Expand Contract Pattern

Continuous Delivery for Databases

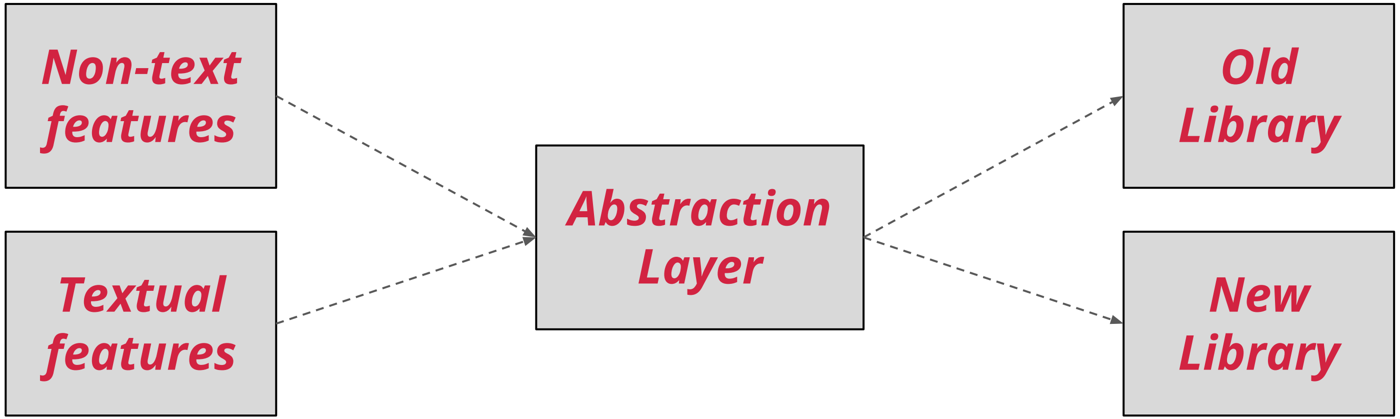
@leenasn

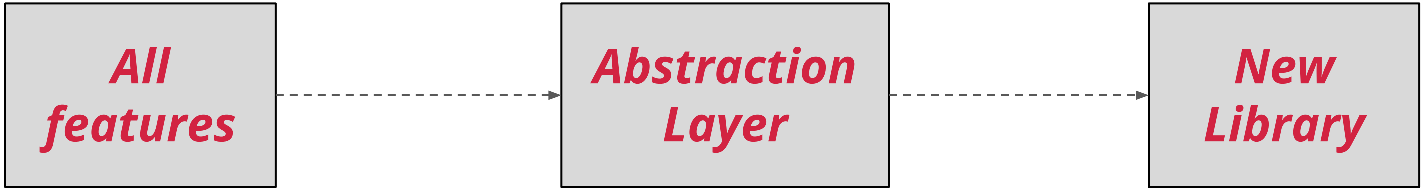
https://pixabay.com/en/hands-hand-raised-hands-raised-220163/

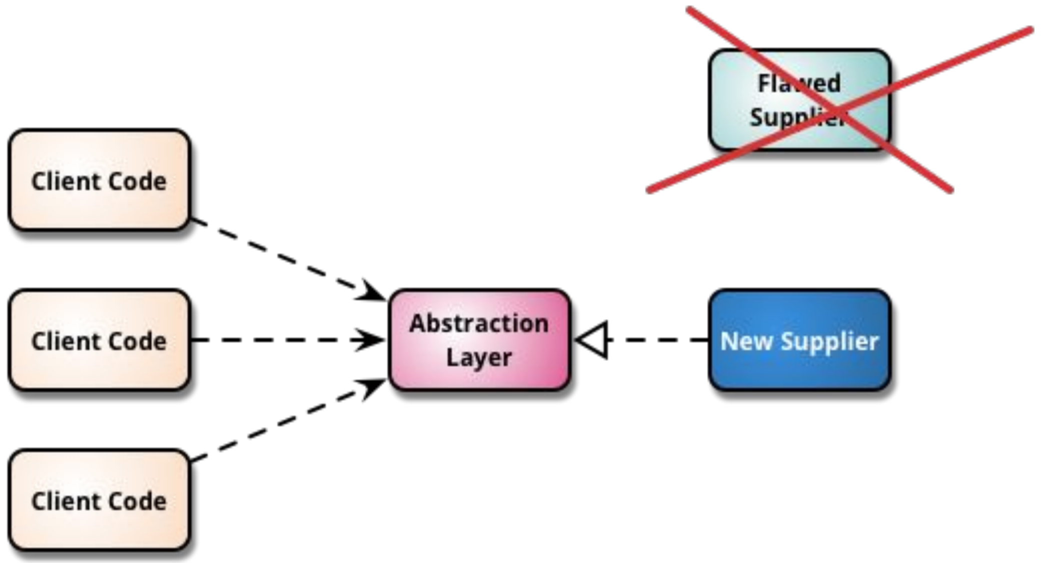
A few years back

https://pixabay.com/en/art-watercolors-arts-and-crafts-1851483/

https://pixabay.com/photos/time-for-a-change-new-ways-letters-2015164/


Non-text features
Textual features
Abstraction Layer
Old Library
New Library


All features
Abstraction Layer
New Library



Branch by Abstraction

https://martinfowler.com/bliki/BranchByAbstraction.html

Refactoring is a controlled technique for improving the design of an existing code base. Its essence is applying ***a series of small behavior-preserving transformations***, each of which "too small to be worth doing".

However the cumulative effect of each of these transformations is quite significant.

Refactoring

https://martinfowler.com/books/refactoring.html

Ability to get changes of all types—including new features, configuration changes, bug fixes and experiments—into production, or into the hands of users, *safely* and *quickly* in a *sustainable* way.

