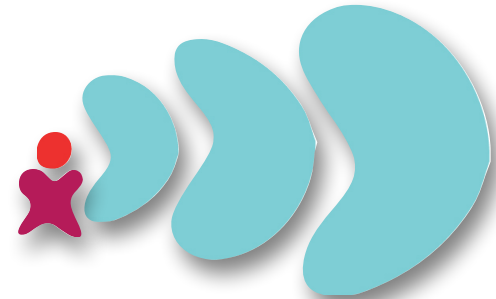
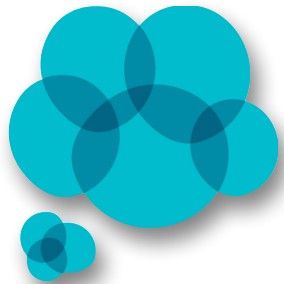


WHeRe Do idEaS CoMe FRoM?



Creating, Cultivating, & Communicating Intellectual Property



ThoughtWorks®

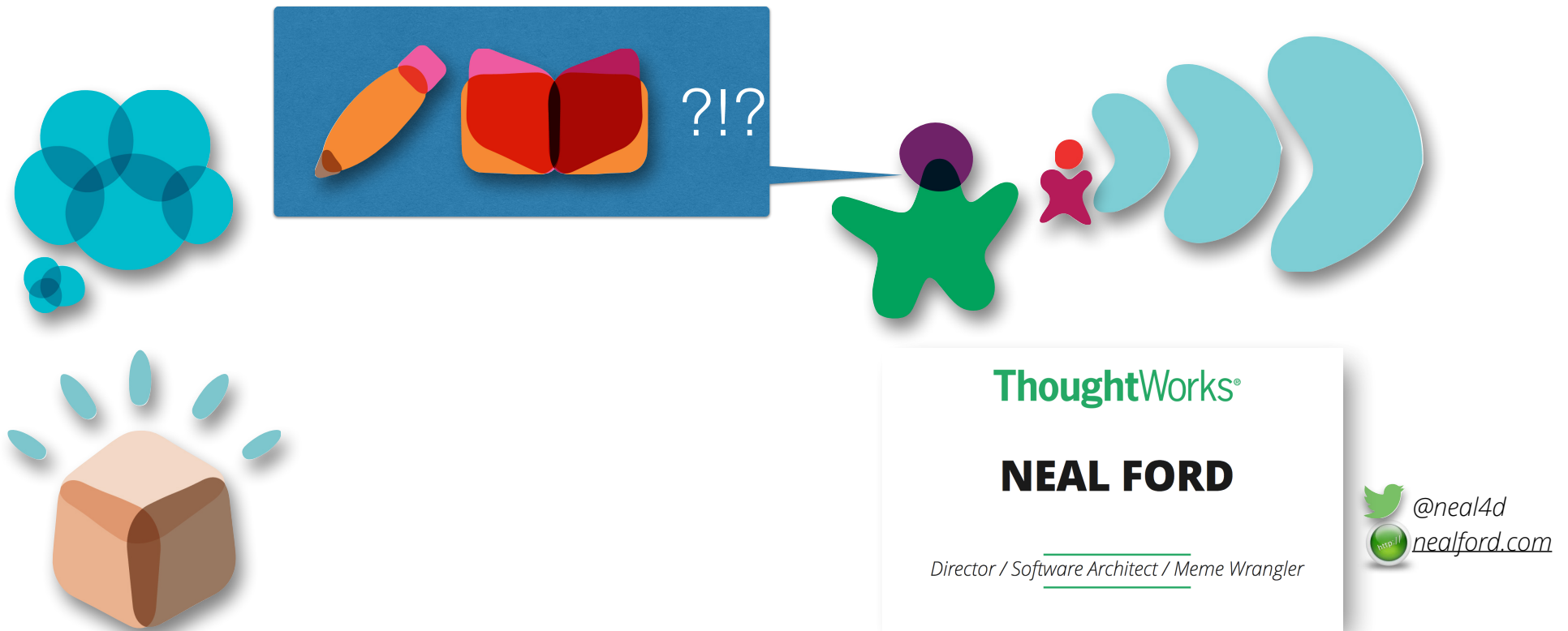
NEAL FORD

Director / Software Architect / Meme Wrangler

 [@neal4d](#)
 nealford.com

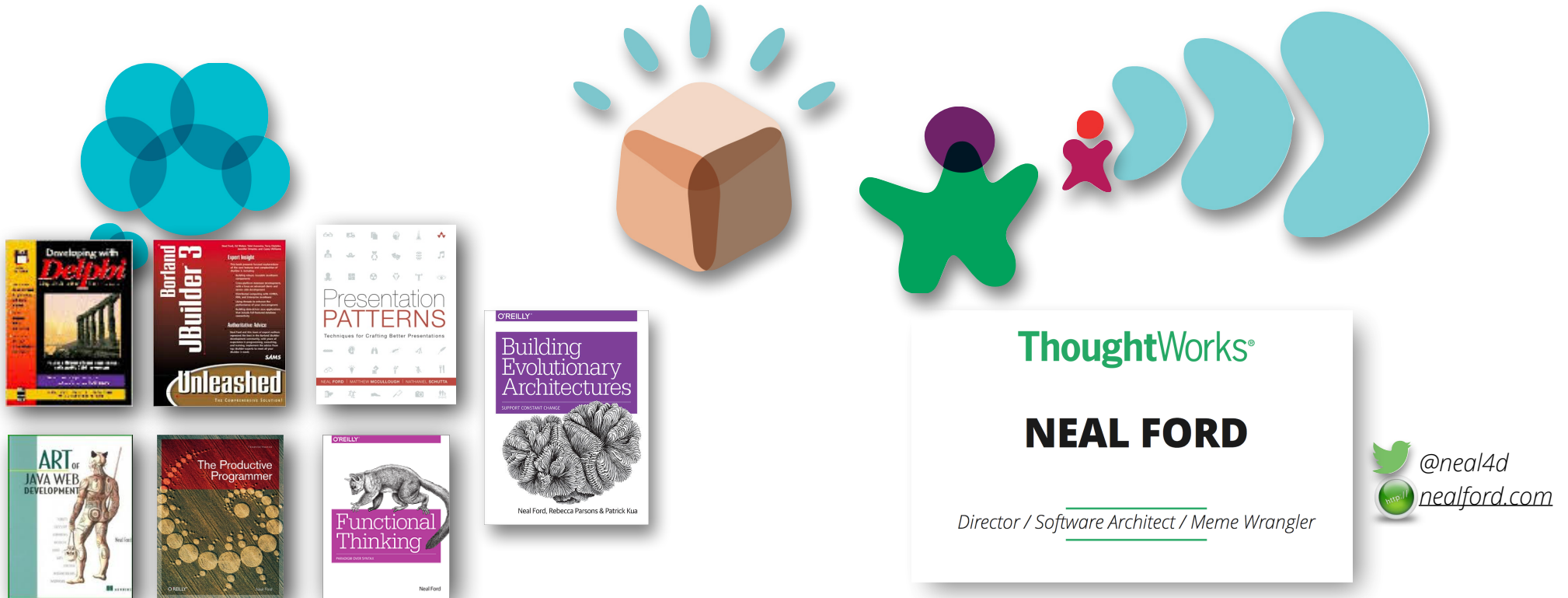
WheRe did thiS TaLk coMe FRoM?

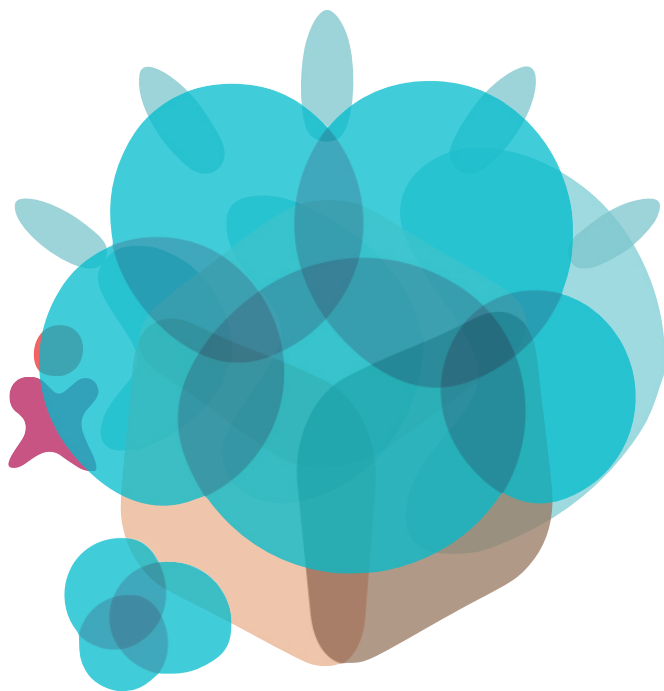
Creating, Cultivating, & Communicating Intellectual Property



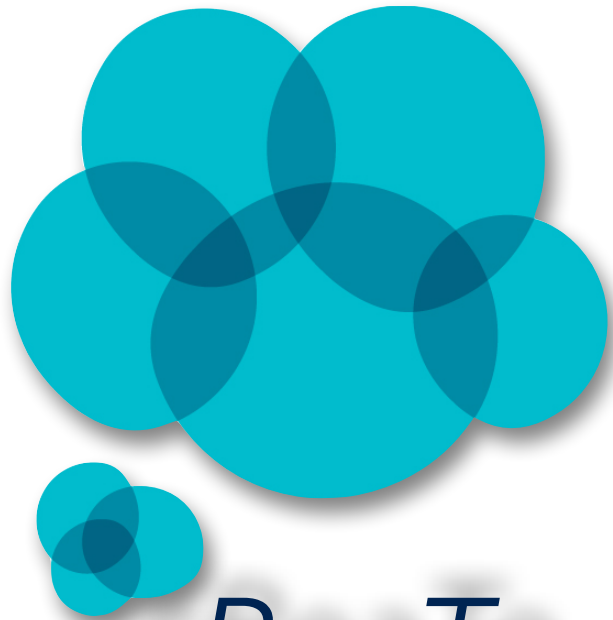
Where did this Talk come From?

Creating, Cultivating, & Communicating Intellectual Property

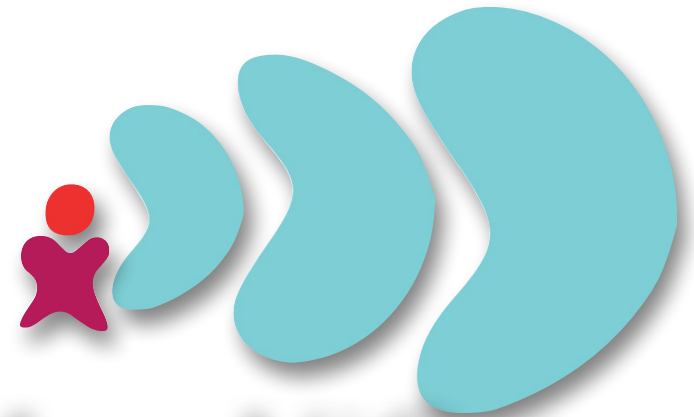




cuLtiVaTe



cReaTe

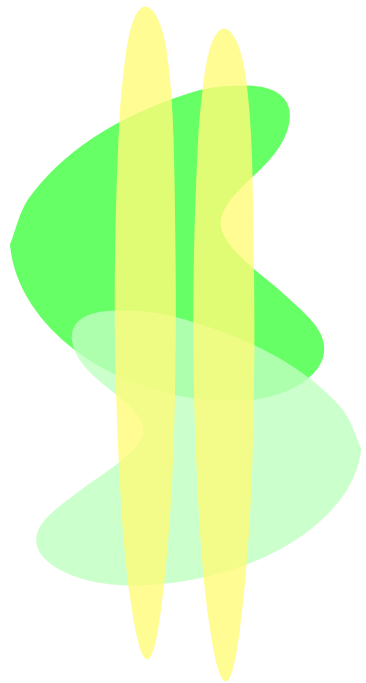


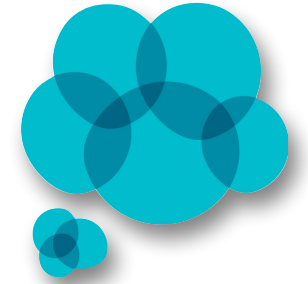
coMmuNiCate



cReaTe

Who owns your IP ?





cReaTe

comparison/
survey

symbiosis

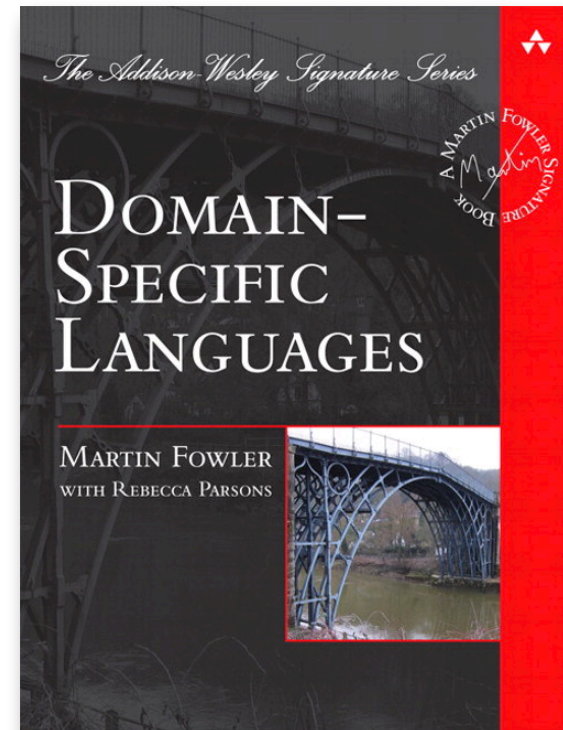
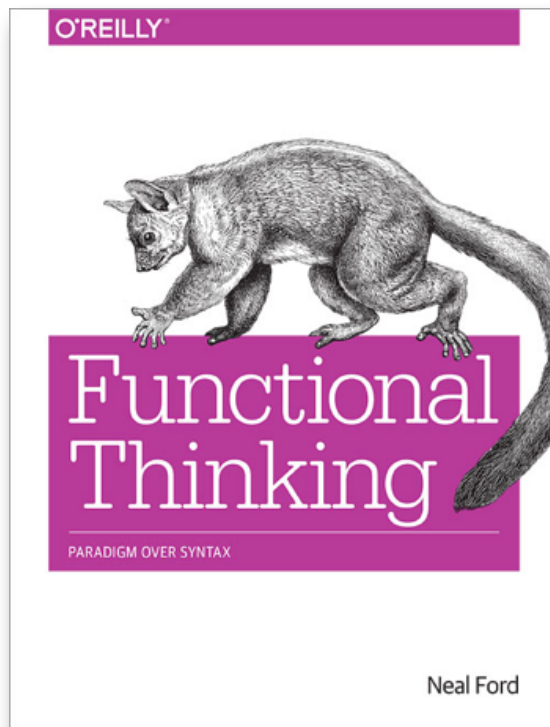
original IP

