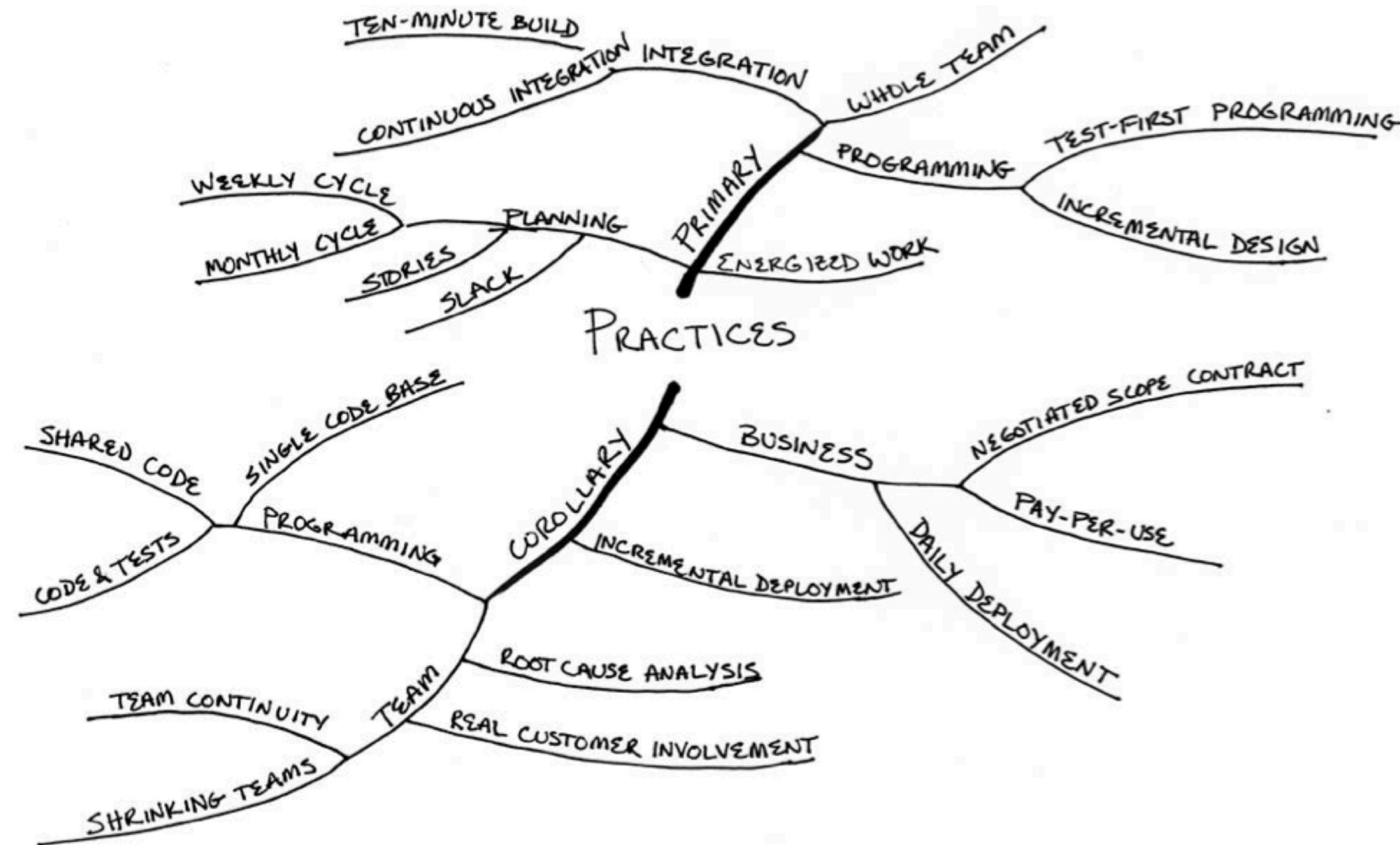




Principios de XP

- Humanidad
- Economía
- Beneficio mutuo
- Auto-similitud
- Mejora
- Diversidad
- Reflexión
- Flujo
- Oportunidad
- Redundancia
- Fallo
- Calidad
- Pequeños pasos
- Responsabilidad aceptada

Prácticas de XP



Prácticas de XP

- Prácticas de trabajo en equipo:
 - Sit together
 - Whole team
 - Energized work
 - Slack
- Prácticas de diseño:
 - Informative workspace
 - Stories
 - Incremental design
- Prácticas de programación:
 - Pair programming
 - Test-first programming
- Prácticas de integración:
 - Ten minutes build
 - Continuous integration
- Prácticas de planificación:
 - Weekly cycle
 - Quaterly cycle

Prácticas de XP en el desarrollo de Eclipse

- Erich Gamma fue el líder de desarrollo de Eclipse
- En 2004 comenta en el prefacio del libro de Beck que las prácticas de XP fueron fundamentales para el desarrollo del proyecto:
 - Pruebas desde el principio, frecuentes y automatizadas
 - Diseño incremental
 - Despliegue diario
 - Implicación del cliente
 - Integración continua
 - Ciclos de desarrollo cortos
 - Planificación incremental