Continuous Delivery

https://continuousdelivery.com/

https://pixabay.com/photos/hourglass-time-hours-clock-620397/

Good Karma

Story-telling CRM because **Facts Tell but Stories Sell**

http://goodkarmayoga.in/

Evolvability

https://www.dreamstime.com/

Database?

https://pixabay.com/vectors/cranium-head-human-people-persons-2028555/

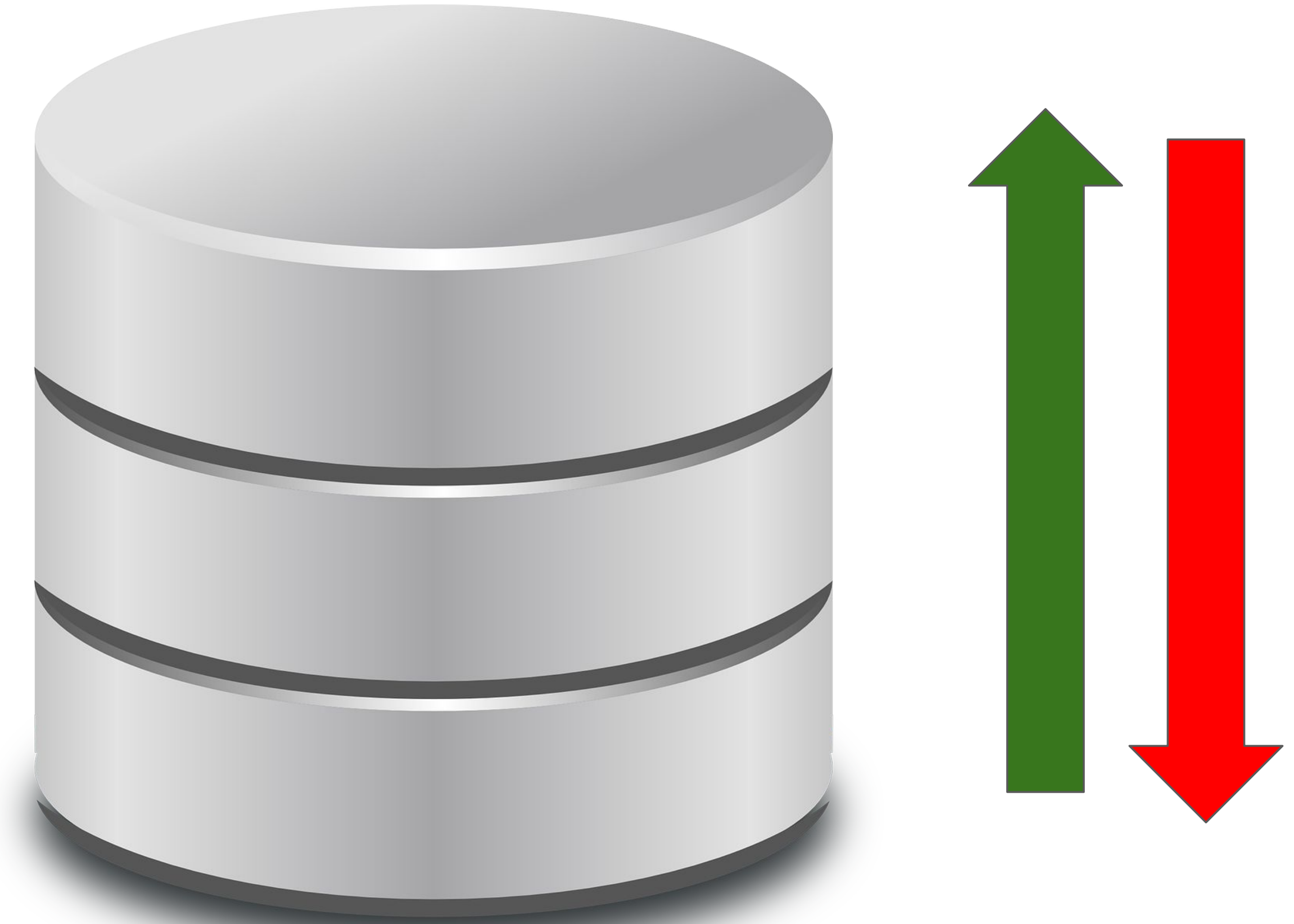
Slow

https://unsplash.com/photos/elayN_YscVg

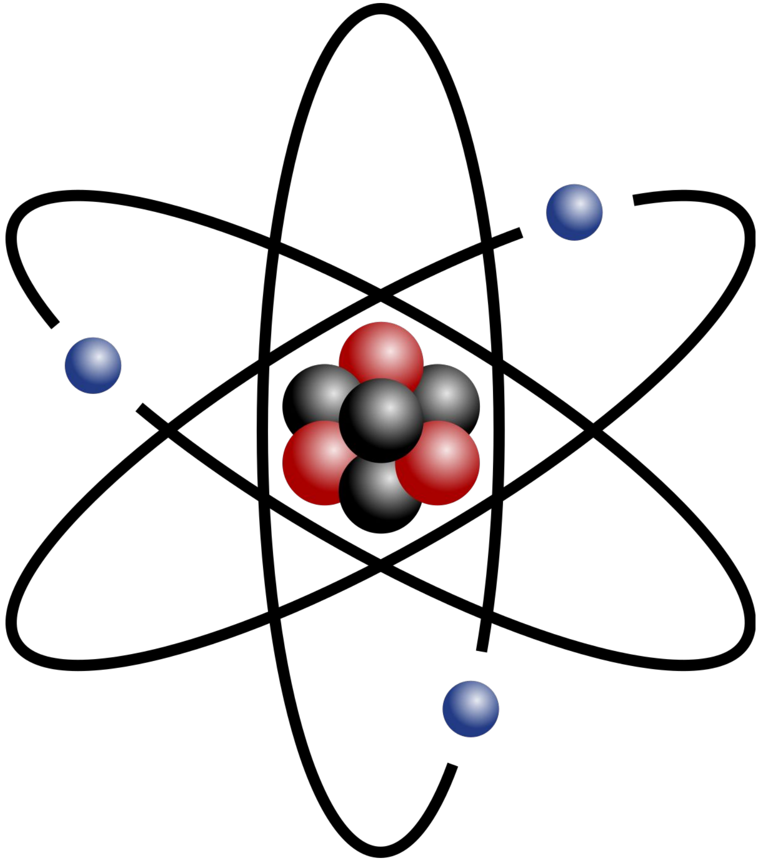
High risk

https://pixabay.com/en/balance-risk-courage-risky-3062272/

ACID



https://pixabay.com/vectors/database-storage-data-storage-152091/



Atomic Deployments

https://pixabay.com/en/juggle-artists-circus-money-1027844/

Safe

https://pixabay.com/photos/lifesaver-life-buoy-safety-rescue-933560/

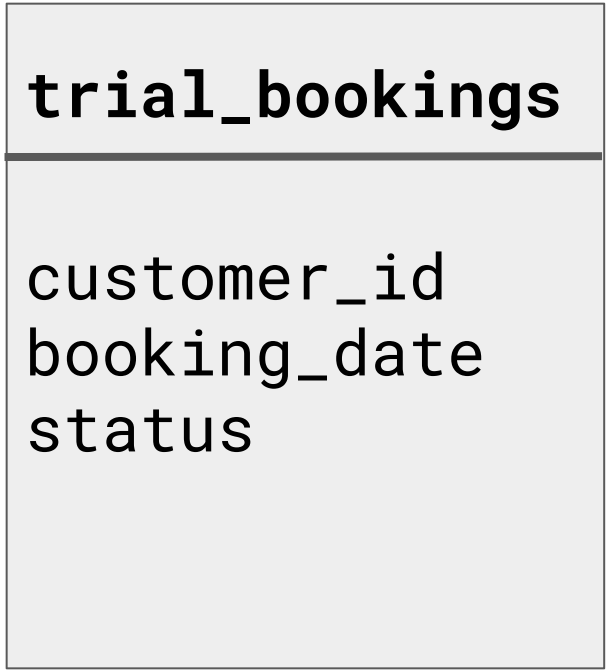
Less risky