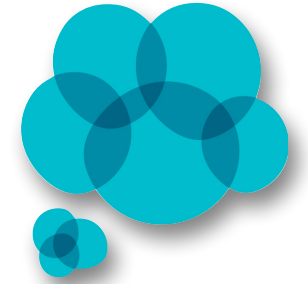
metaphorical
mapping

catalog



comparison/ survey





cReaTe

comparison/
survey

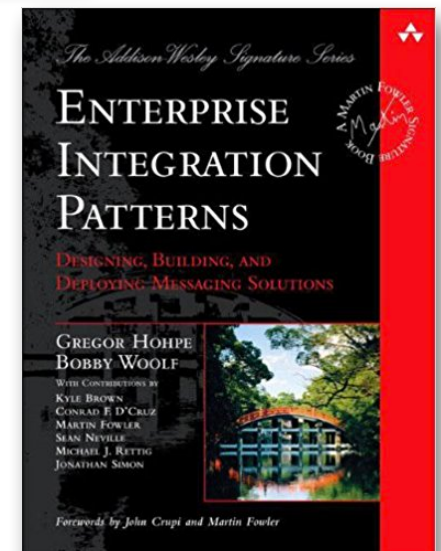
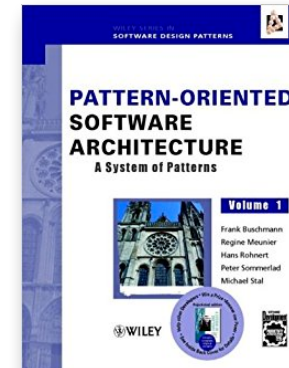
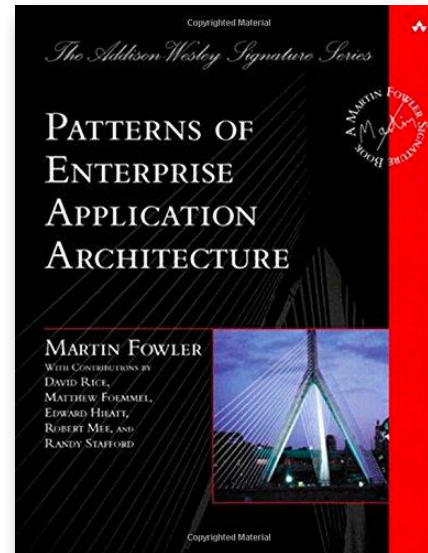
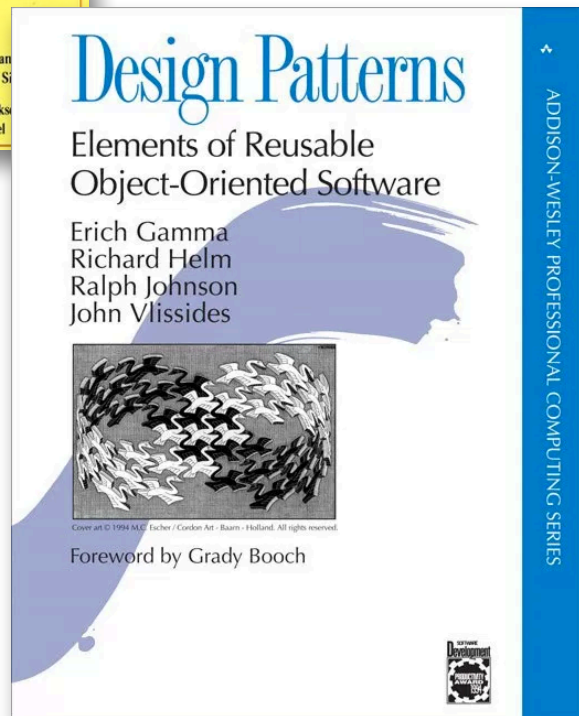
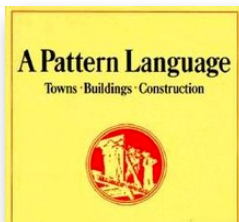
symbiosis

original IP

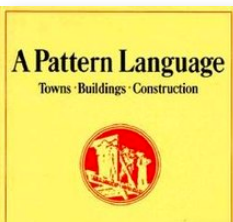
metaphorical
mapping

catalog

catalog



catalog

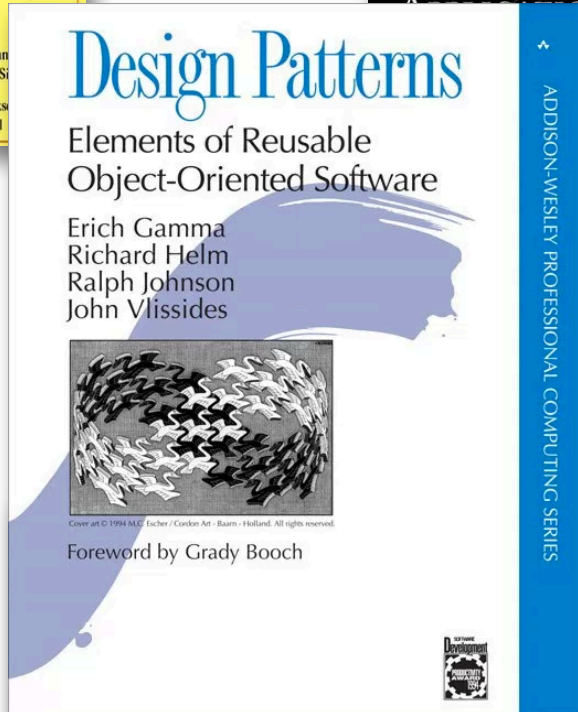


Christopher Alexander
Sara Ishikawa · Murray Silverstein
WITH
Max Jacobson · Ingrid Fiksenberg
Shlomo Angel



PATTERNS OF ENTERPRISE ARCHITECTURE

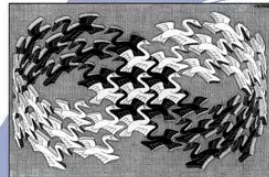
A MARTIN FOWLER SIGNATURE BOOK



Design Patterns

Elements of Reusable
Object-Oriented Software

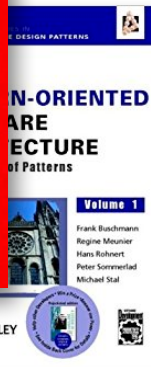
Erich Gamma
Richard Helm
Ralph Johnson
John Vlissides



Cover art © 1994 MAF Escher / Gordon Art - Baarn - Holland. All rights reserved.

Foreword by Grady Booch

ADDISON-WESLEY PROFESSIONAL COMPUTING SERIES

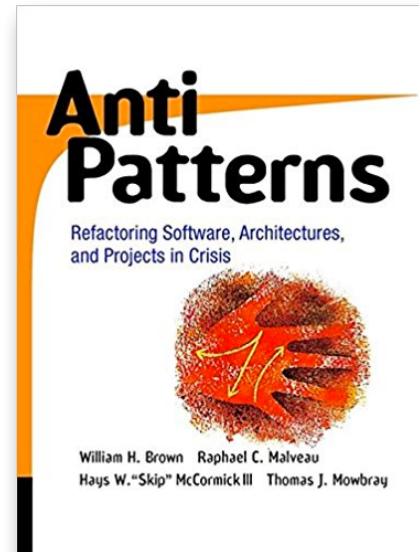


DOMAIN-DRIVEN DESIGN TACKLING COMPLEXITY IN THE HEART OF SOFTWARE

Volume 1

Frank Buschmann
Regine Meunier
Hans Rohnert
Peter Sommerlad
Michael Stal

WILEY

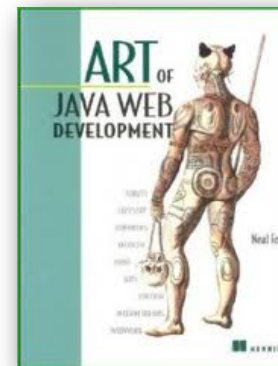


Anti Patterns

Refactoring Software, Architectures,
and Projects in Crisis



William H. Brown · Raphael C. Malveau
Hays W. "Skip" McCormick III · Thomas J. Mowbray

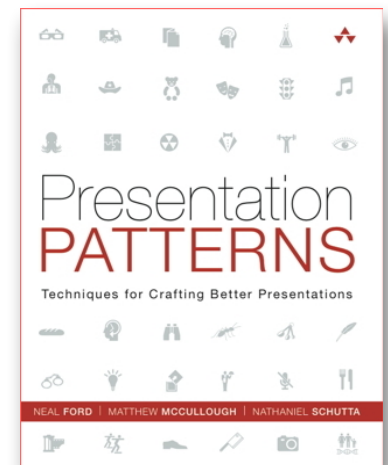


ART OF JAVA WEB DEVELOPMENT



Neal Ford

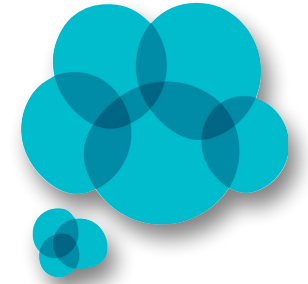
NO STARCH PRESS



Presentation PATTERNS

Techniques for Crafting Better Presentations

NEAL FORD · MATTHEW MCCULLOUGH · NATHANIEL SCHUTTA



cReaTe

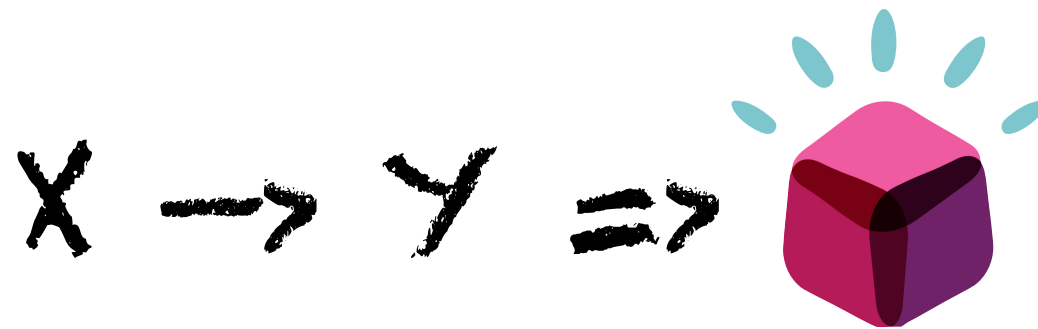
comparison/
survey

symbiosis

original IP

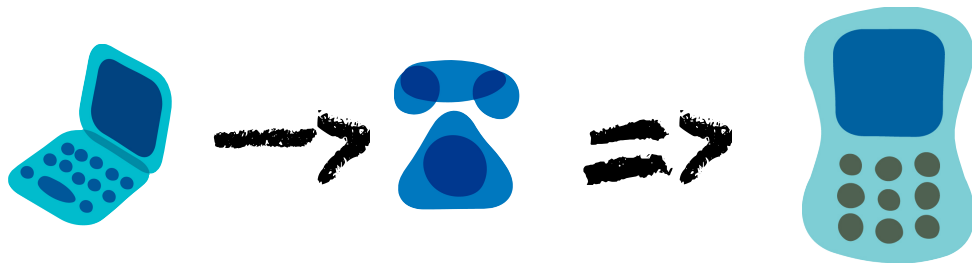
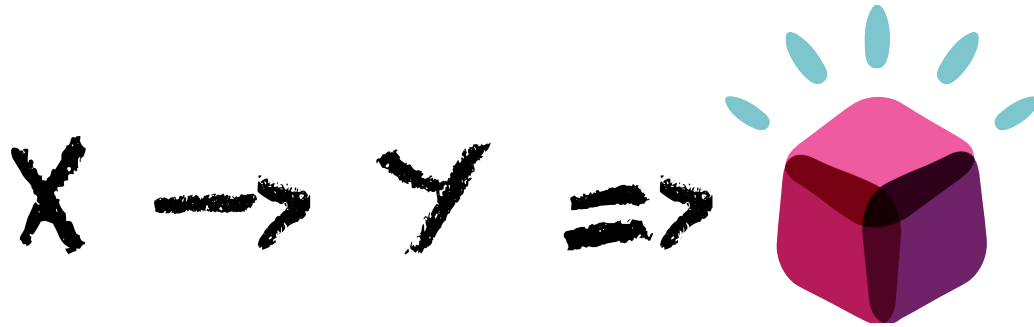
metaphorical
mapping

catalog

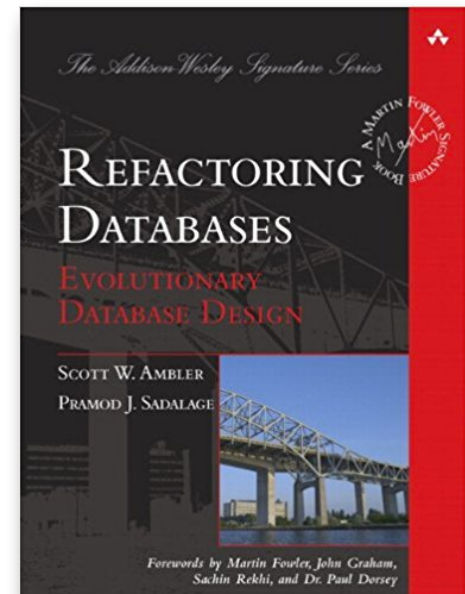
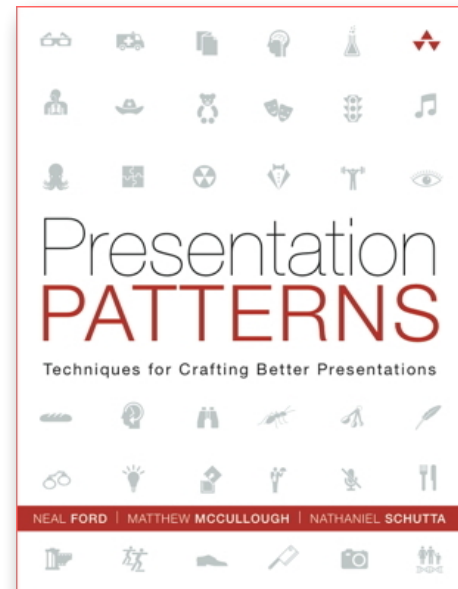