Prácticas equipo Sit Together



Prácticas equipo Whole Team

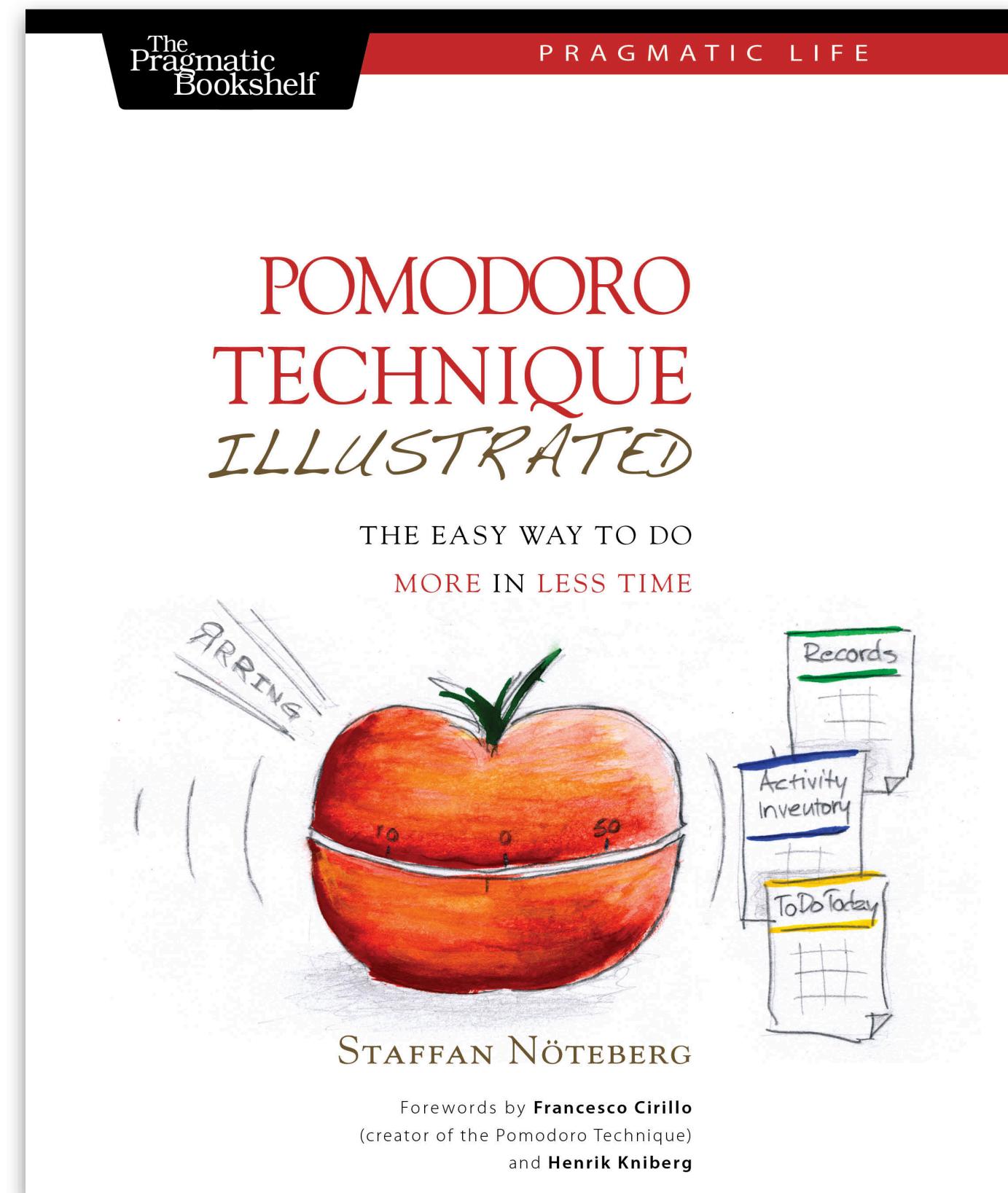
SKILLS NEEDED TO IMPLEMENT TOP X BACKLOG ITEMS						
	TEST	DB	WEB	JAVA	DOMAIN	CM
LISA	•	•	•	★	I'm good at Java!	•
JOE	•	★		•	•	
FRED	•			★	•	•
JENNY	•		★	•		
DAVID	★			•	★	•
ERIK			★	★	•	★

I can't test, but I'm not so good at it.

I don't know CM at all. But I'm willing to learn!

I won't even go near a database!

Prácticas equipo Energized Work

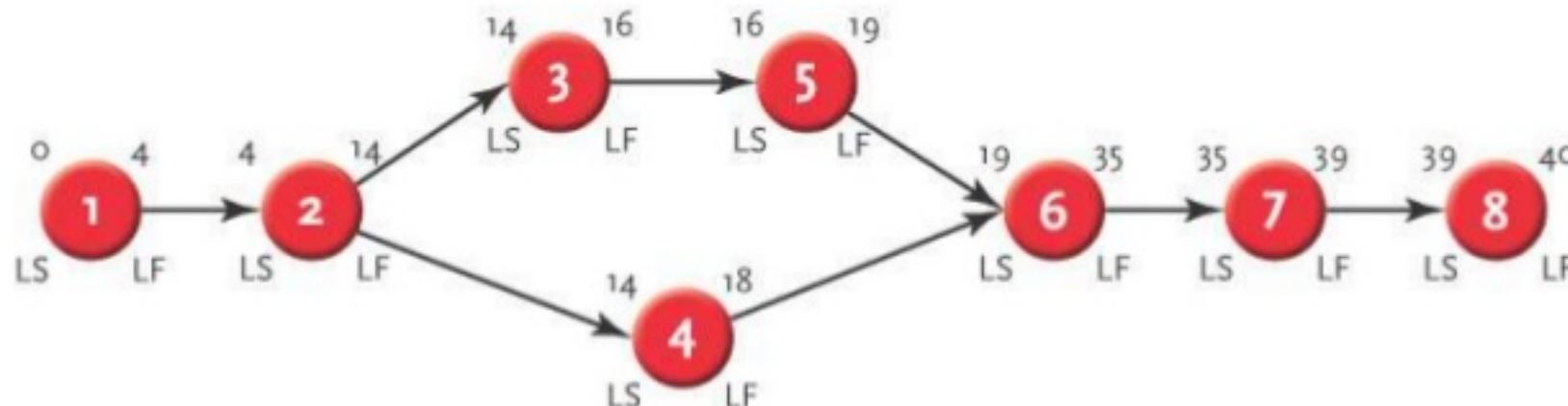


Prácticas equipo

Slack

Determining Slack

- Slack - The amount of time an activity on a non-critical path can be delayed without affecting the duration of the project (i.e., without putting it on the critical path)
 - Uses four calculated values
 - Early start - Earliest an activity can start (based on prerequisites)
 - Early finish - Earliest it can finish (based on prerequisites & duration)
 - Late start - Latest an activity can start and not delay the project
 - Late finish - Latest an activity can finish and not delay the project