A ***database refactoring*** is a small change to your database schema (the table structures, data itself, stored procedures, and triggers) which improves its design without changing its semantics.

http://databaserefactoring.com/

- Enables ***Continuous Delivery***
- Supports ***evolutionary*** development

https://www.dreamstime.com/stock-photos-example-stamp-image28420393

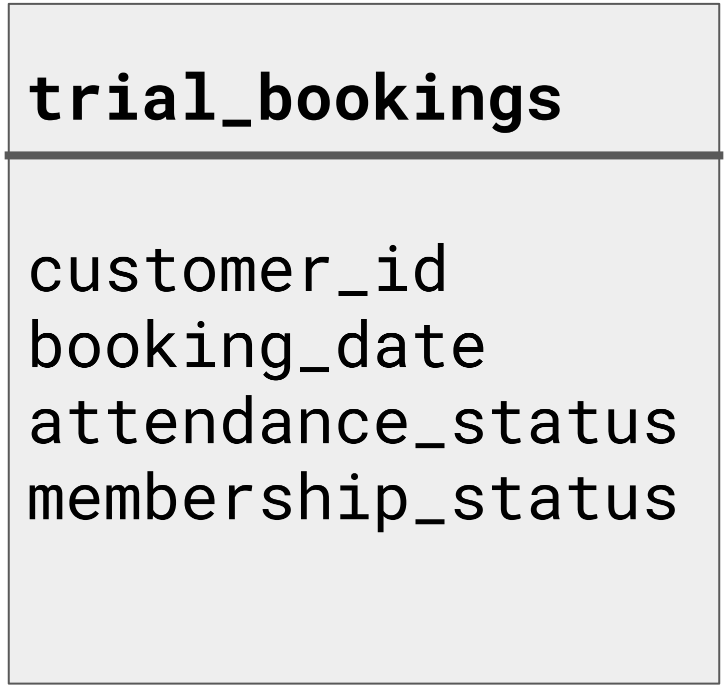


Split Column

trial_bookings

```
customer_id
status[old field]
booking_date
attendance_status
membership_status
```

Transition



trial_bookings

customer_id
booking_date
attendance_status
membership_status

New Schema

```
trial_bookings
--------------------
customer_id
booking_date
attendance_status
membership_status
```

Old Schema

```
trial_bookings
--------------------
customer_id
status[old field]
booking_date
attendance_status
membership_status
```