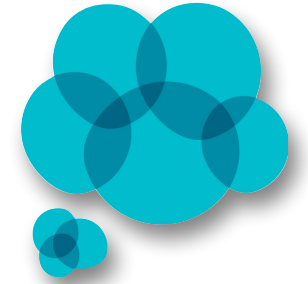
metaphorical
mapping





metaphorical
mapping





cReaTe

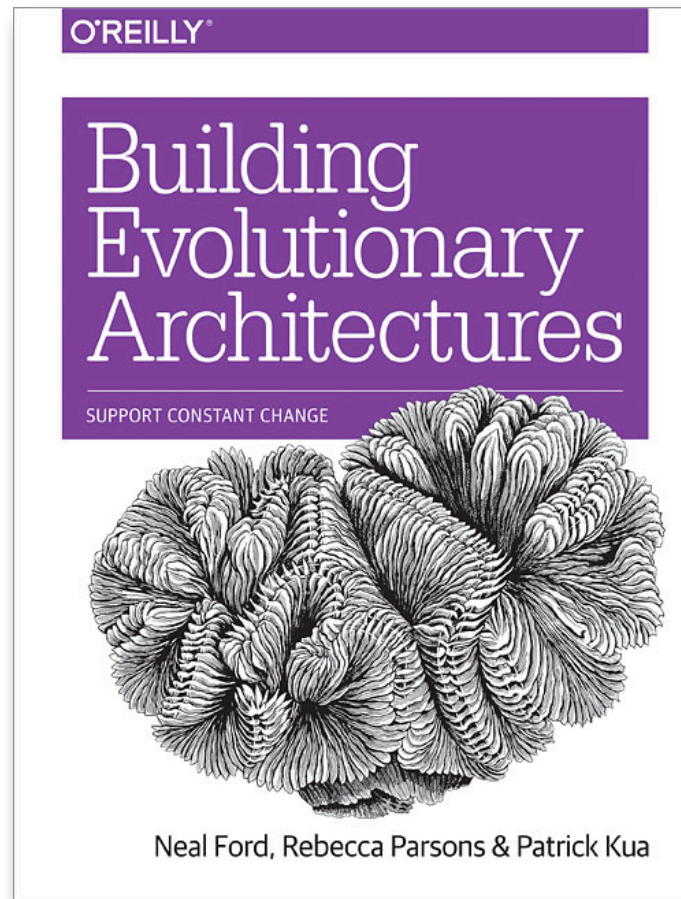
comparison/
survey

symbiosis

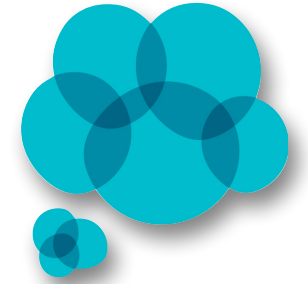
original IP

metaphorical
mapping

catalog



symbiosis between seemingly unrelated subjects



cReaTe

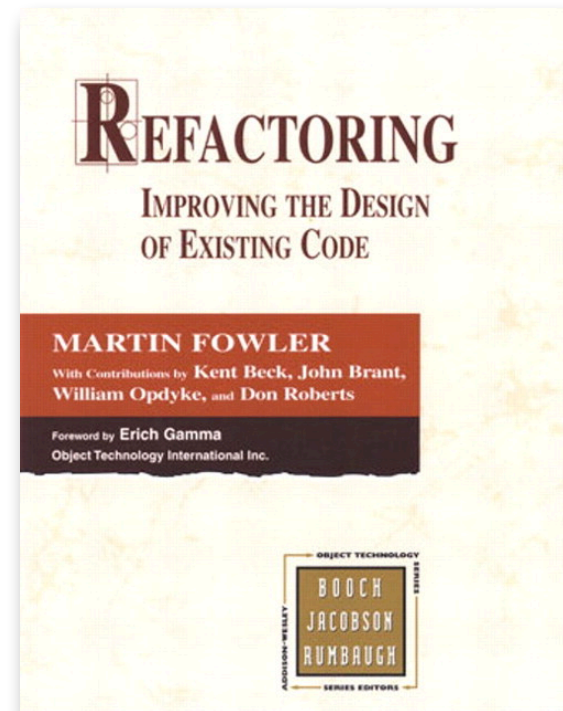
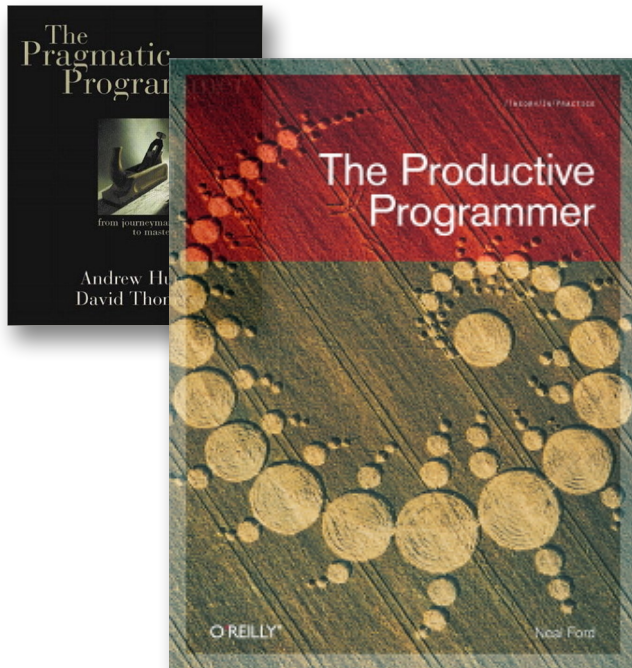
comparison/
survey

symbiosis

original IP

metaphorical
mapping

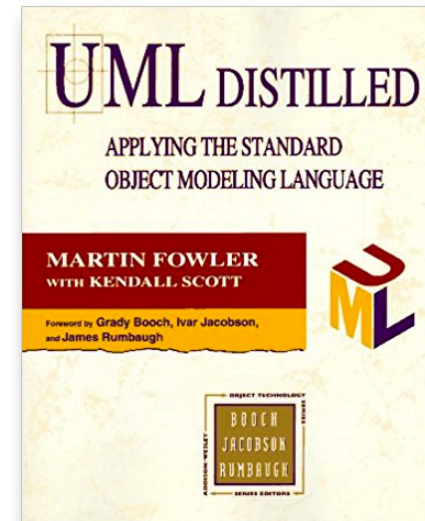
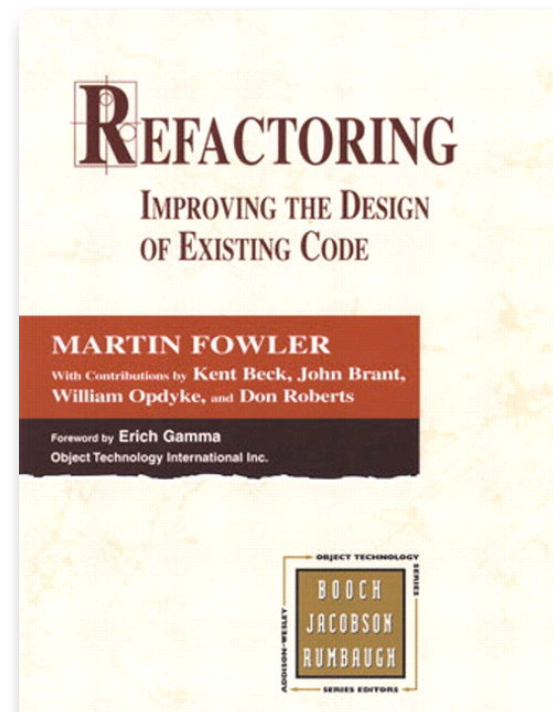
catalog

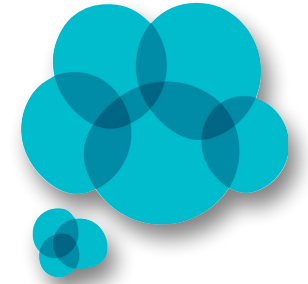


derivative IP



original IP





cReaTe

comparison/
survey

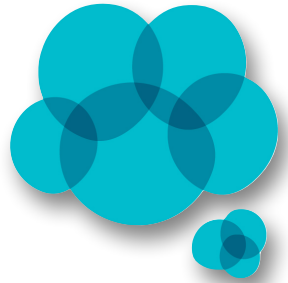
symbiosis

original IP

metaphorical
mapping


catalog

how do ideas evolve?



apply a new axiom

ax·i·om

/ˈaksēəm/ 

noun

a statement or proposition that is regarded as being established, accepted, or self-evidently true.

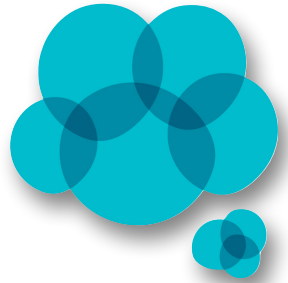
"the **axiom** that supply equals demand"

synonyms: accepted truth, general truth, [dictum](#), [truism](#), [principle](#); [More](#)

- MATHEMATICS

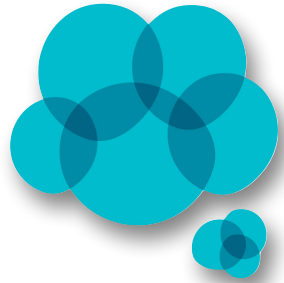
a statement or proposition on which an abstractly defined structure is based.

how do ideas evolve?



mutation

how do ideas evolve?



untangling

complect

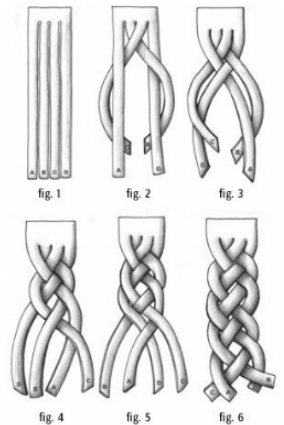
/kəm'plekt/

verb

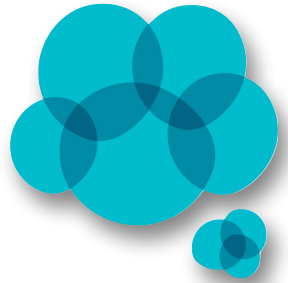
1. **(transitive)** (**archaic**) to interweave or entwine

Word Origin

C16: from Latin *complecti*; see [complex](#)



how do ideas evolve?



oblique strategies

OBLIQUE STRATEGIES

OBLIQUE STRATEGIES

Over one hundred worthwhile dilemmas

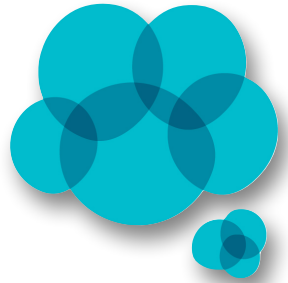
BRIAN ENO and PETER SCHMIDT

Sixth, again slightly revised edition, 2013
© 1975 Brian Eno and Peter Schmidt

Printed May 2013 in an edition of 500

of which this is number: 31

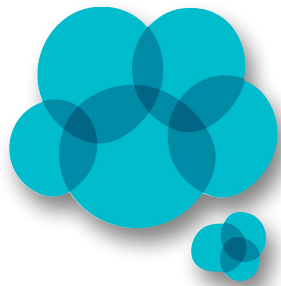
how do ideas evolve?