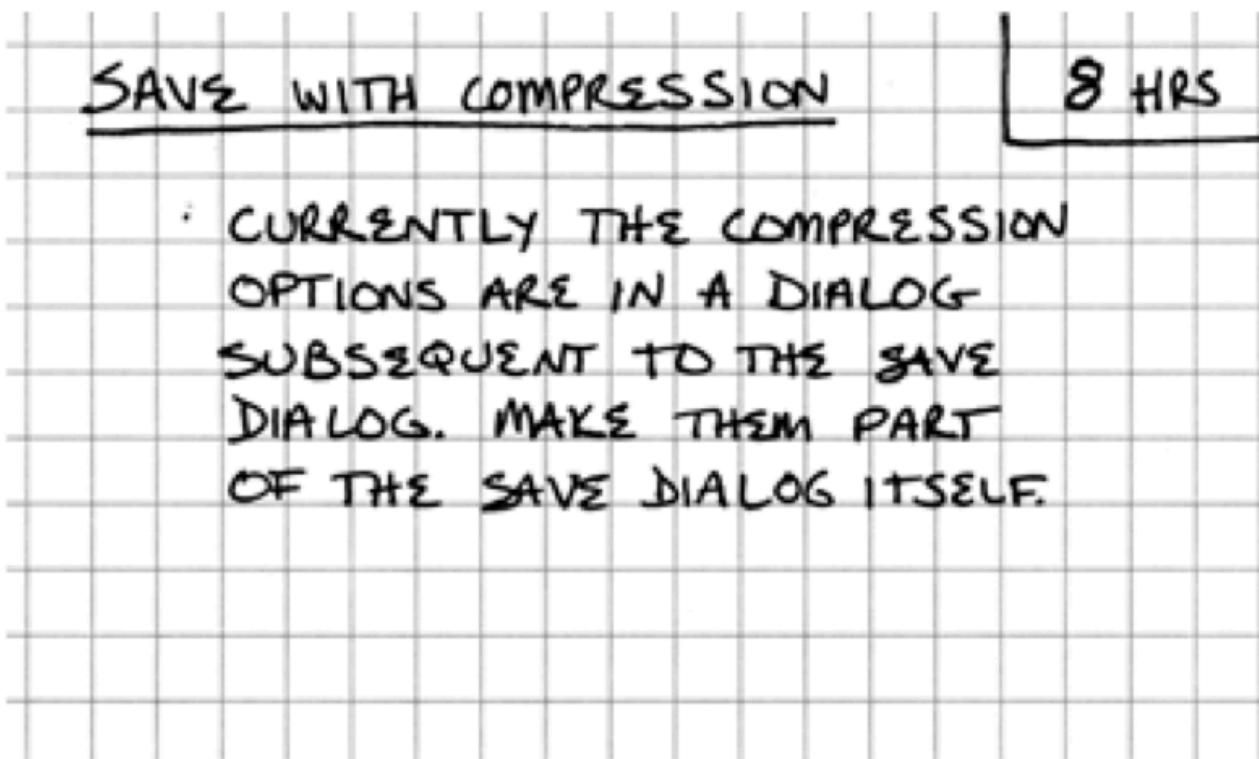


Prácticas diseño

Informative workspace



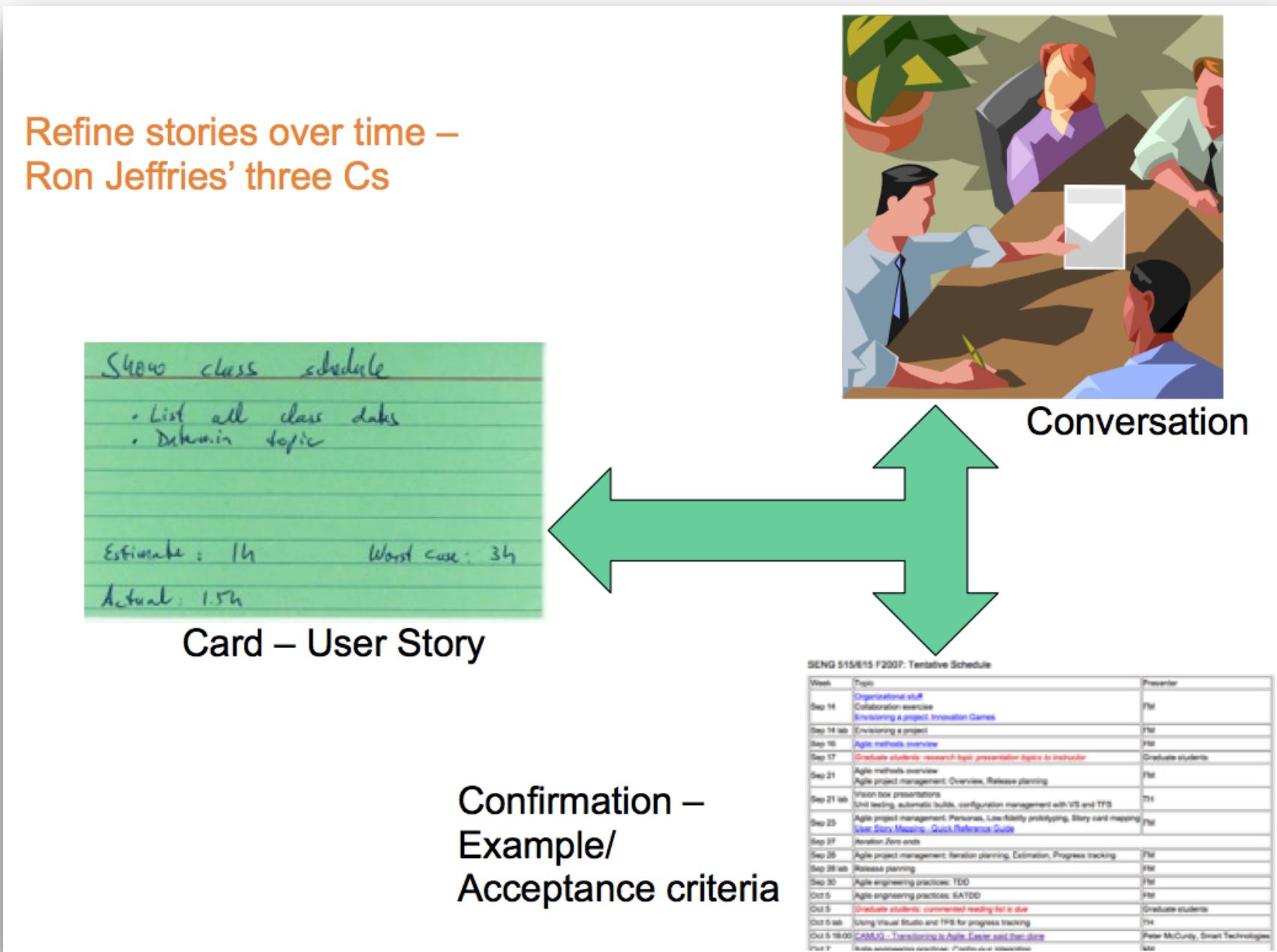
Prácticas diseño Stories



Title: User cancels reservation
Description As a user with a reservation, I want to cancel my reservation so that I get a refund. →
Success Criteria <ul style="list-style-type: none">▪ Given I am a premium member, when I cancel under 24 hours, then I incur no penalty.▪ Given I am a non-premium member, when I cancel less than 24 hours in advance, then I pay 50% fee.▪ Given I am a site member, when I cancel my reservation, then I am emailed a confirmation.
Business Value Estimate _____
Development Effort Estimate _____
ROI Estimate _____
Story Owner _____