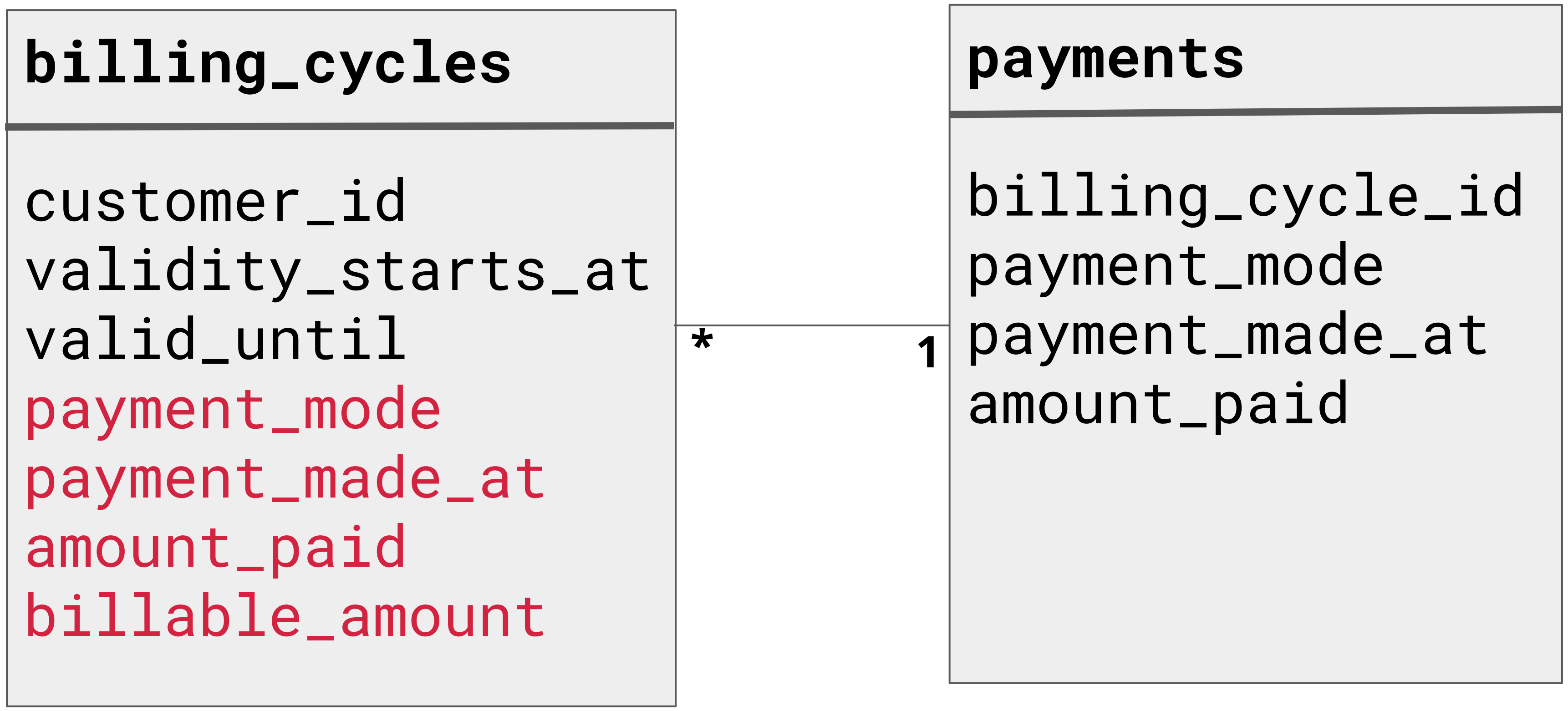
Transition Period

```
trial_bookings
--------------------
customer_id
booking_date
attendance_status
membership_status
```

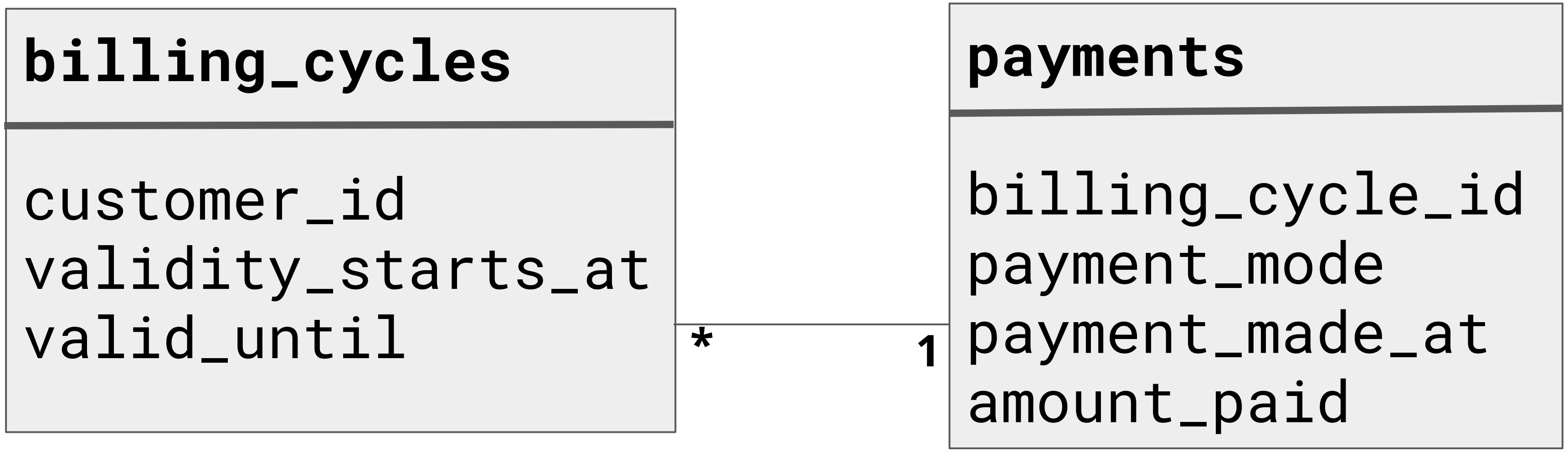
New Schema

billing_cycles
customer_id
validity_starts_at
valid_until
payment_mode
Payment_made_at
amount_paid