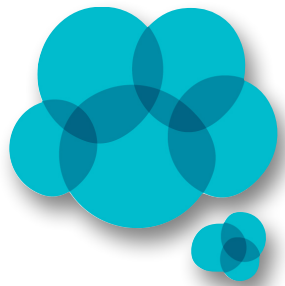


oblique strategies

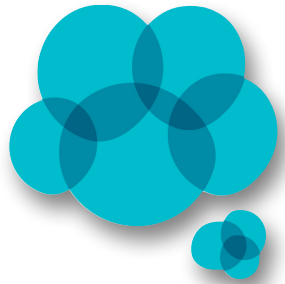


ideas have no species

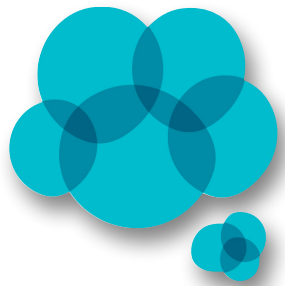




daydreaming
is
important

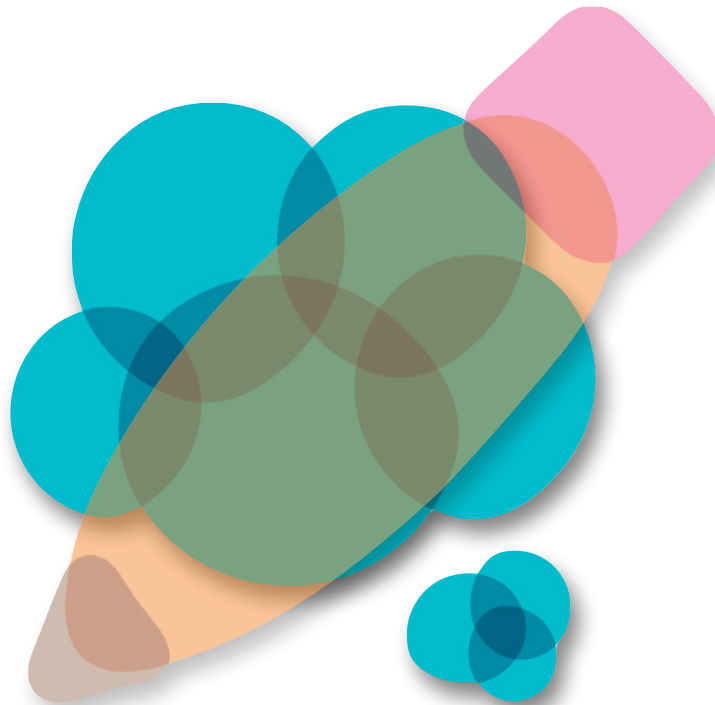


creativity isn't linear

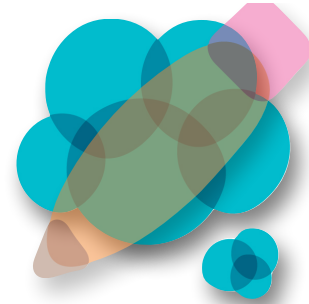


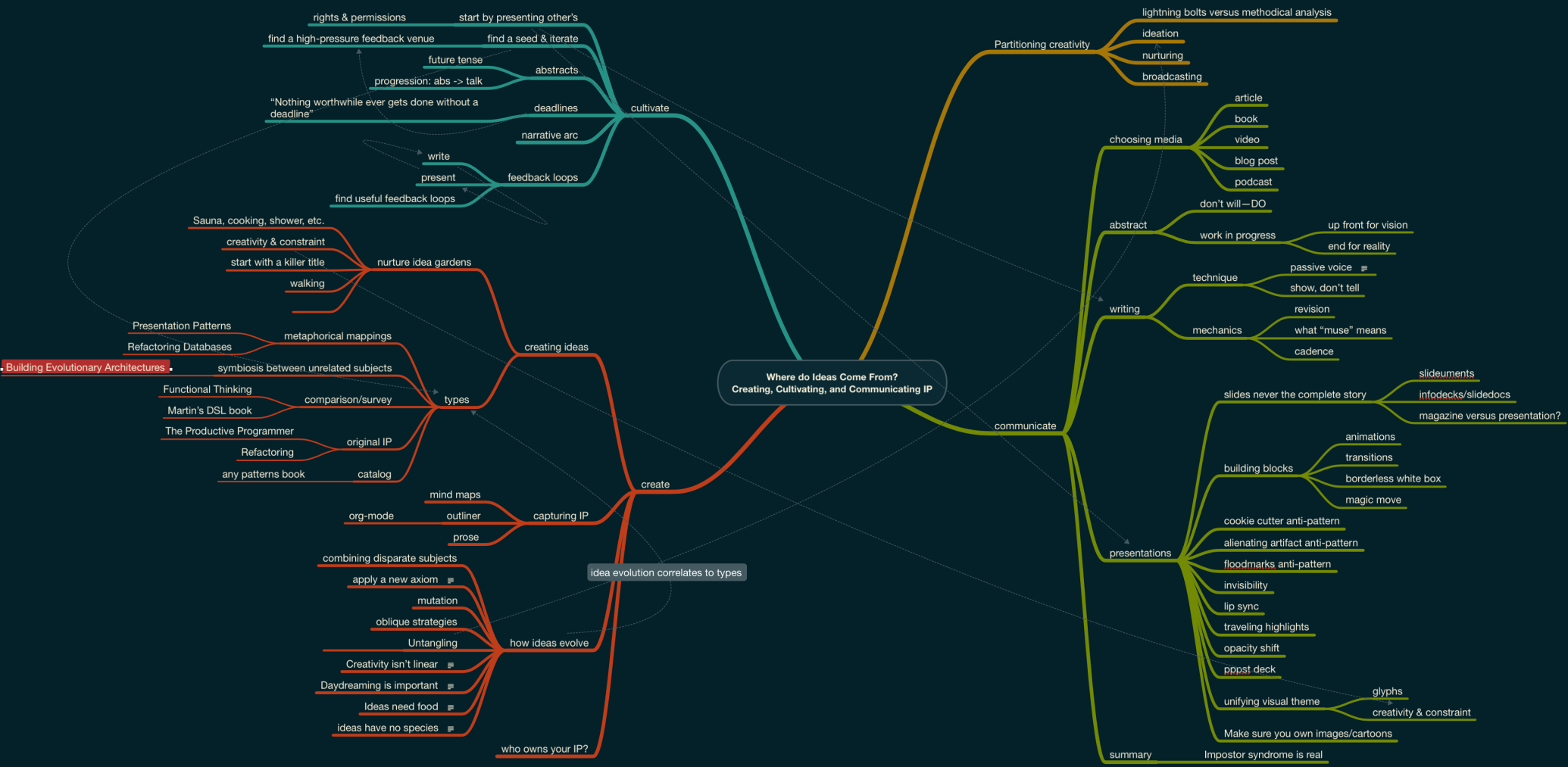
creativity
needs
constraint

capturing IP



capturing IP









```
#+TITLE: Example .org file
#+AUTHOR: Bastien
#+DATE: 2012-09-08 sam.
```

* Make a new website for orgmode

** TODO Make screenshots

SCHEDULED: <2012-09-07 ven.>

** NEXT [#A] Add them to the website

SCHEDULED: <2012-09-08 sam.>

** NEXT Publish the new website

DEADLINE: <2012-09-09 dim.>

* Plain-text tables and spreadsheet

N	N^2	N^3	N^4	sqrt(n)	sqrt(N)
1	1	1	1	1	1
2	4	8	16	1.4142	1.1892
3	9	27	81	1.7321	1.3161

#+TBLFM: \$2=\$1^2::\$3=\$1^3::\$4=\$1^4::\$5=sqrt(\$1)::6=sqrt(\$5)

website.org All L1 [0] [(Org +5 Fill Narrow)] sam. sept. 8 08:10 0.13

Column Formulas

\$2 = \$1^2

\$3 = \$1^3

\$4 = \$1^4

\$5 = sqrt(\$1)

\$6 = sqrt(sqrt(\$1))

U:*** *Edit Formulas* All L5 [0] [(Fundamental +5)] sam. sept. 8 08:10 0.13

* Plain-text tables and spreadsheet...

* Working with source code

website.org Top L27 [0] [(Org +5 Fill)] sam. sept. 8 08:15 0.30
(No changes need to be saved)

* Plain-text tables and spreadsheet

N	N^2	N^3	N^4	sqrt(n)	sqrt(N)
/	<		>	<	>
1	1	1	1	1	1
2	4	8	16	1.4142	1.1892
3	9	27	81	1.7321	1.3161

#+TBLFM: \$2=\$1^2::\$3=\$1^3::\$4=\$1^4::\$5=sqrt(\$1):: \$6=sqrt(sqrt(\$1))

--- website.org All L1 [0] [(Org +5 Fill Narrow)] sam. sept. 8 08:10 0.13

Column Formulas

\$2 = \$1^2

\$3 = \$1^3

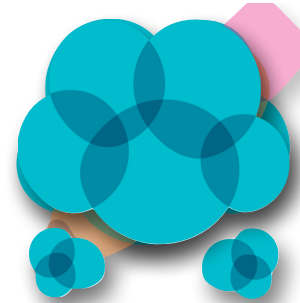
\$4 = \$1^4

\$5 = sqrt(\$1)

\$6 = sqrt(sqrt(\$1))