- Vocabulario común con negocio
- Domain Driven Design

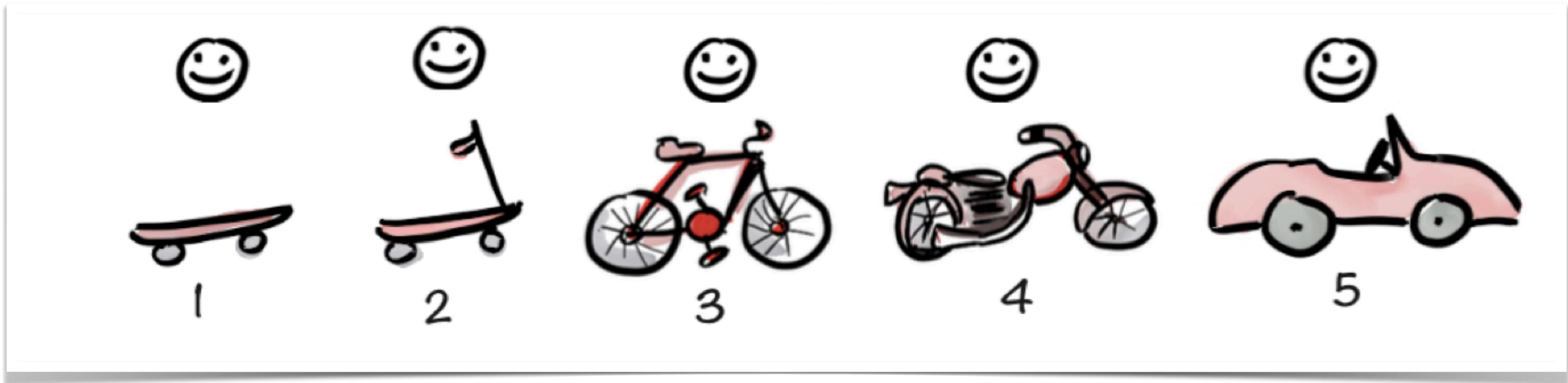
Prácticas diseño Stories



Independent
Negotiable
Valuable
Estimable
Small
Testable