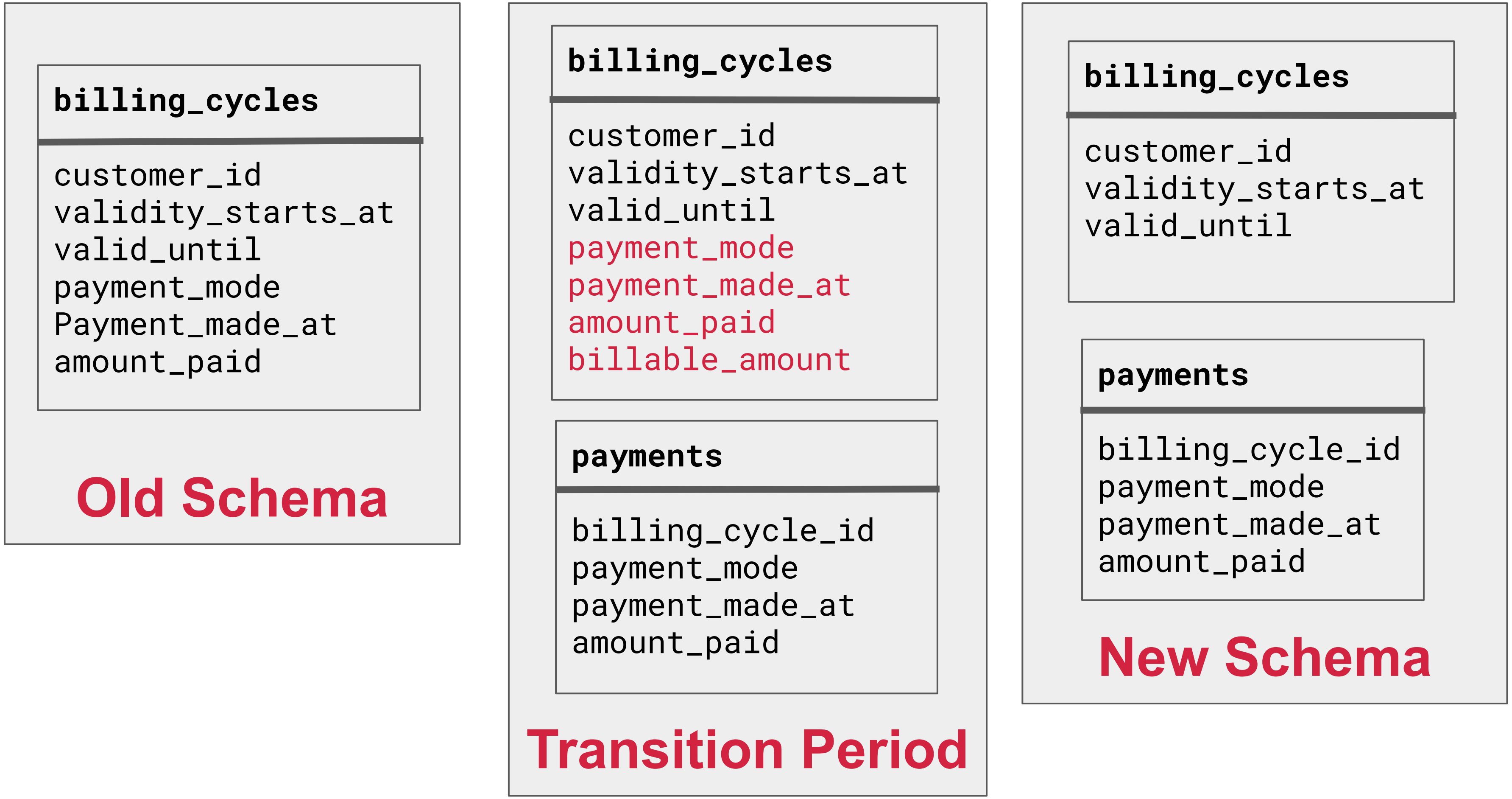
Split table

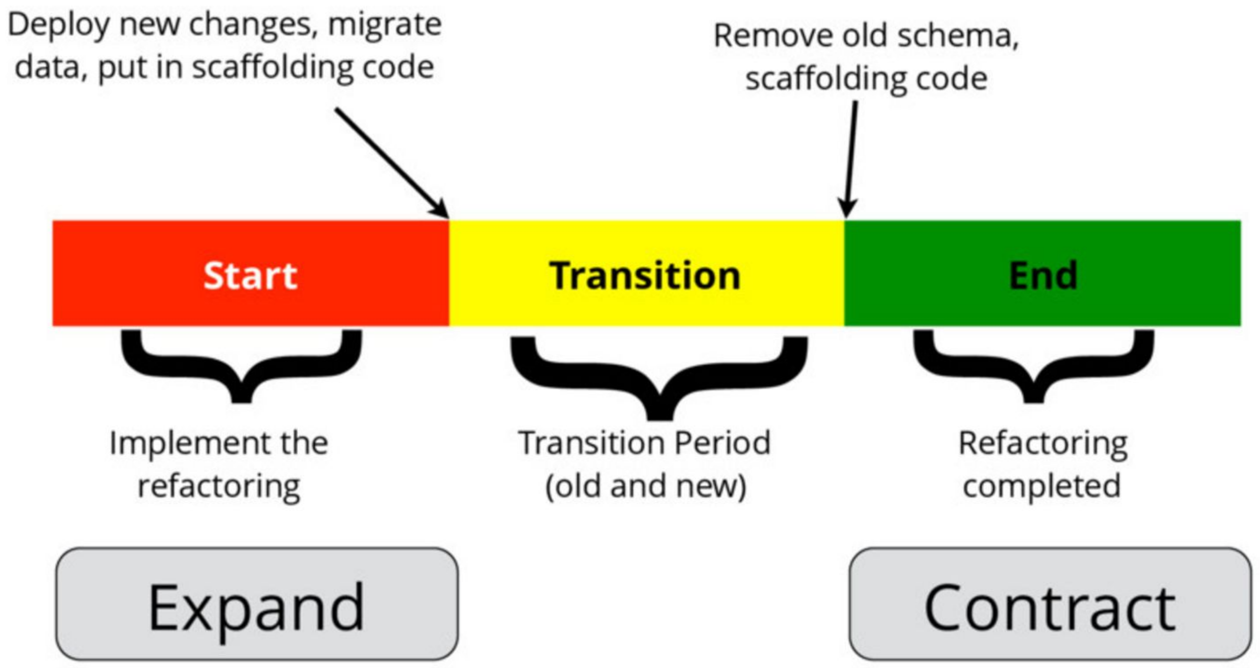

billing_cycles
customer_id
validity_starts_at
valid_until
payment_mode
payment_made_at
amount_paid
billable_amount
*
1
payments
billing_cycle_id
payment_mode
payment_made_at
amount_paid

Transition


billing_cycles
customer_id
validity_starts_at
valid_until
*
1
payments
billing_cycle_id
payment_mode
payment_made_at
amount_paid