U:*** *Edit Formulas* All L5 [0] [(Fundamental +5)] sam. sept. 8 08:10 0.13

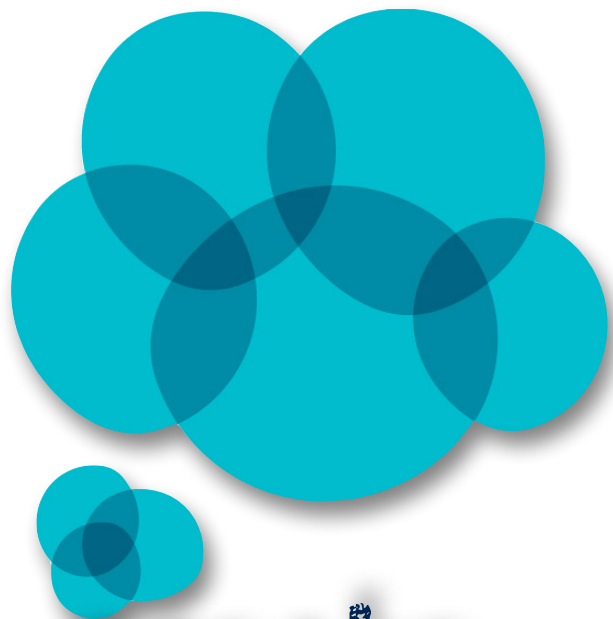
capturing IP



prose

AsciiDoc
Text based document generation
<http://asciidoc.org/>

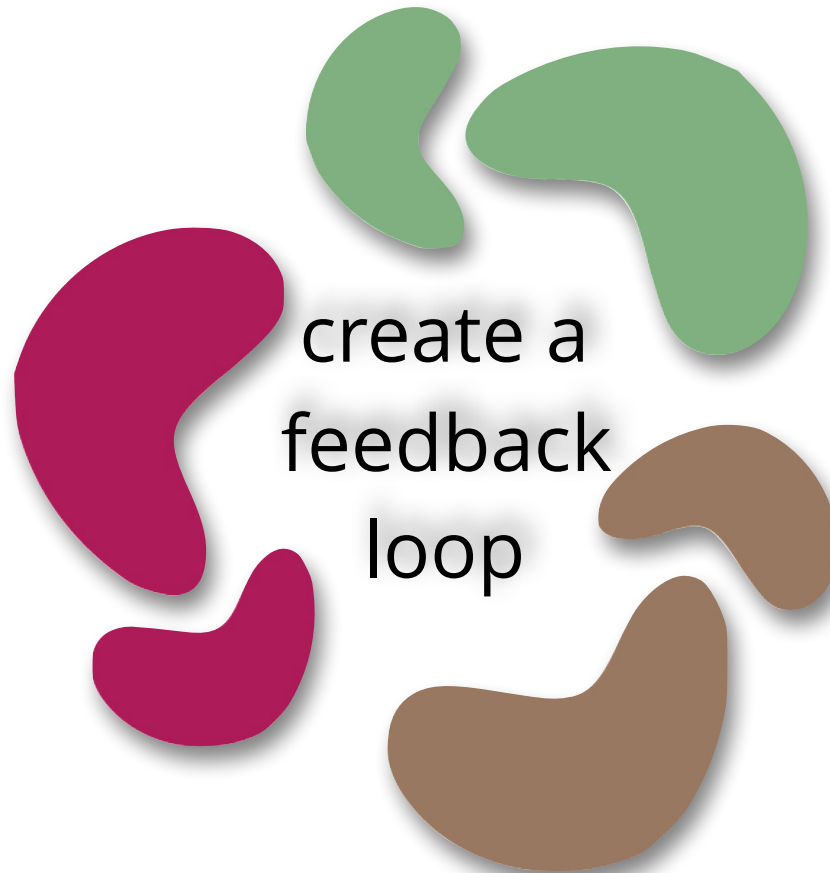
cultivate



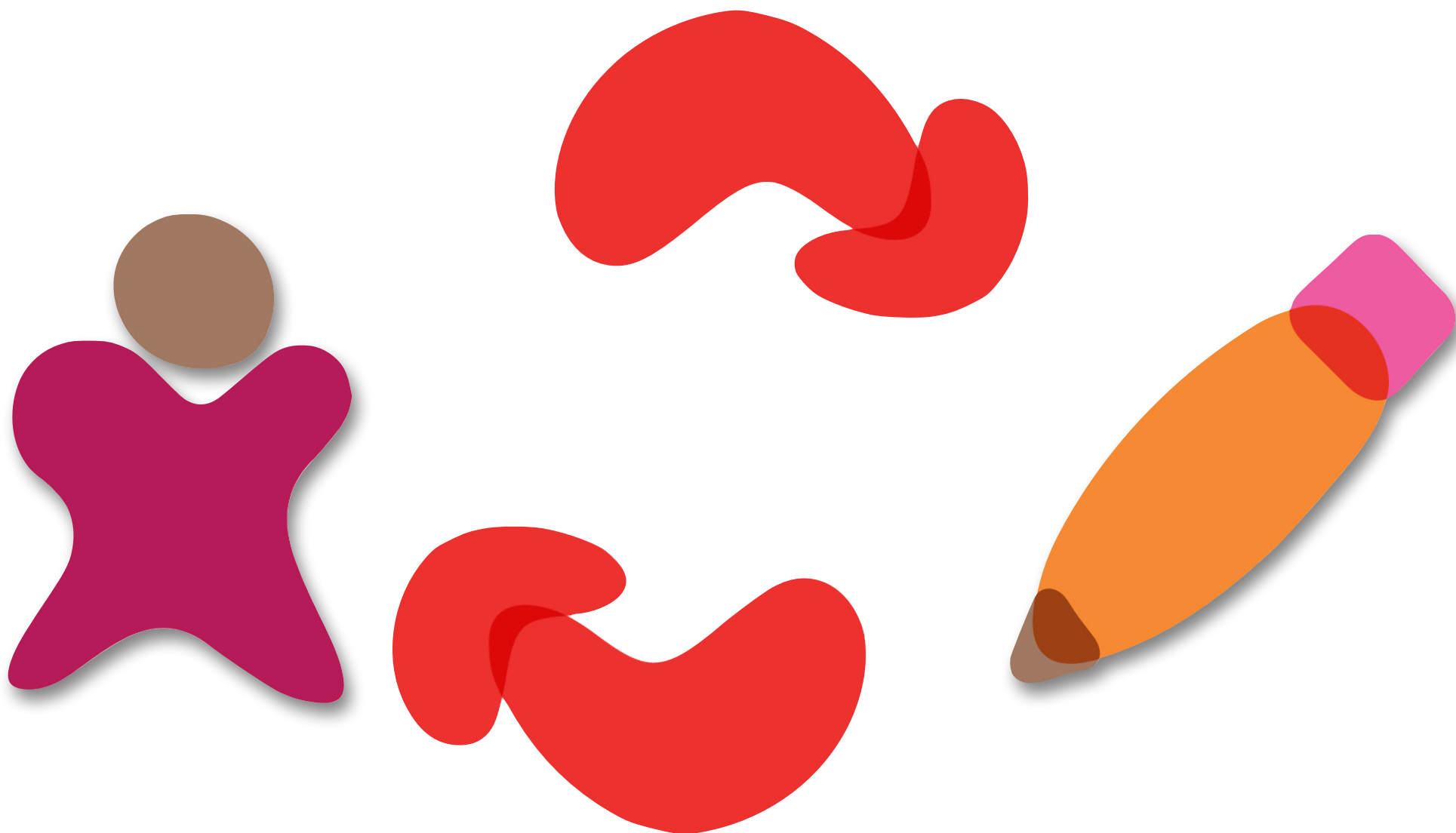
create



communicate





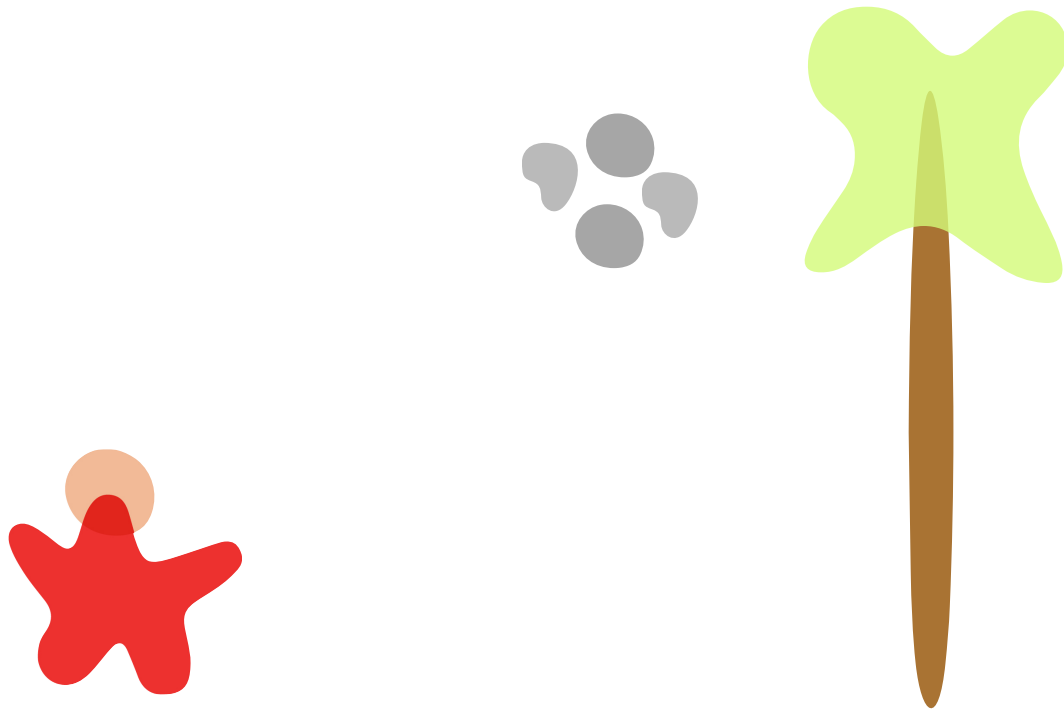






narrative arc





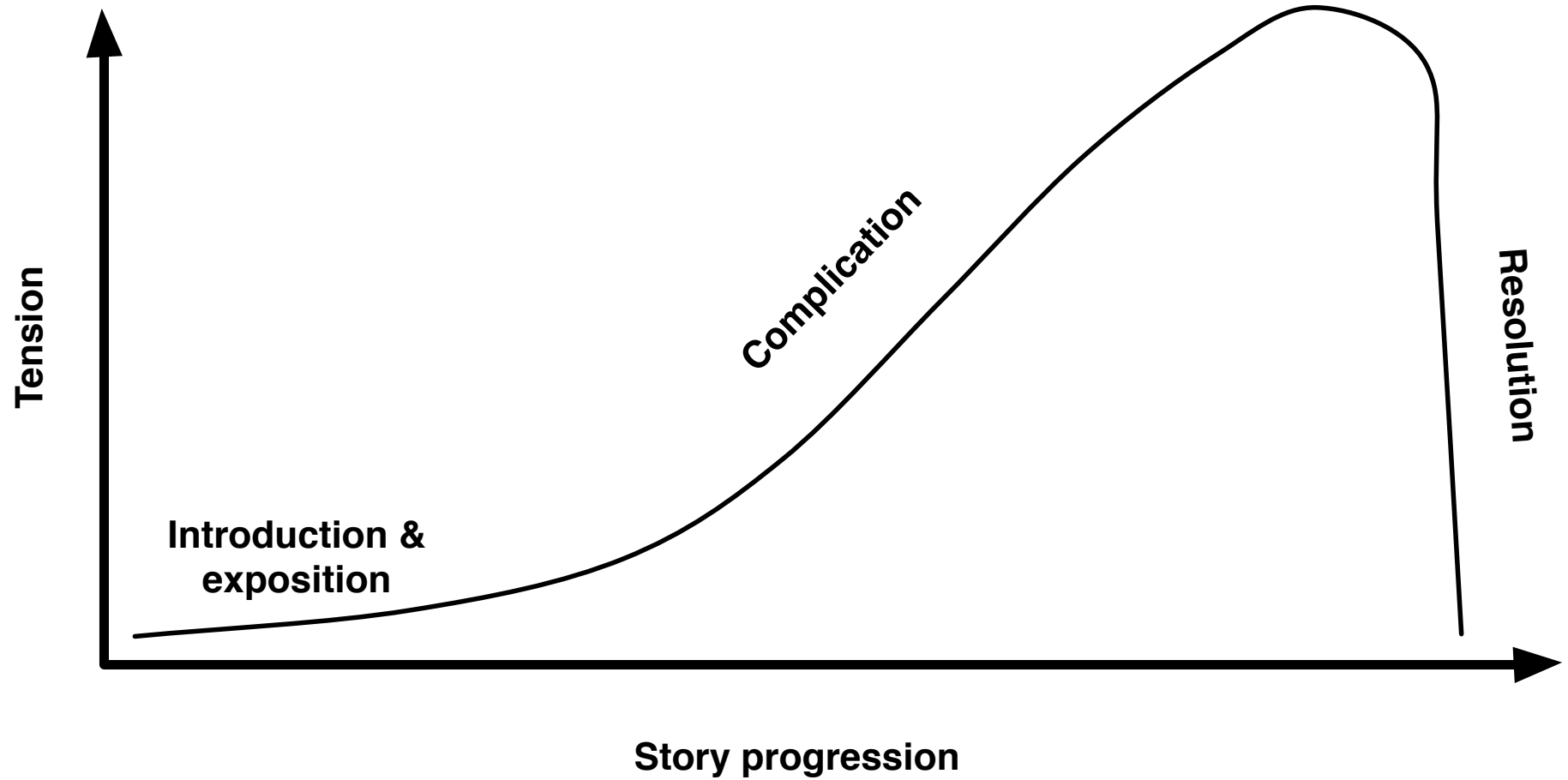
"Get your protagonist up a tree. Throw rocks at him.
Then get him down."

—Syd Fields

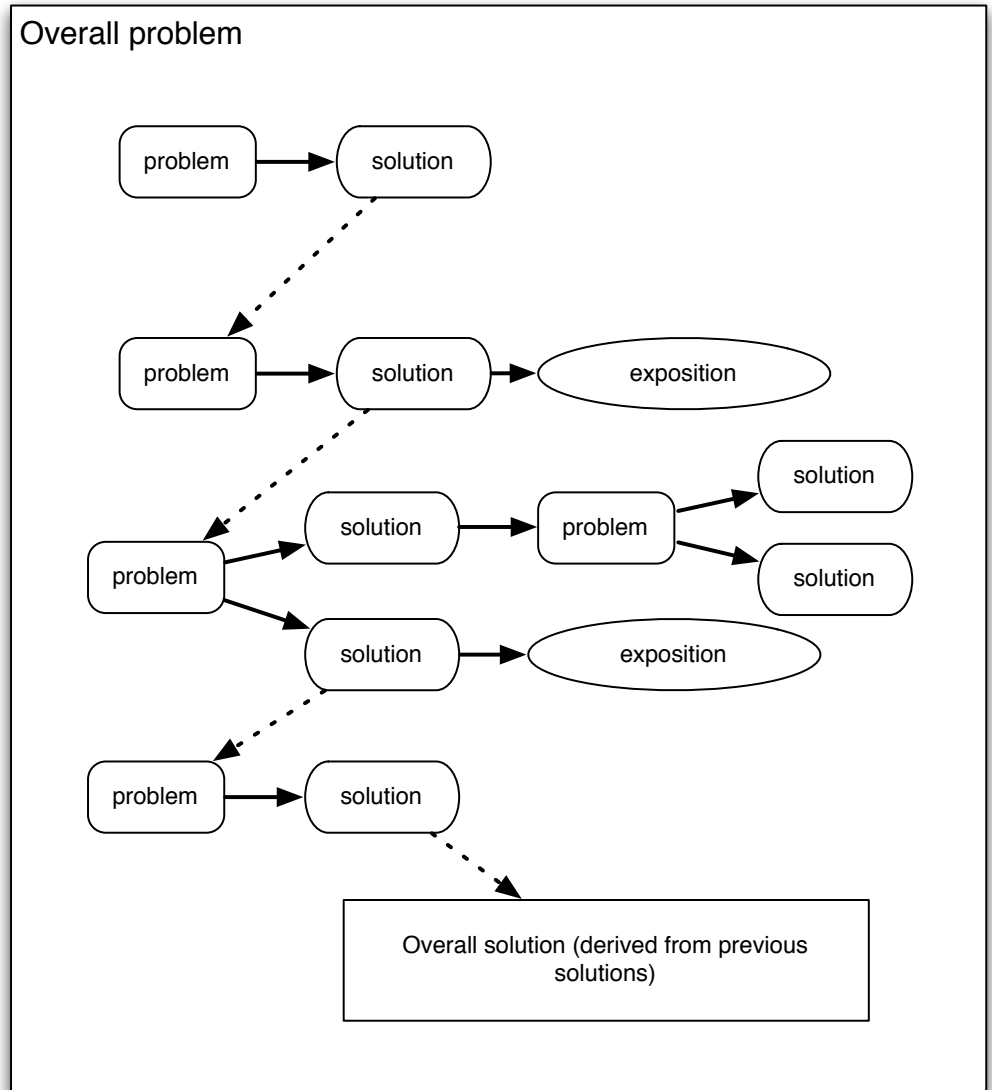


narrative arc

Climax



Technical Narrative?



tdt

test-driven design

housekeeping

what i cover:

why watch it fail?

design benefits of tdd

design benefits of tdd

Overall Problem

Exposition

perfect number:

Problem

Solution

the simplest thing!

first test

next test

Problem

static tends to propagate

take 2

should we keep this test?

Problem

Solution

keep trivial tests as canaries

next test

yuck!

problems:

Problem

Solution

composed method

refactor

more tests

boundary conditions

Problem

Solution

next test: sumOfFactors()

Problem

Solution

rethink the abstraction:

Problem

Solution

next: calculateFactors()

Problem

Solution

refactor

more tests

Overall Solution

Overall Solution

Problem

Solution

Classifier

Exposition

complete tests

helpers

Exposition

cyclomatic complexity

Problem

Solution

Problem

Solution

how id's help

dry secret: private is ignored!

Solution

documentation

Problem

Solution

testing what's hard to test

Problem

Solution

add private methods!

mock

Solution

mocking

with

java-like syntax

Solution

idiomatic

Overall Solution Summary

? 's

123

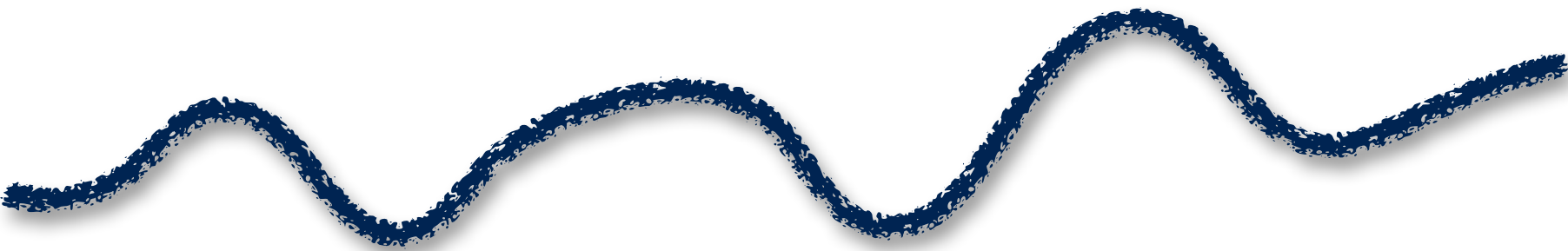


don't worry too much about the end when you start



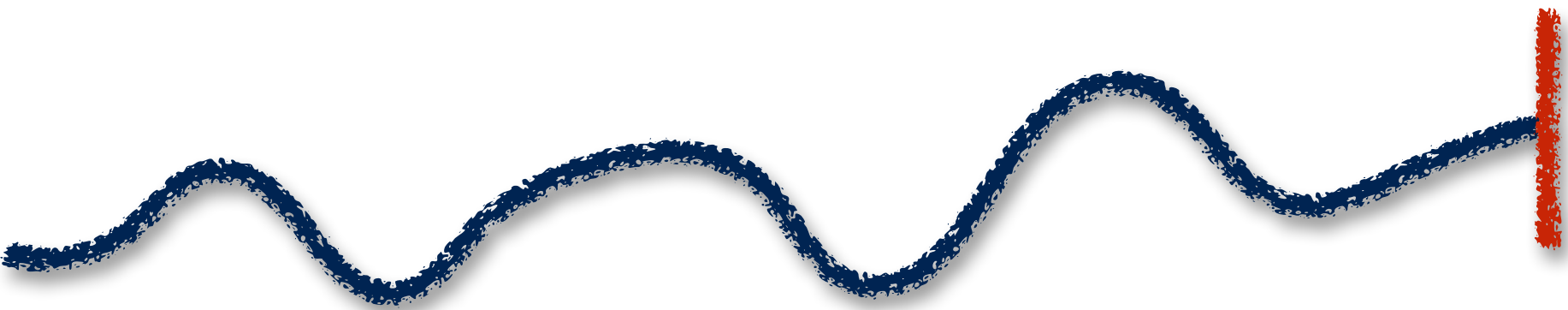


don't worry too much about the end when you start





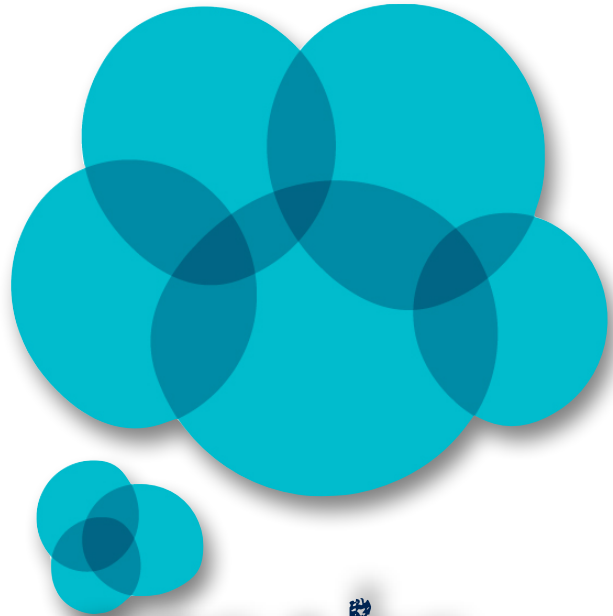
nothing useful gets done
without a deadline!