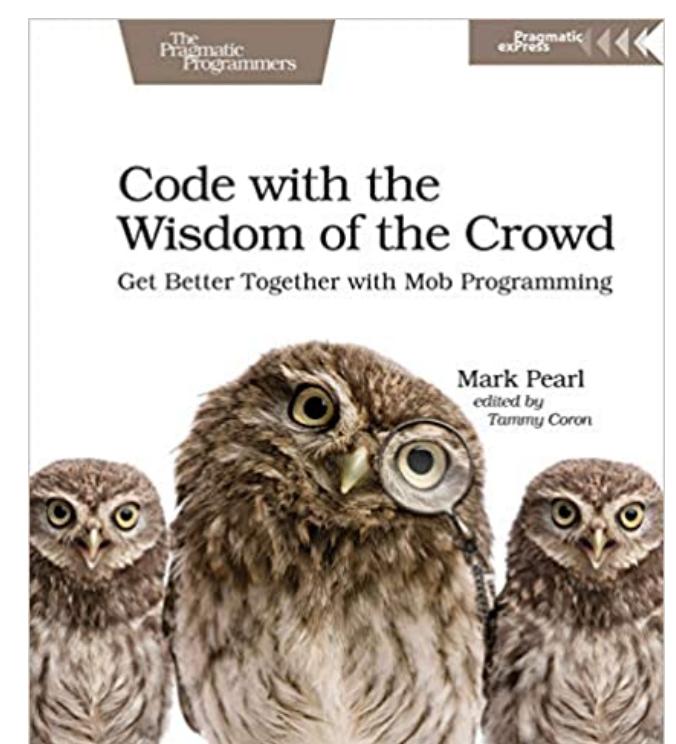
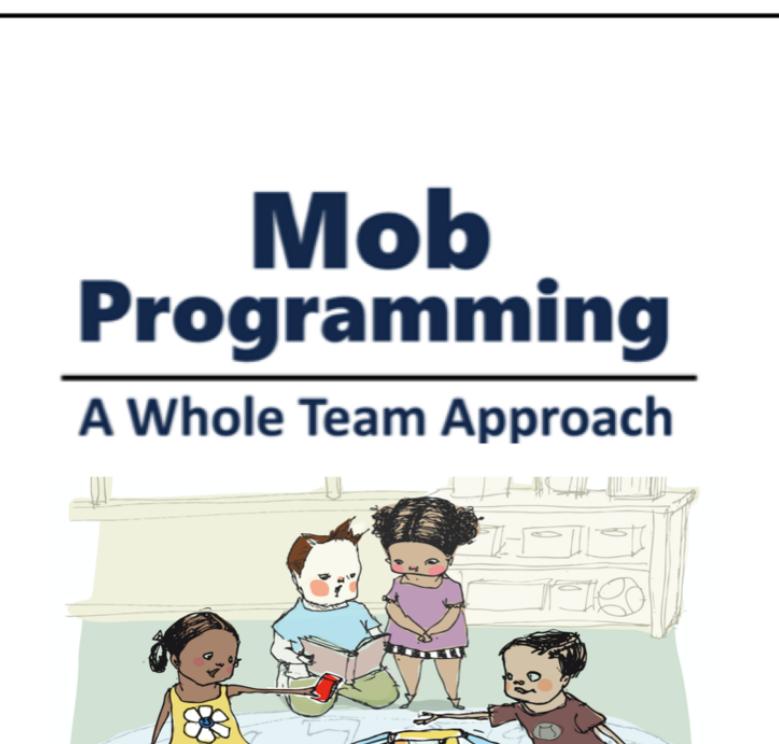
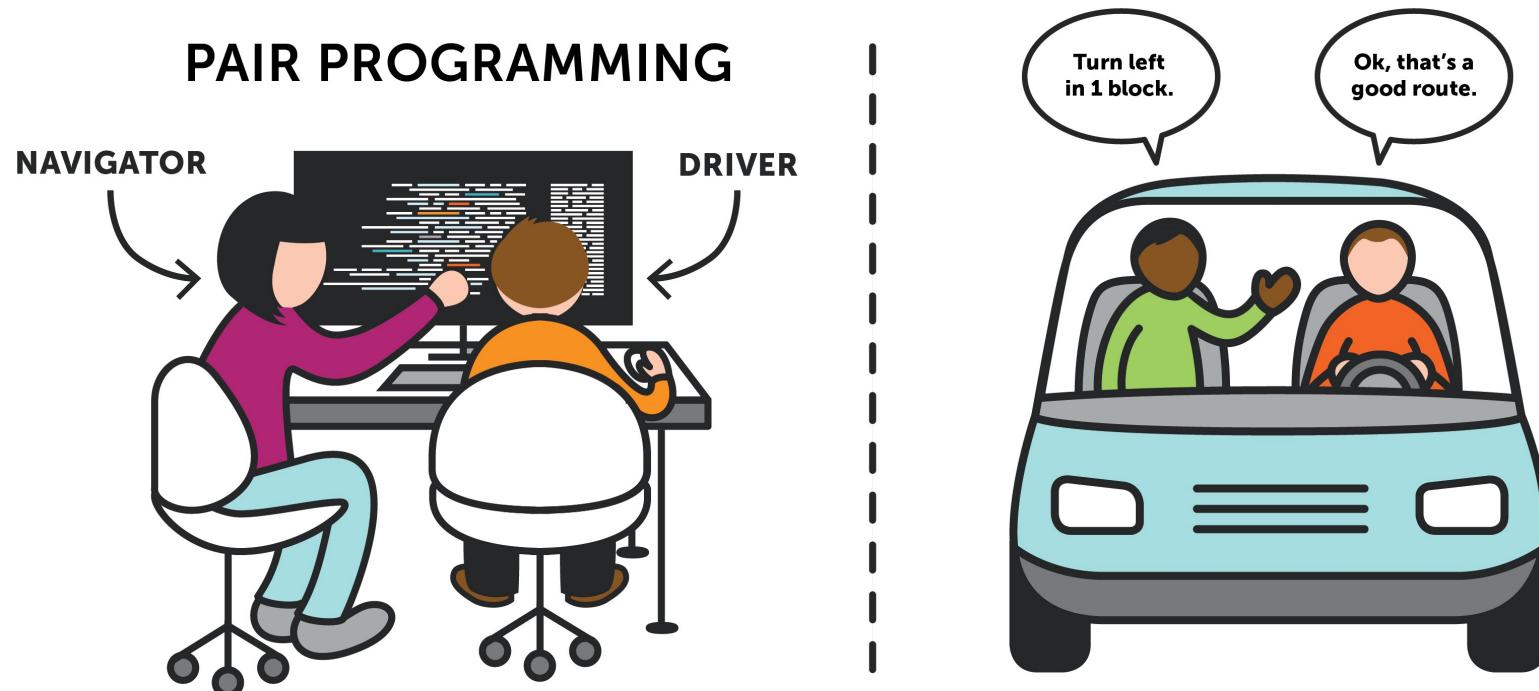
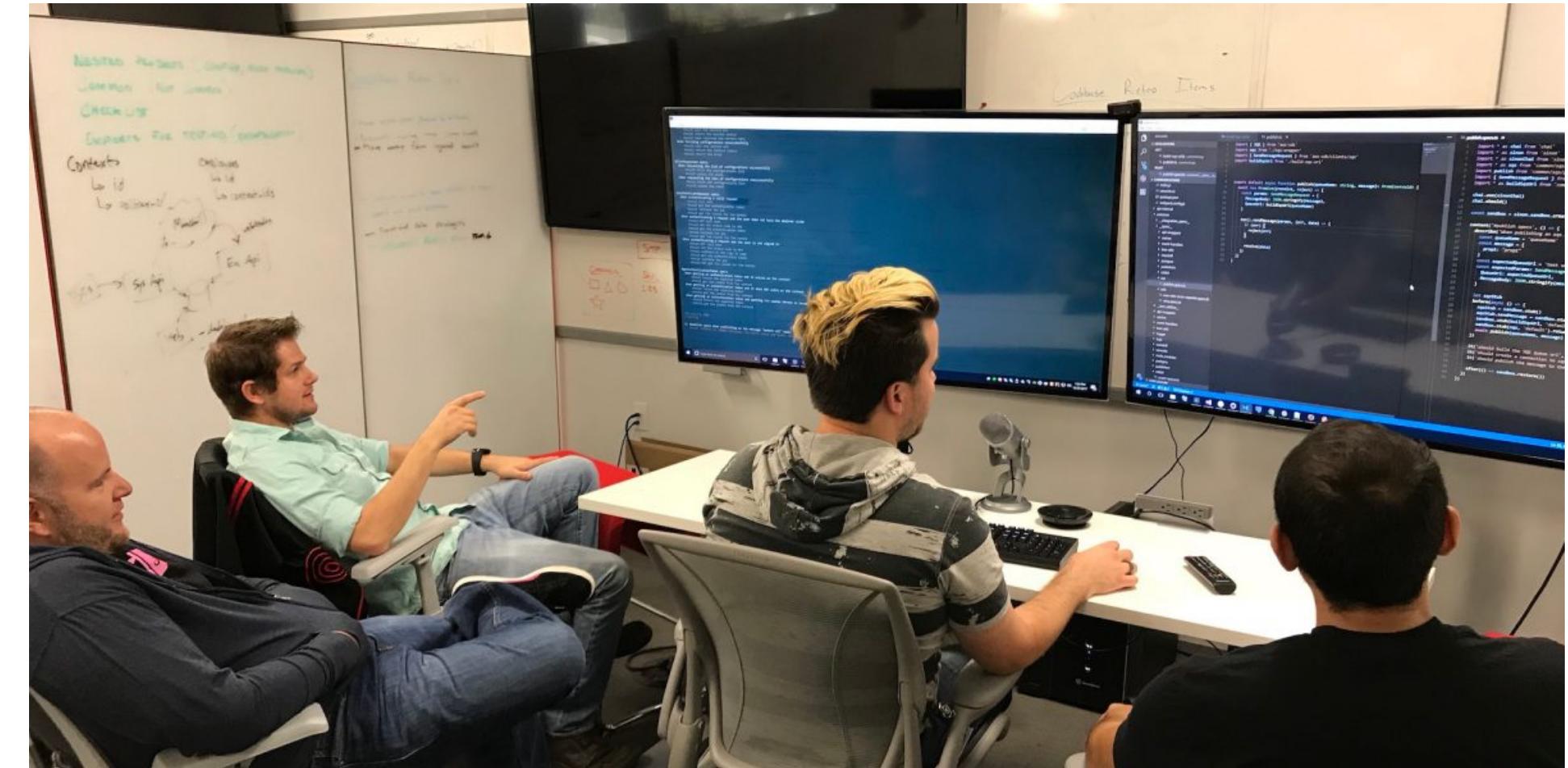
Prácticas diseño