New Schema

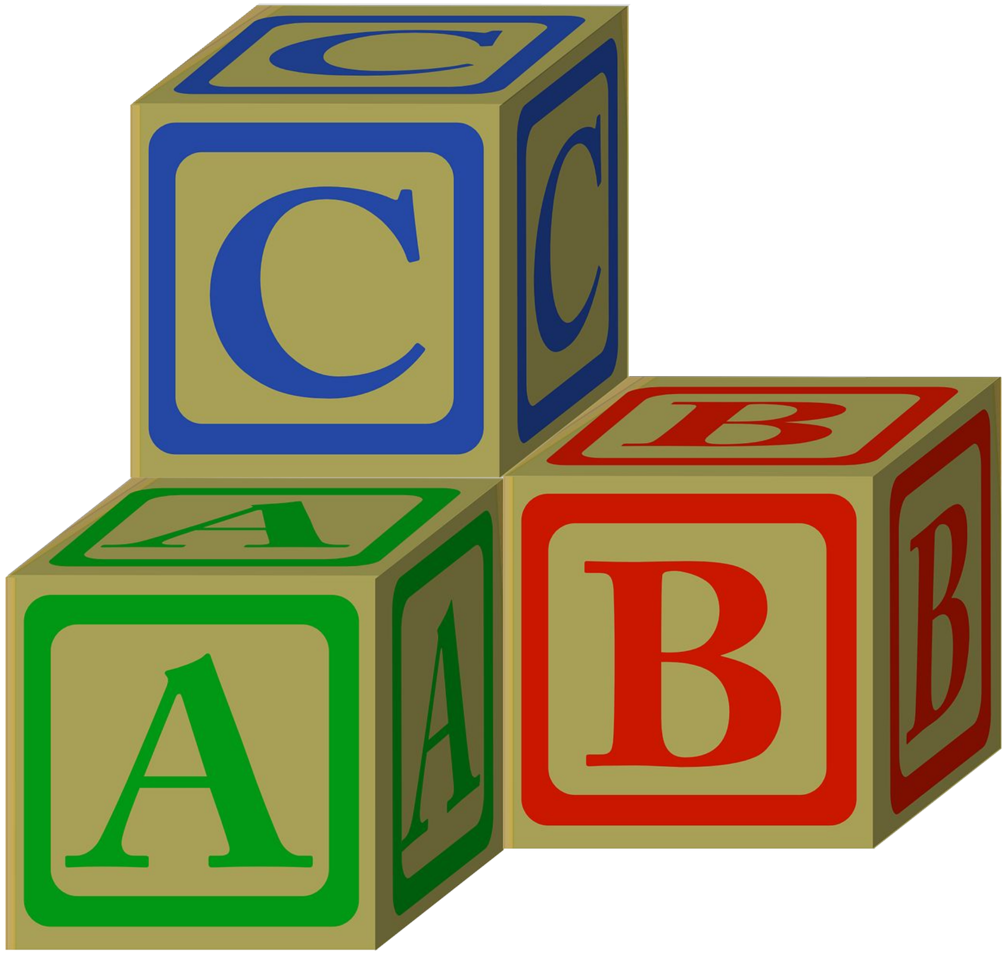

billing_cycles
customer_id
validity_starts_at
valid_until
payment_mode
Payment_made_at
amount_paid
Old Schema
billing_cycles
customer_id
validity_starts_at
valid_until
payment_mode
payment_made_at
amount_paid
billable_amount
payments
billing_cycle_id
payment_mode
payment_made_at
amount_paid
Transition Period
billing_cycles
customer_id
validity_starts_at
valid_until
payments
billing_cycle_id
payment_mode
payment_made_at
amount_paid
New Schema


Deploy new changes, migrate data, put in scaffolding code
Remove old schema, scaffolding code
Start
Transition
End
Implement the refactoring
Transition Period (old and new)
Refactoring completed
Expand
Contract

https://martinfowler.com/articles/evodb.html

- ★ Structural Refactoring
- ★ Data Quality Refactoring
- ★ Referential Integrity Refactoring
- ★ Transformation
- ★ Architectural Refactoring
- ★ Method Refactoring