find your creative time of the day
(and defend it)



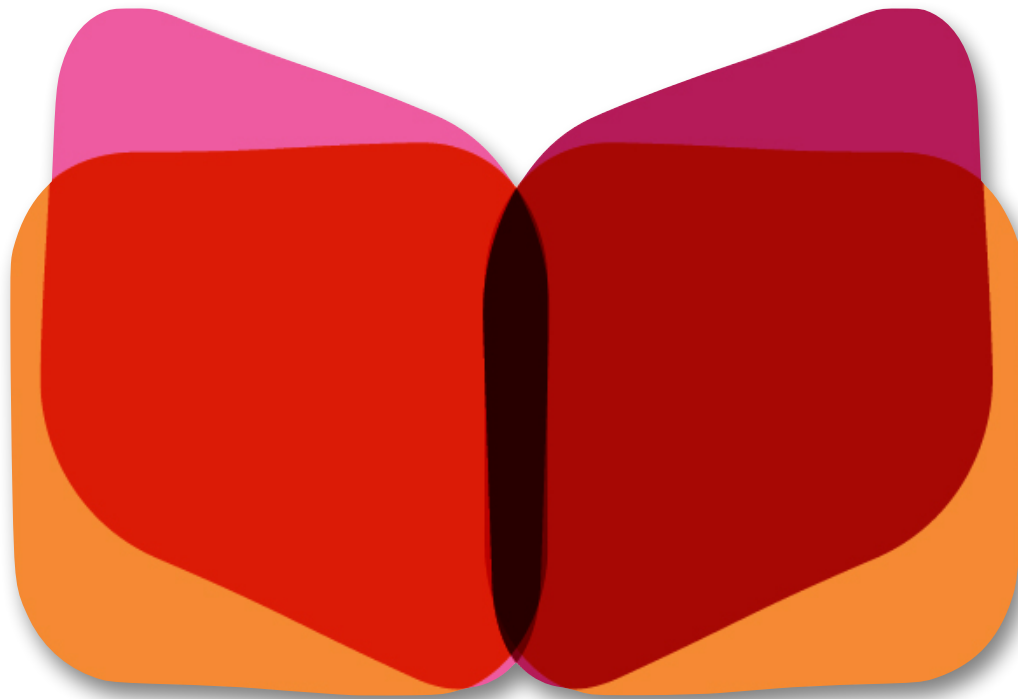
cultivate



create

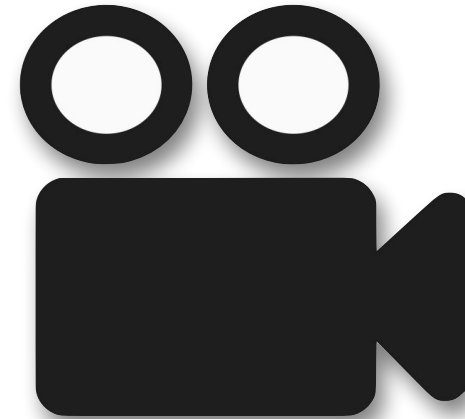
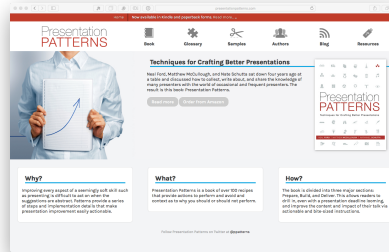
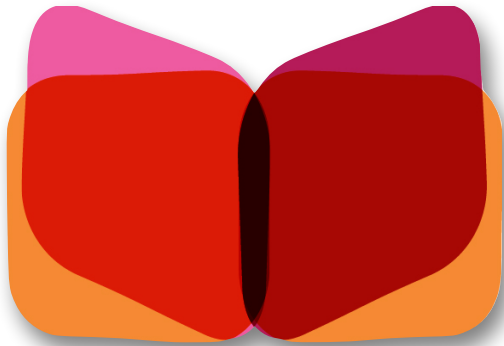


communicate

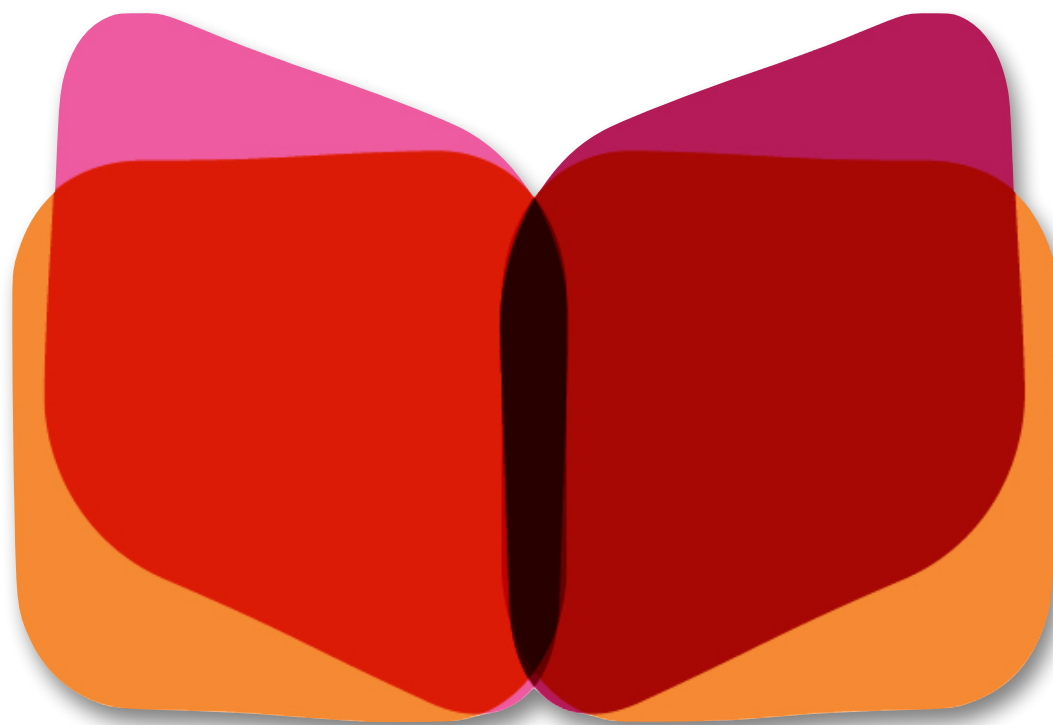


communicate

Appropriate Media

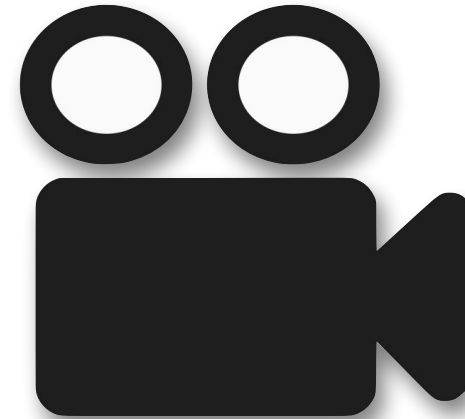
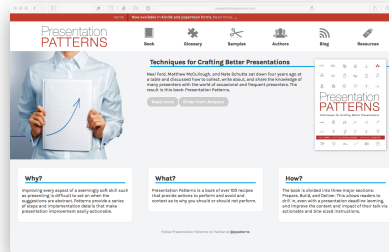
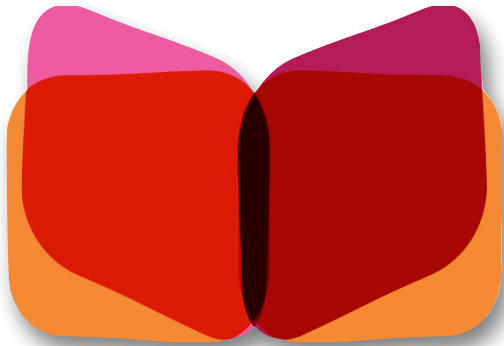



communicate

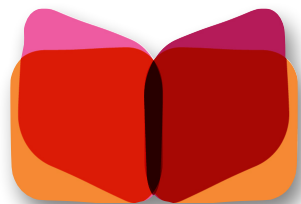
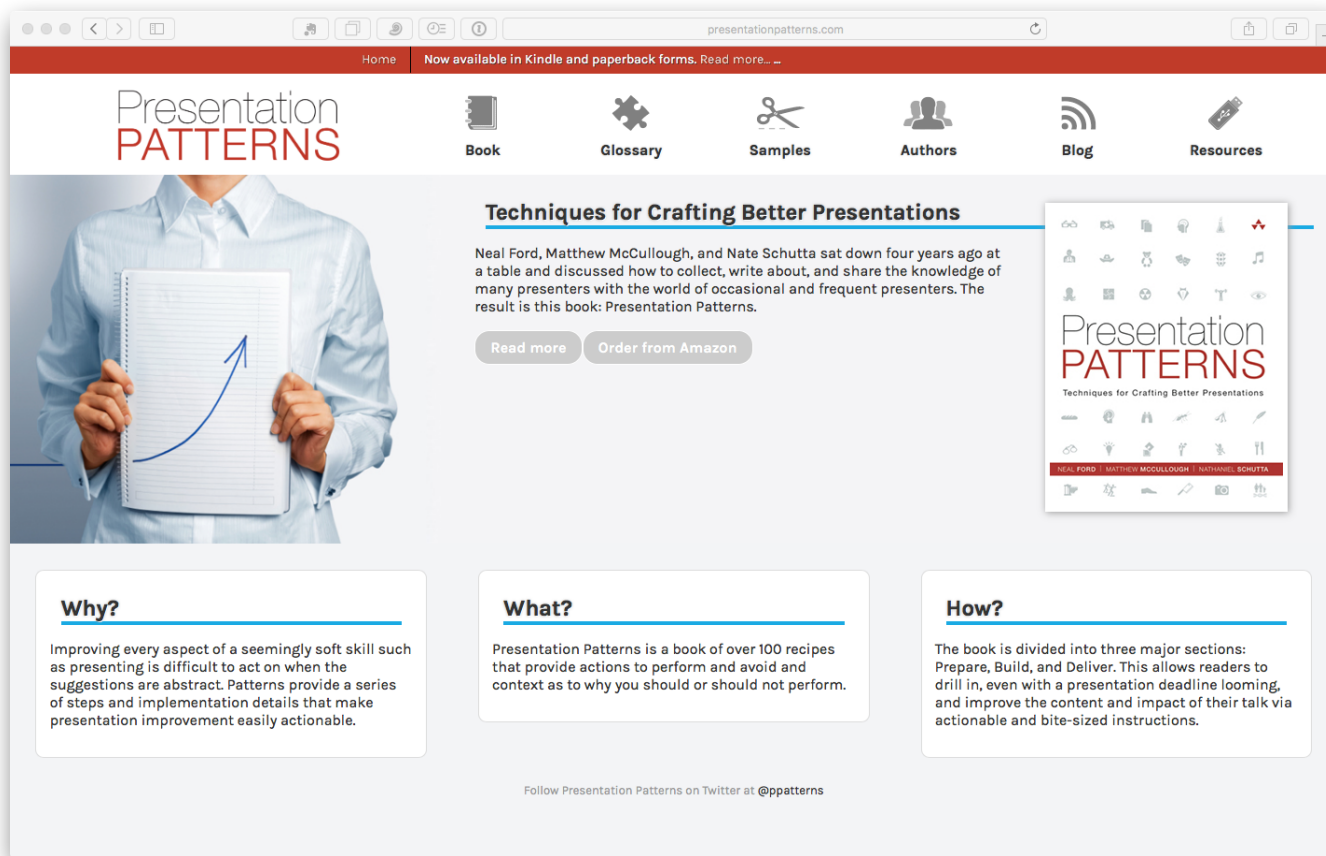


Books

Appropriate Media

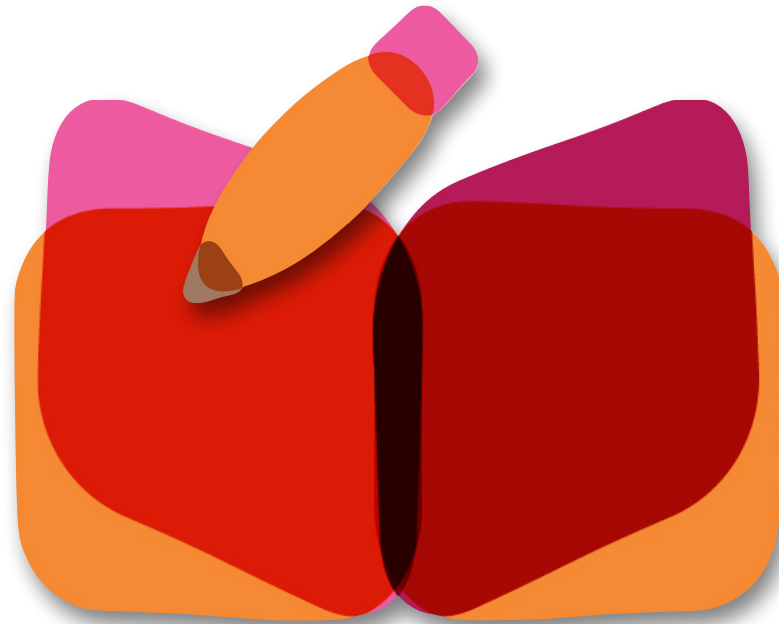



communicate



Article/Blog series

Writing





Passive Voice

Passive voice occurs when you make the object of an action into the subject of a sentence.