Incremental design



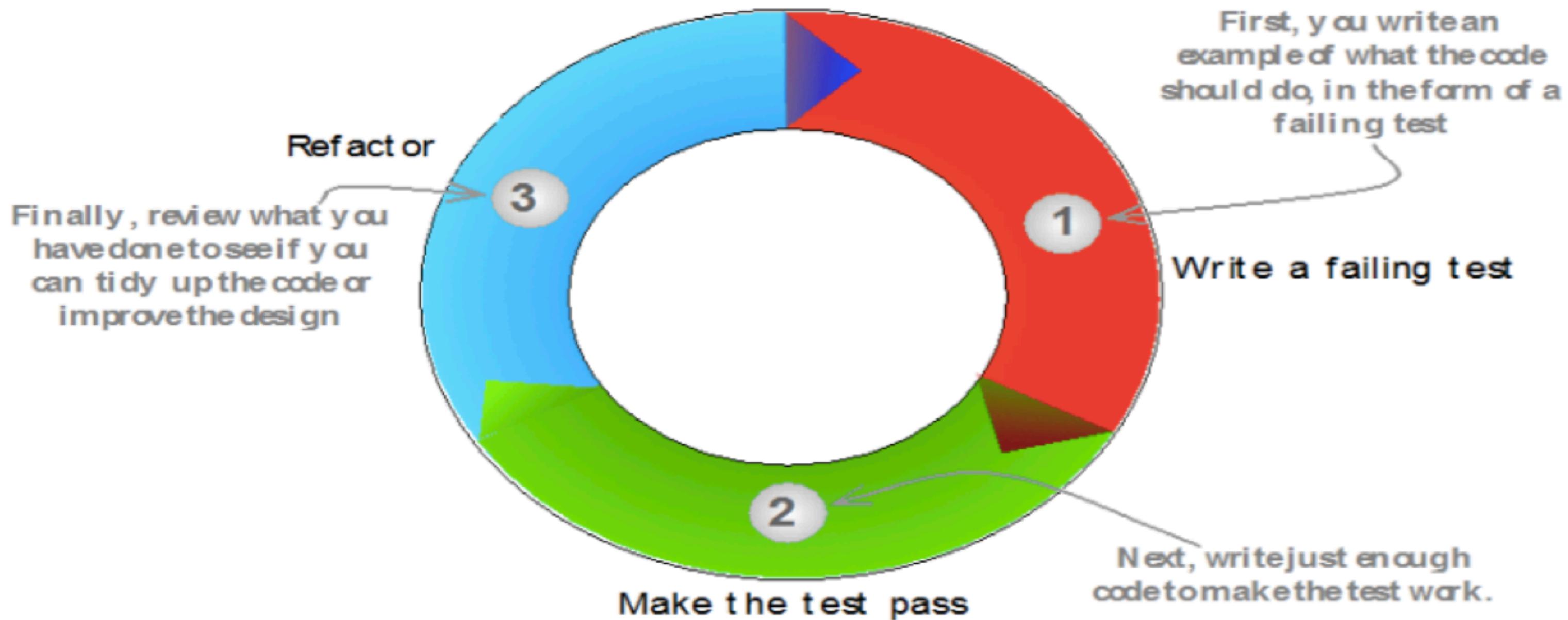
Prácticas programación

Pair programming



Prácticas programación

Test-first programming (TDD)