Types of Refactoring



Basics

https://pixabay.com/vectors/abc-alphabet-blocks-toy-148517/

Versioning

https://git-scm.com/downloads/logos

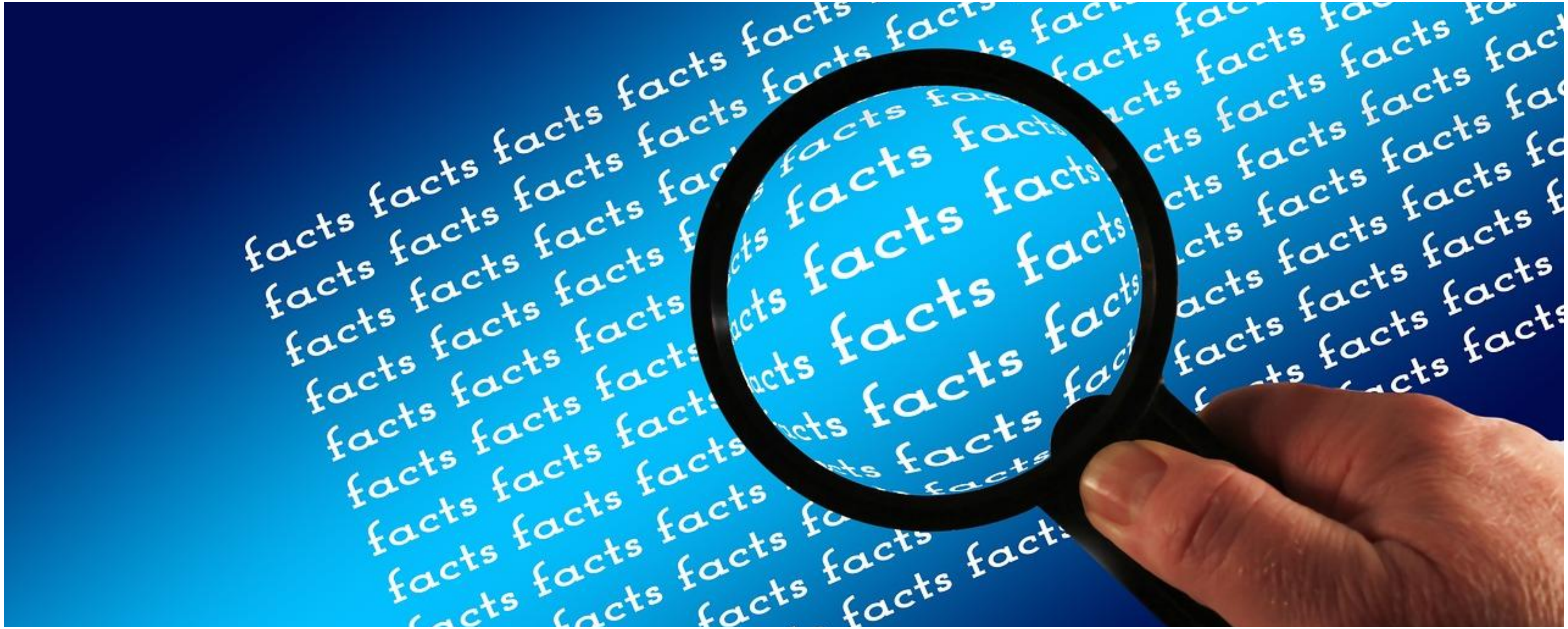
Automation

https://pixabay.com/photos/gears-cogs-machine-machinery-1236578/

https://pixabay.com/en/strategy-arrows-building-startup-3224515/

Transition period

https://pixabay.com/photos/bridge-iron-old-pedestrian-metal-3181356/



https://pixabay.com/en/magnifying-glass-facts-examine-1607160/

GitHub

Move fast and break nothing

https://zachholman.com/talk/move-fast-break-nothing/

Upgrading Rails

https://github.blog/2018-09-28-upgrading-github-from-rails-3-2-to-5-2/

Software Releases Without Major Problems

https://www.forbes.com/sites/quora/2013/08/12/how-do-facebook-and-google-manage-software-releases-without-causing-major-problems/

Strangler Applications

https://paulhammant.com/2013/07/14/legacy-application-strangulation-case-studies/