Why was the road crossed ?



Passive Voice

The metropolis has been scorched by the dragon's fiery breath.

The dragon scorched the metropolis with his fiery breath.

When her house was invaded, Penelope had to think of ways to delay her remarriage.

After suitors invaded her house, Penelope had to think of ways to delay her remarriage.



Passive Voice Myths

1. Use of the passive voice constitutes a grammatical error.
2. Any use of "to be" (in any form) constitutes the passive voice.
3. The passive voice always avoids the first person.
4. You should never use the passive voice.
5. I can rely on my grammar checker to catch the passive voice.



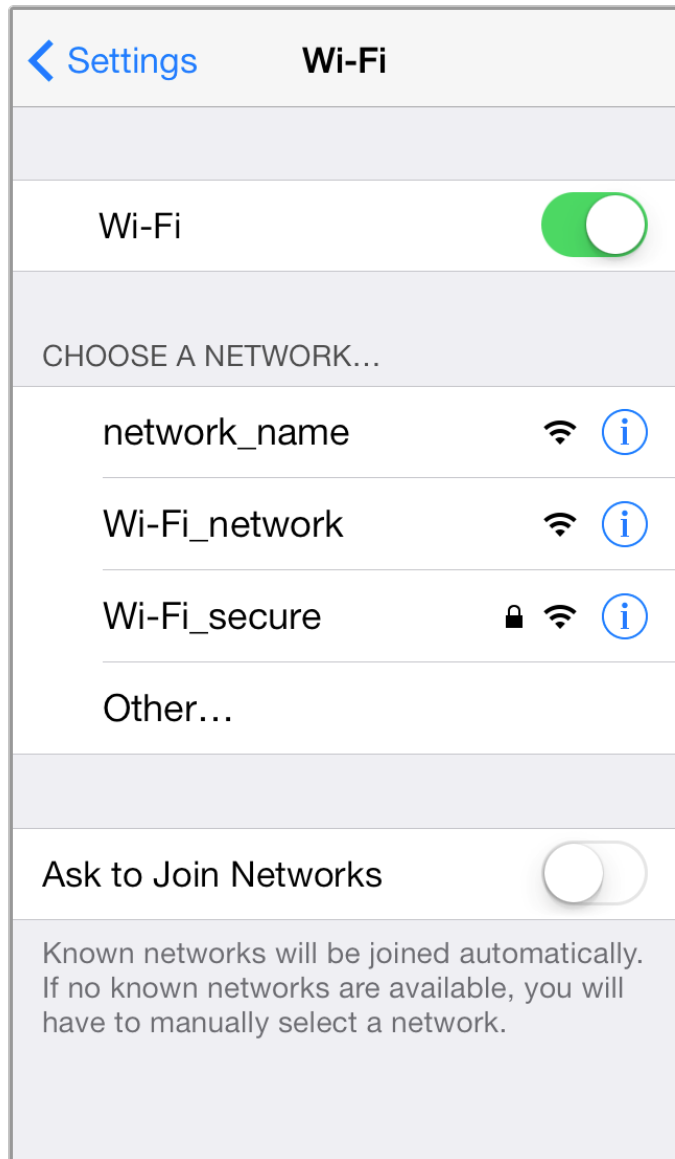
Passive Voice

Mistakes were made.

The Exxon Company accepts that a few gallons might have been spilled.

"Swindles & Perversions"

some people use the passive voice to avoid mentioning responsibility for certain actions



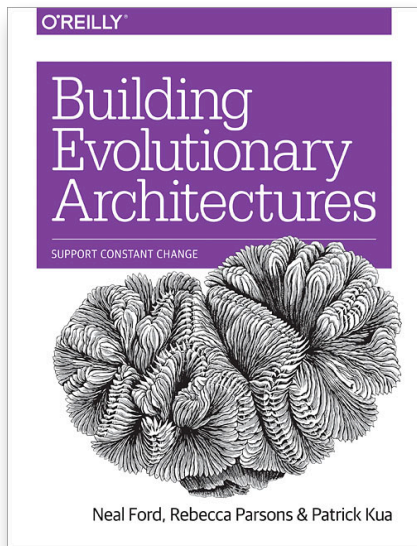
Your phone will join known networks automatically.

Your phone automatically joins known networks.

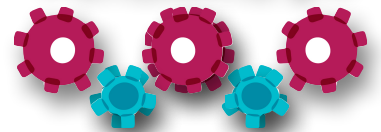
If no known networks are available, you must manually select a network.

Writing

show, don't tell

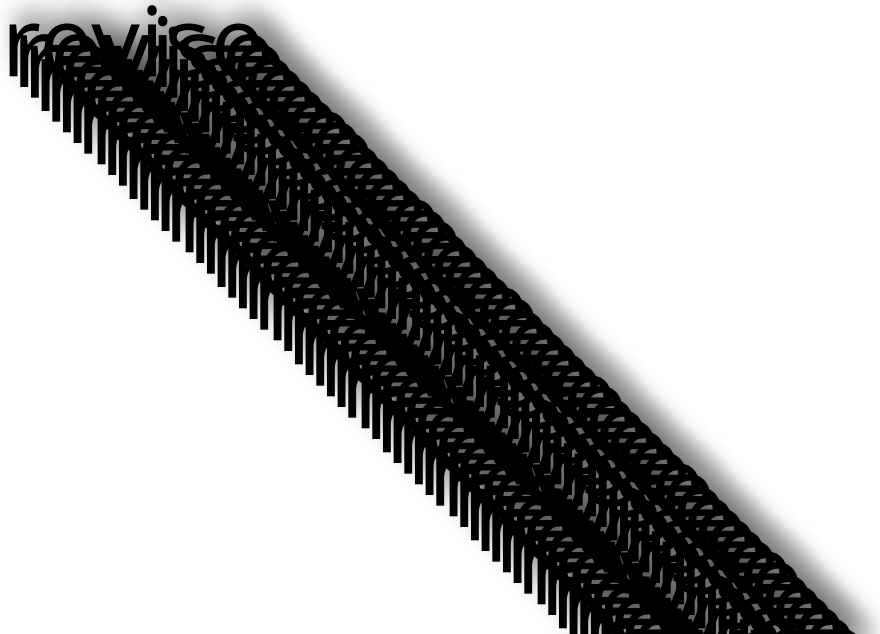


Penultima↑e





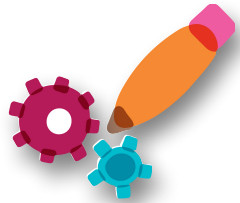
Writing Mechanics



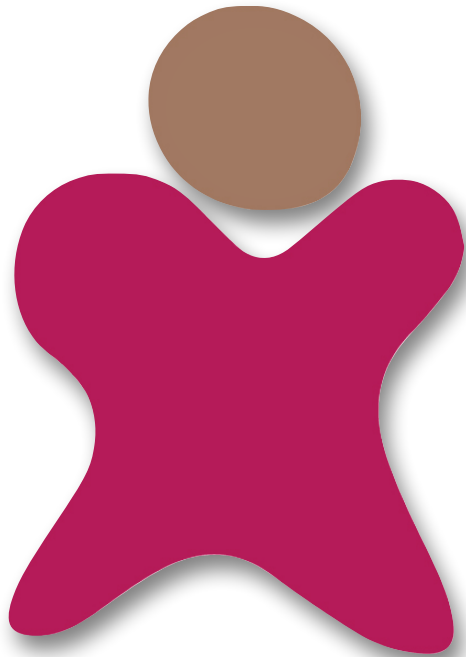


Writing Mechanics





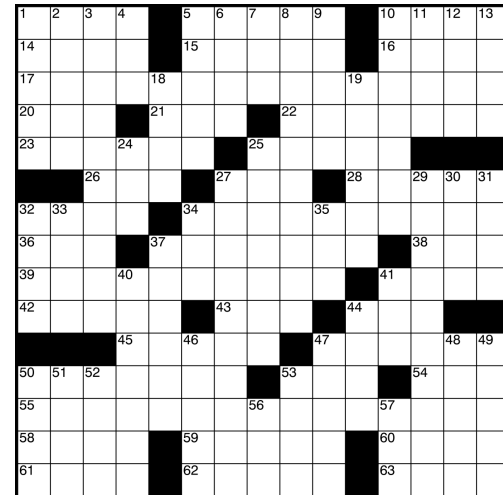
Writing Mechanics



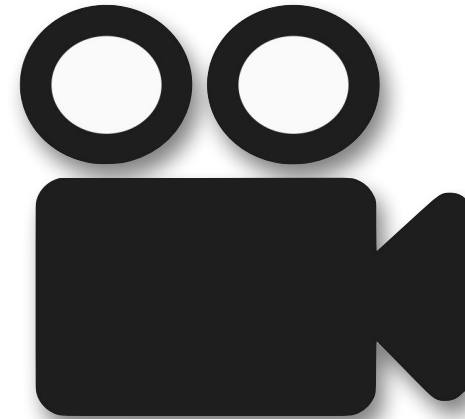
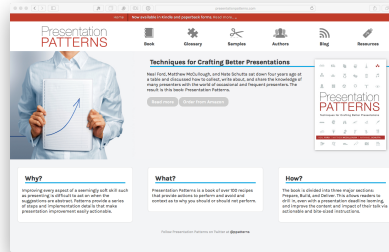
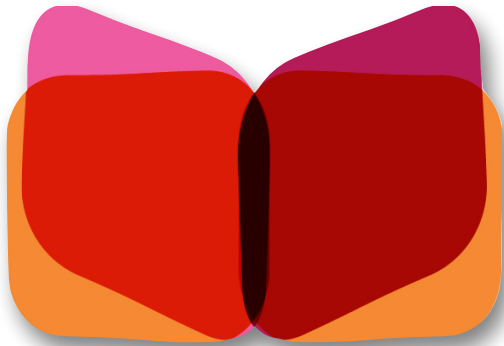
“the muse”



writing benefits from regularity



Appropriate Media

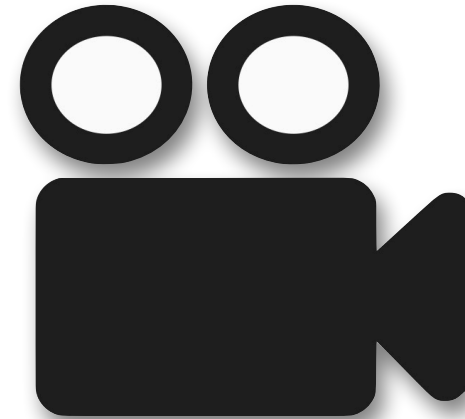
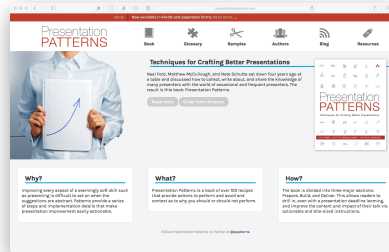
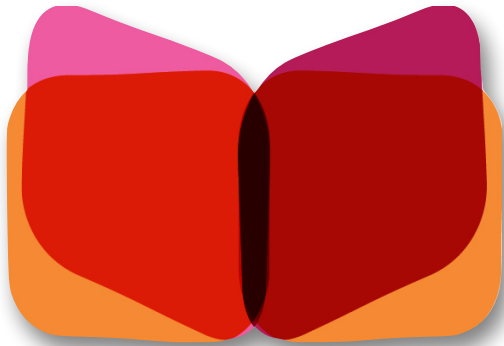