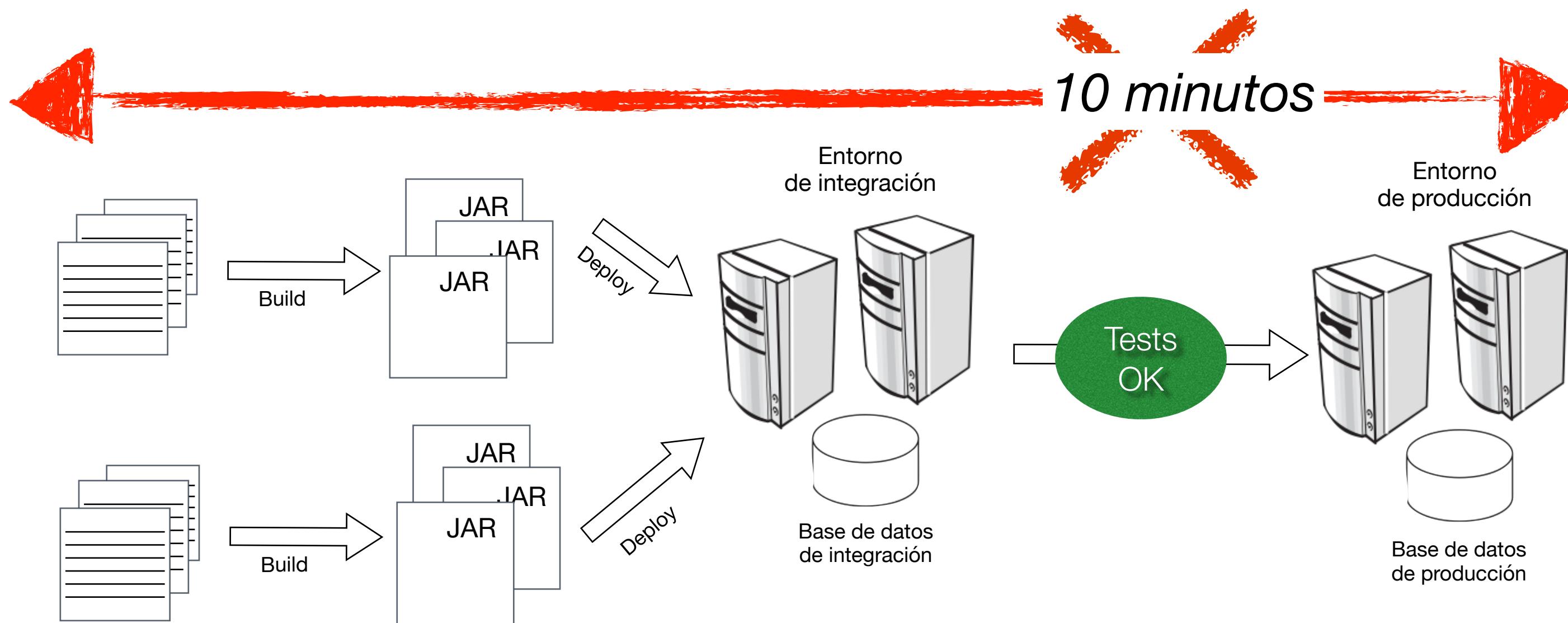
Prácticas programación

Test-first programming (TDD)

1. Se escribe un único test en el que se especifica un pequeño incremento de funcionalidad que el código debe realizar, un ejemplo de lo que el código debe hacer en forma de test. Se añade el test a los ya existentes y se lanzan todos. El test que acabamos de añadir falla (rojo).
2. Se escribe únicamente el código necesario que hace que el test pase (verde).
3. Ahora que tenemos una base de tests que funciona perfectamente, se analiza el código y los tests, se detectan problemas de diseño y se realiza una refactorización (profundizaremos más adelante en esto). Se vuelven a lanzar los tests para asegurarnos que la refactorización no ha roto nada.