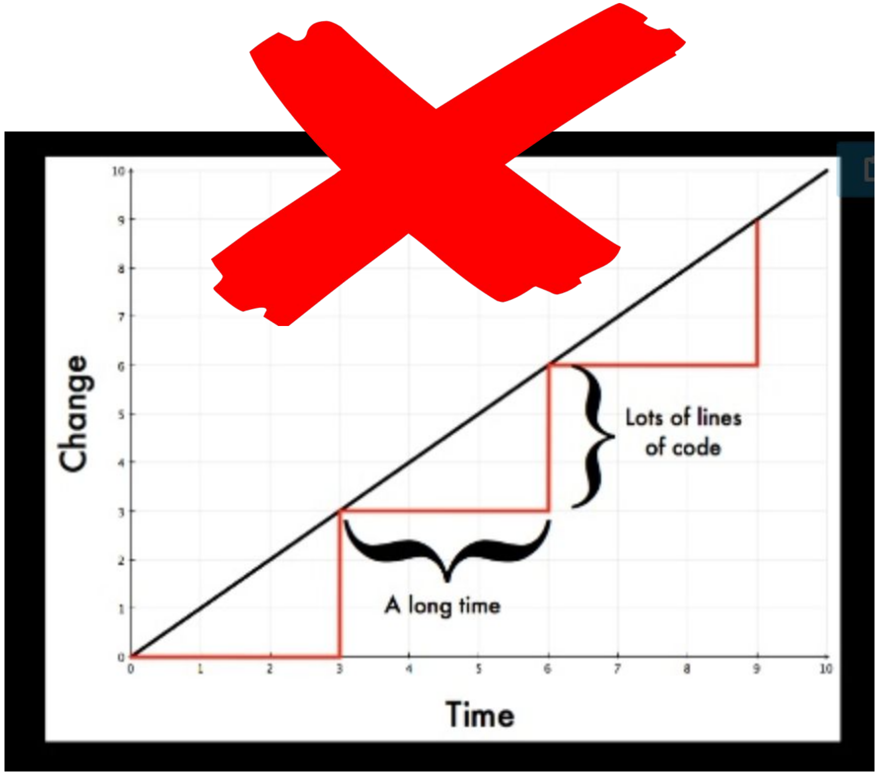
SUMMARY

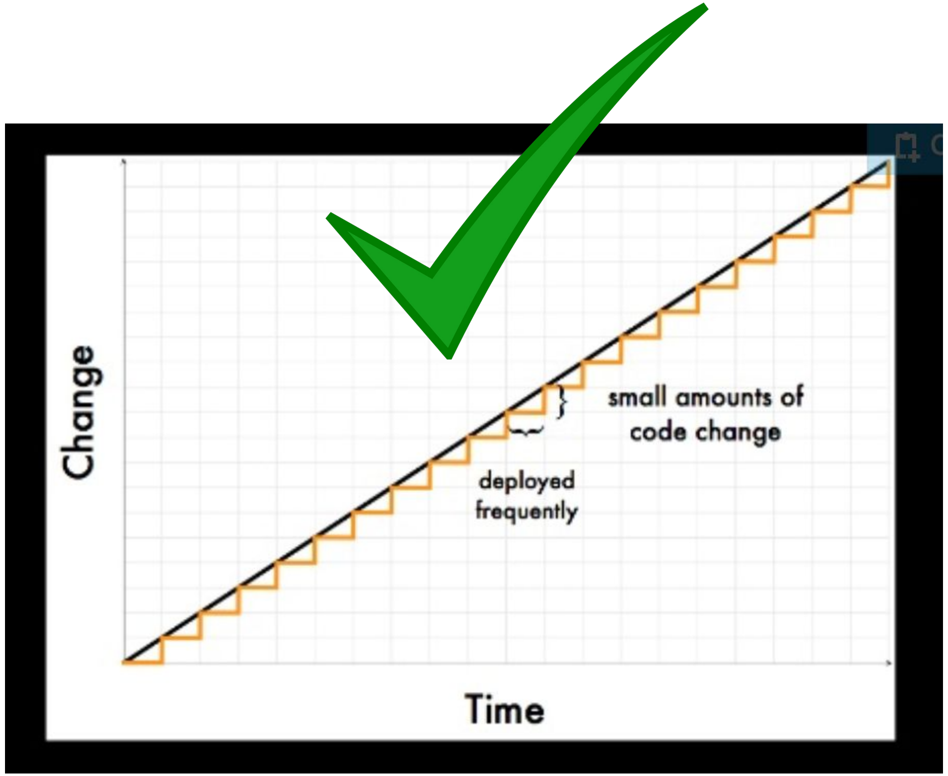
Murphy’s Law

https://www.flickr.com/photos/smemon/5471883005

Low risk releases

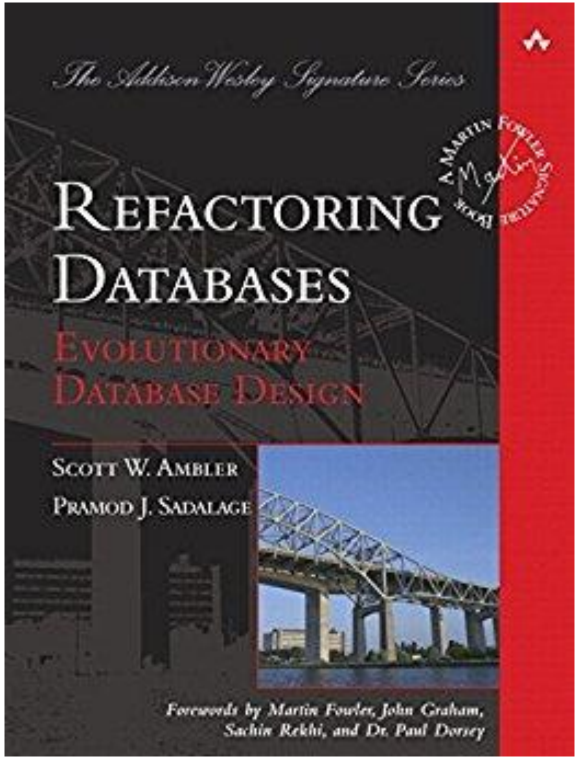
https://pixabay.com/en/button-red-round-shiny-155149/





Small reversible steps

https://www.slideshare.net/jallspaw/ops-metametrics-the-currency-you-pay-for-change


The Addison Wesley Signature Series

A Martin Fowler Signature Book

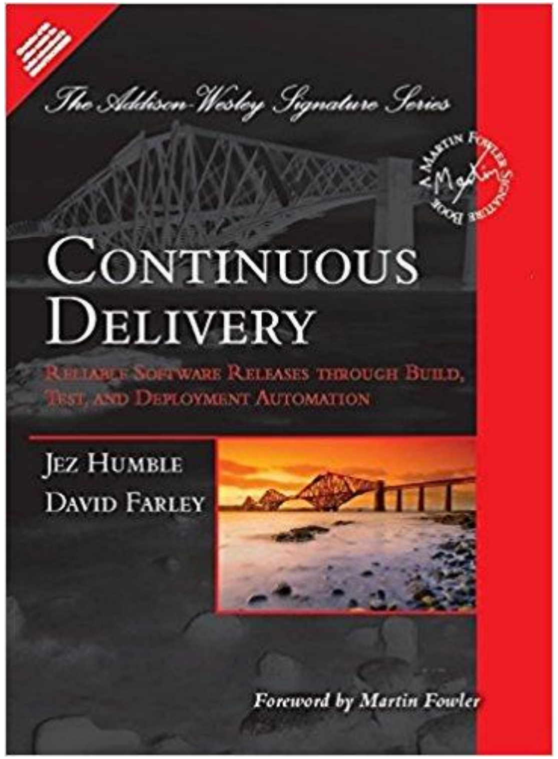
Refactoring Databases

Evolutionary Database Design

Scott W. Ambler

Pramod J. Sadalage

Forewords by Martin Fowler, John Graham, Sachin Rekhi, and Dr. Paul Dorsey


The Addison-Wesley Signature Series

A Martin Fowler Signature Book

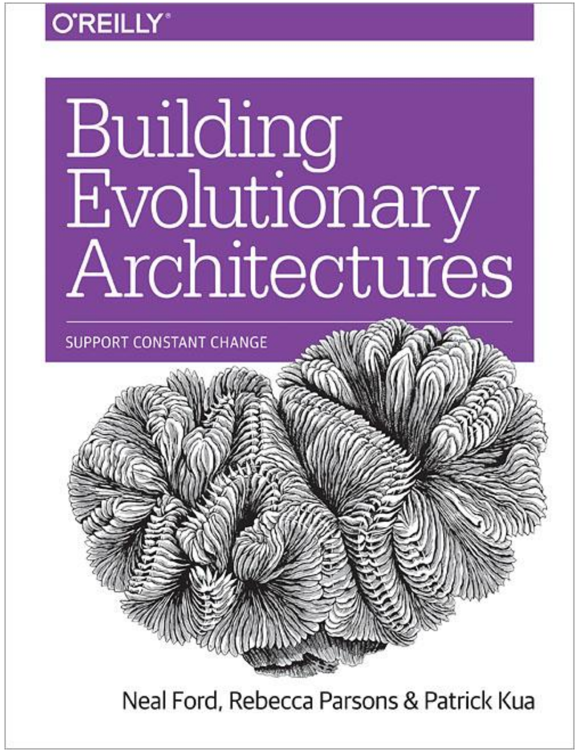
Continuous Delivery

Reliable Software Releases through Build, Test, and Deployment Automation

Jez Humble

David Farley

Foreword by Martin Fowler


O'REILLY

Building Evolutionary Architectures

SUPPORT CONSTANT CHANGE

Neal Ford, Rebecca Parsons & Patrick Kua

References

Atomicity and Ratcheting
Split Table - Database Refactoring
Continuous Delivery of Databases
Expand Contract Pattern - Continuous Delivery of Databases
Database Refactoring
Four Principles of Low-Risk Software Releases
Branch by Abstraction
Move Fast @ Github
Legacy Application Strangulation
Make Large Scale Changes Incrementally with Branch By Abstraction

References

Leena S N

@leenasn / leena.sn@multunus.com

https://medium.com/@leenasn