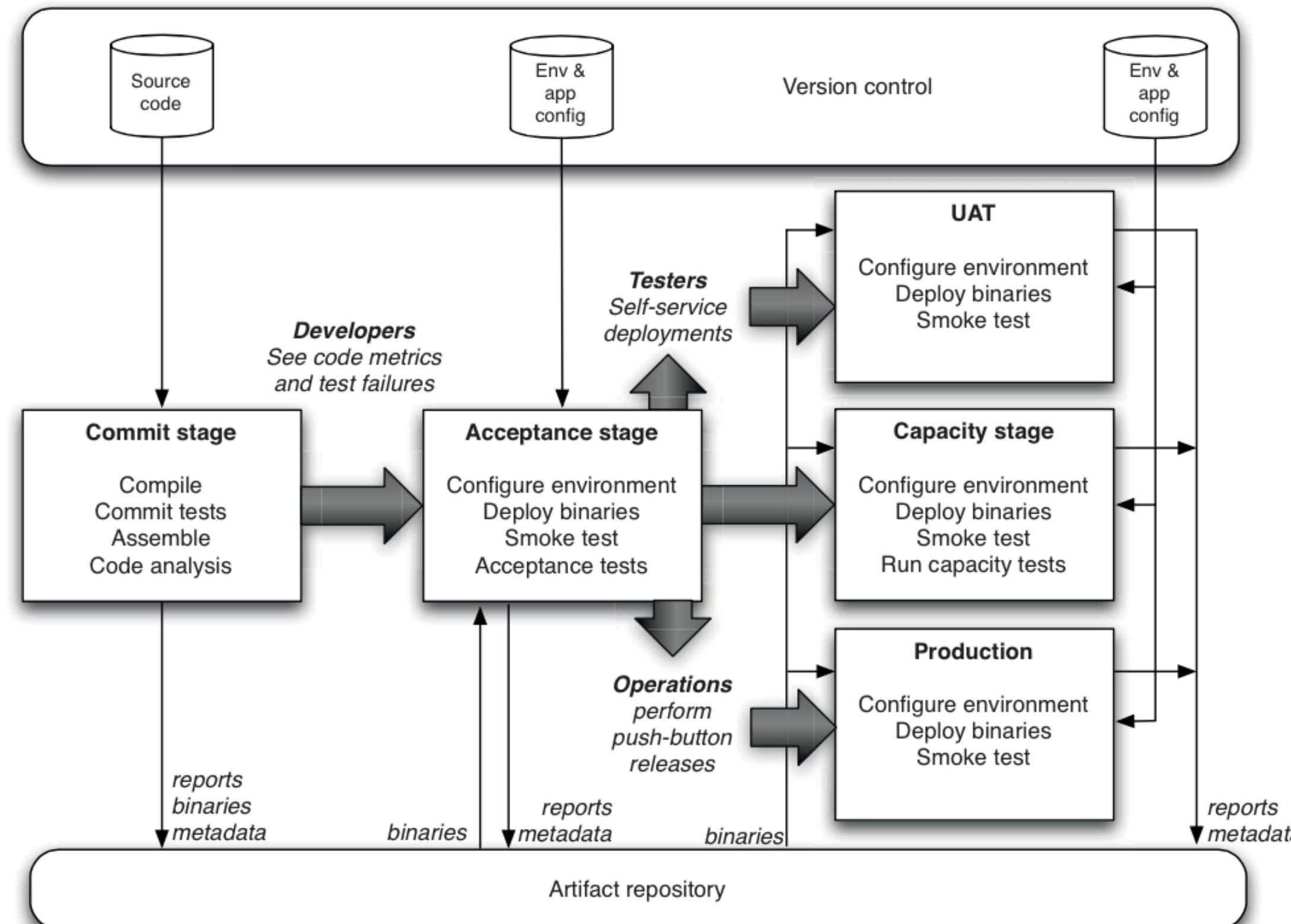
Prácticas integración

Ten-minute build



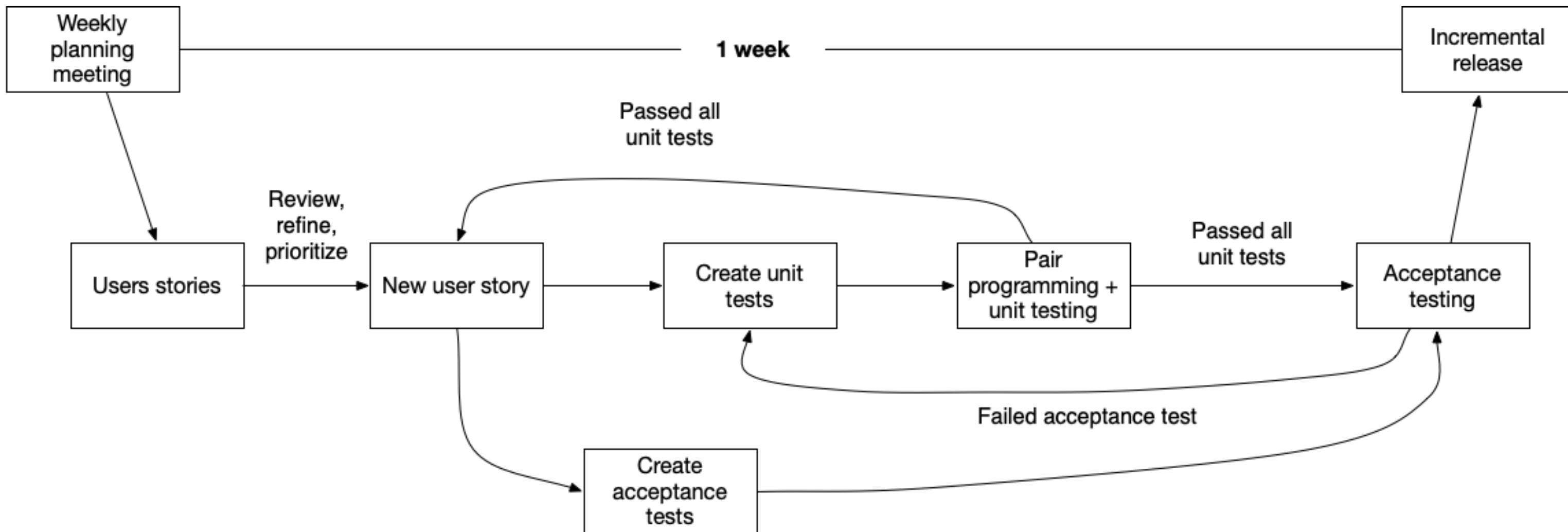
Prácticas integración

Continuous integration



Prácticas planificación

Weekly Cycle



Prácticas planificación Quaterly Cycle

- Identificar los cuellos de botella, especialmente los que no están controlados por el equipo.
- Planificar el tema o los temas para el trimestre.
- Escoger historias que sumen un trimestre para cumplir con los temas escogidos.
- Centrarse en una visión general de cómo el proyecto encaja en la organización y del valor que añade.

Prácticas secundarias

Team

- Real Customer Involvement
- Team Continuity
- Shrinking Teams

Business

- Root-Cause Analysis
- Negotiated Scope Contract
- Pay-Per-Use

Programming

- Shared Code
- Code and Test
- Single Code Base

Delivering

- Incremental Deployment
- Daily Deployment

[Code](#)[Issues](#)[Pull requests](#)[Actions](#)[Projects](#)[Wiki](#)[Security](#)[master](#)[apuntes-mads / apuntes / extreme-programming / extreme-programming.md](#)**domingogallardo** Reorganización de las prácticas

Latest commit 9ec05b0

1 contributor

850 lines (666 sloc) | 36.4 KB

[Raw](#)

eXtreme Programming

Historia de XP

El origen de Extreme Programming (XP) se encuentra en la comunidad Smalltalk, en particular entre Kent Beck y Ward Cunningham a finales de los 80 en temas relacionados con el diseño y programación orientada a objetos. Por ejemplo, en 1989, publicaron el artículo [A laboratory for teaching object-oriented design](#).