 
communicate

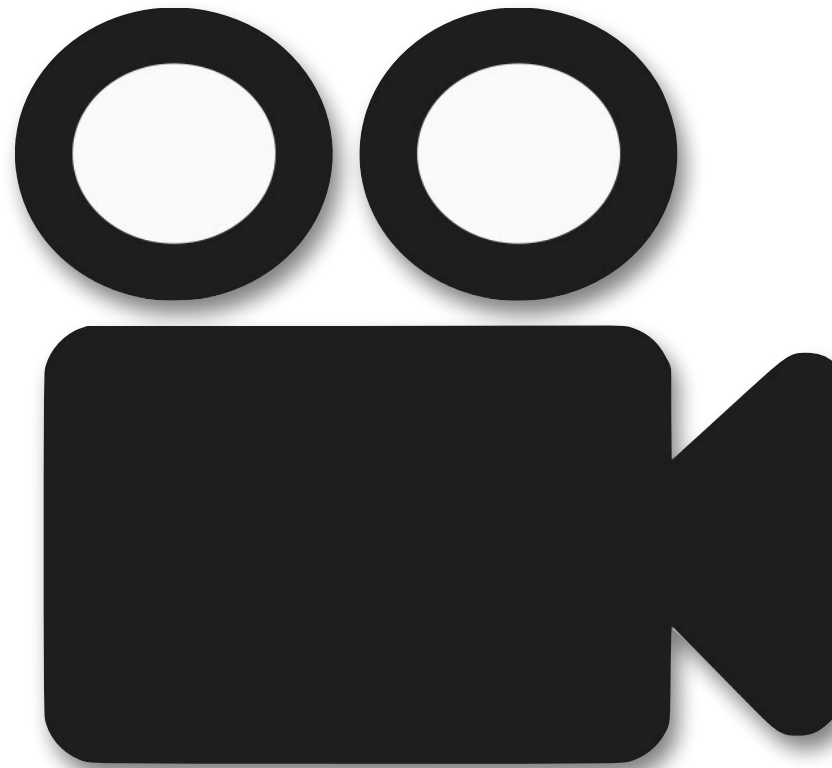


Podcast

Appropriate Media

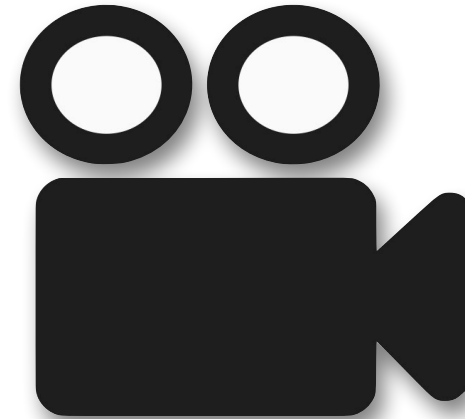
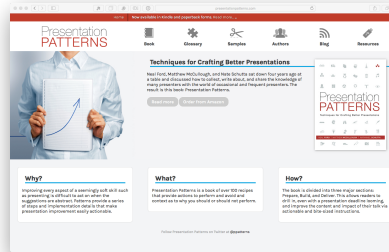
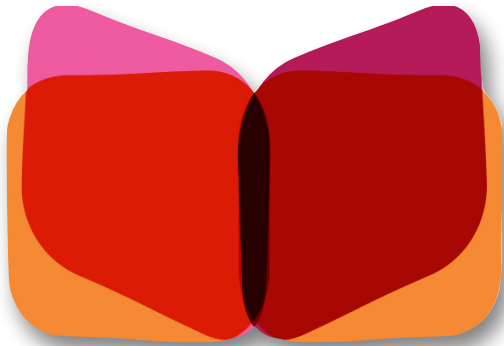



communicate



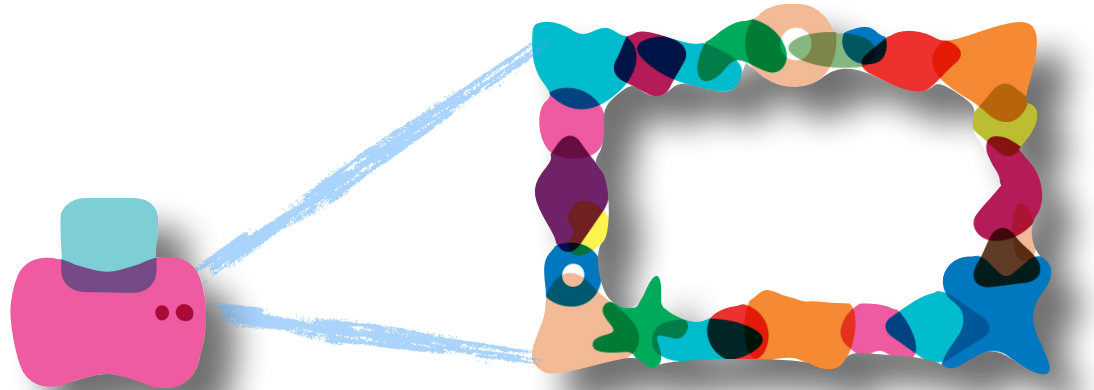
(streaming) Video

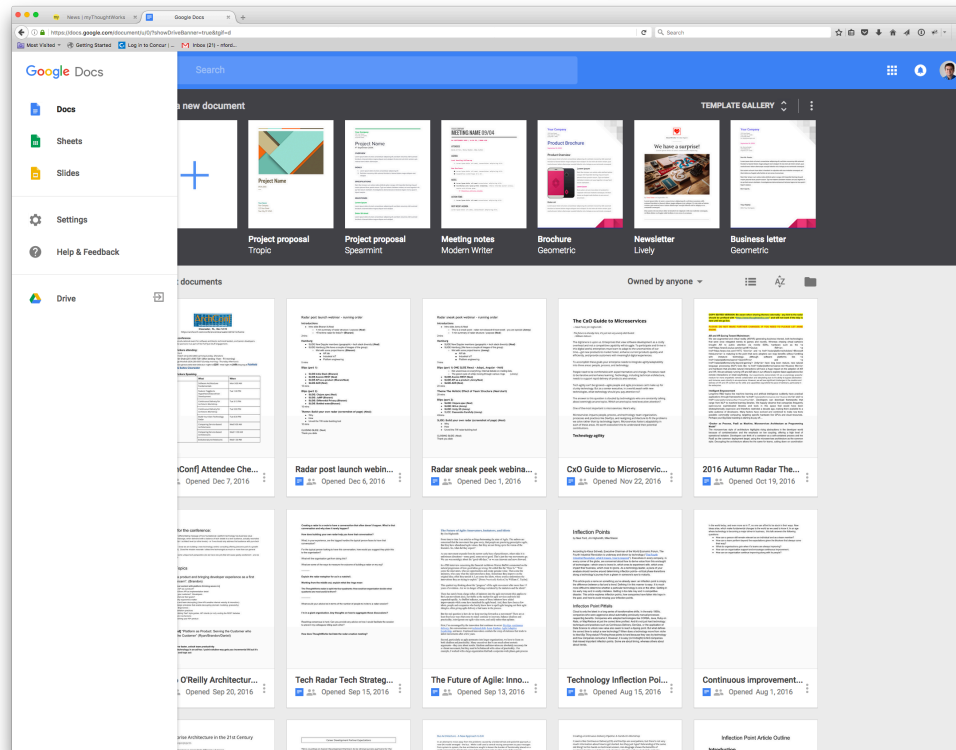
Appropriate Media

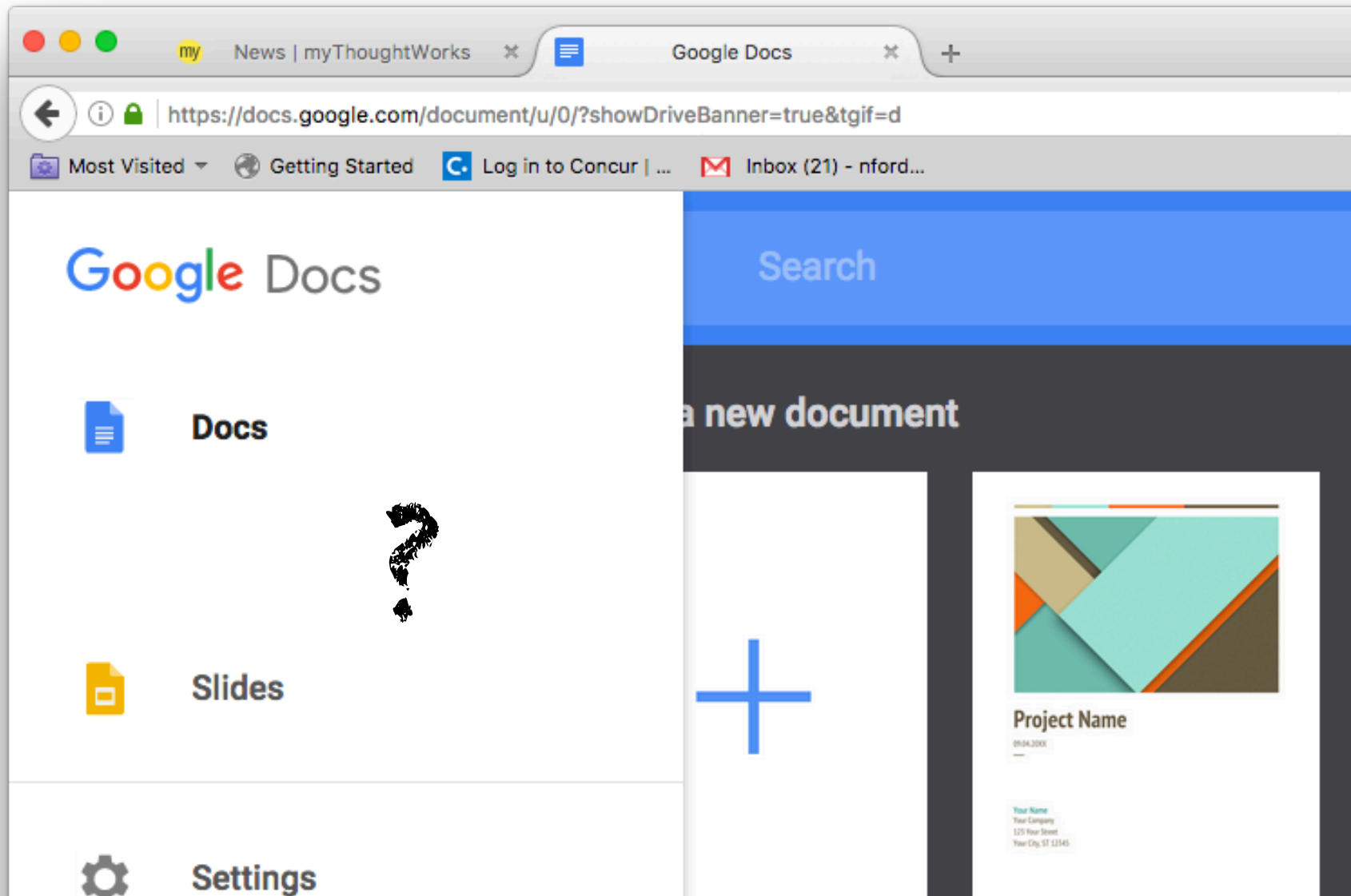


 
communicate

Presentations











if you aren't manipulating time,
you're in the wrong medium

Presentation



Presenter controls
exposition rate.

Slideument

Infodeck



Reader controls
exposition rate.

Testing Strategies in a Microservice Architecture

ThoughtWorks®

There has been a shift in service based architectures over the last few years towards smaller, more focussed "micro" services. There are many benefits with this approach such as the ability to independently deploy, scale and maintain each component and parallelize development across multiple teams. However, once these additional network partitions have been introduced, the testing strategies that applied for monolithic in process applications need to be reconsidered.

Here, we plan to discuss a number of approaches for managing the additional testing complexity of multiple independently deployable components as well as how to have tests and the application remain correct despite having multiple teams each acting as guardians for different services.

18 November 2014



Toby Clemson is a developer at ThoughtWorks with a passion for building large scale distributed business systems. He has worked on projects in four continents and is currently based in New York.

My thanks to **Martin Fowler** for his continued support in compiling this infodeck. Thanks also to **Danilo Sato, Dan Coffman, Steven Lowe, Chris Ford, Mark Taylor, Praful Todkar, Sam Newman** and **Marcos Matos** for their feedback and contributions.

Hints for using this deck

<http://martinfowler.com/articles/microservice-testing>



if you aren't manipulating time,
you're in the wrong medium



writing abstracts

you don't have to have a full talk
to have an abstract



writing abstracts



“don’t will...DO!”



writing abstracts

submitting abstracts —> deadlines



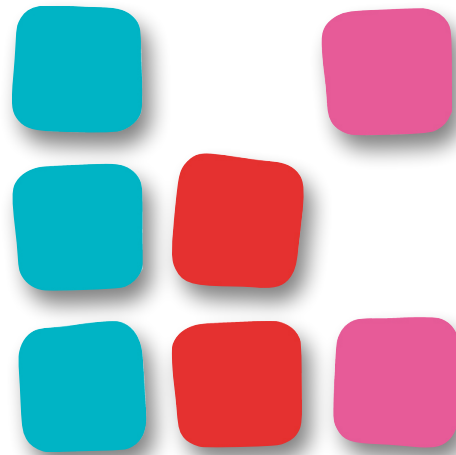
2 versions of writing^abstracts

write at the beginning for direction

rewrite at the end

(when you know what it's actually about)

building blocks



building blocks

transition

movement between slides

animation

movement on slides



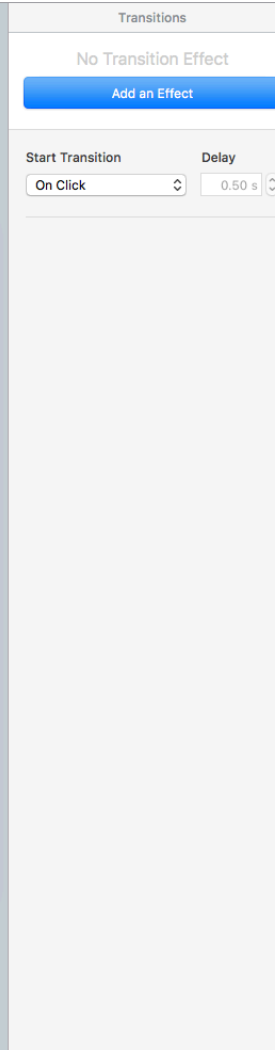
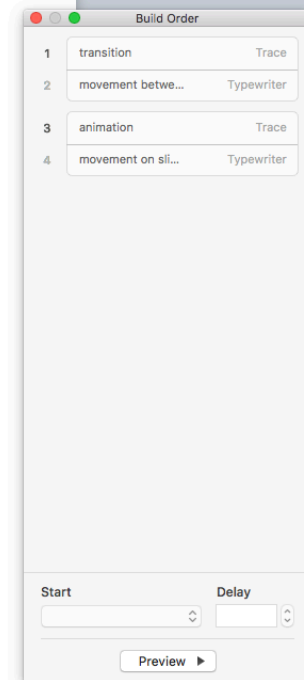
building blocks

transition

movement between slides

animation

movement on slides



subtle animations

dissolve

fade & scale

wipe

wipe

typewriter

dynamic animations

anvil

flame

drift

skid

skid

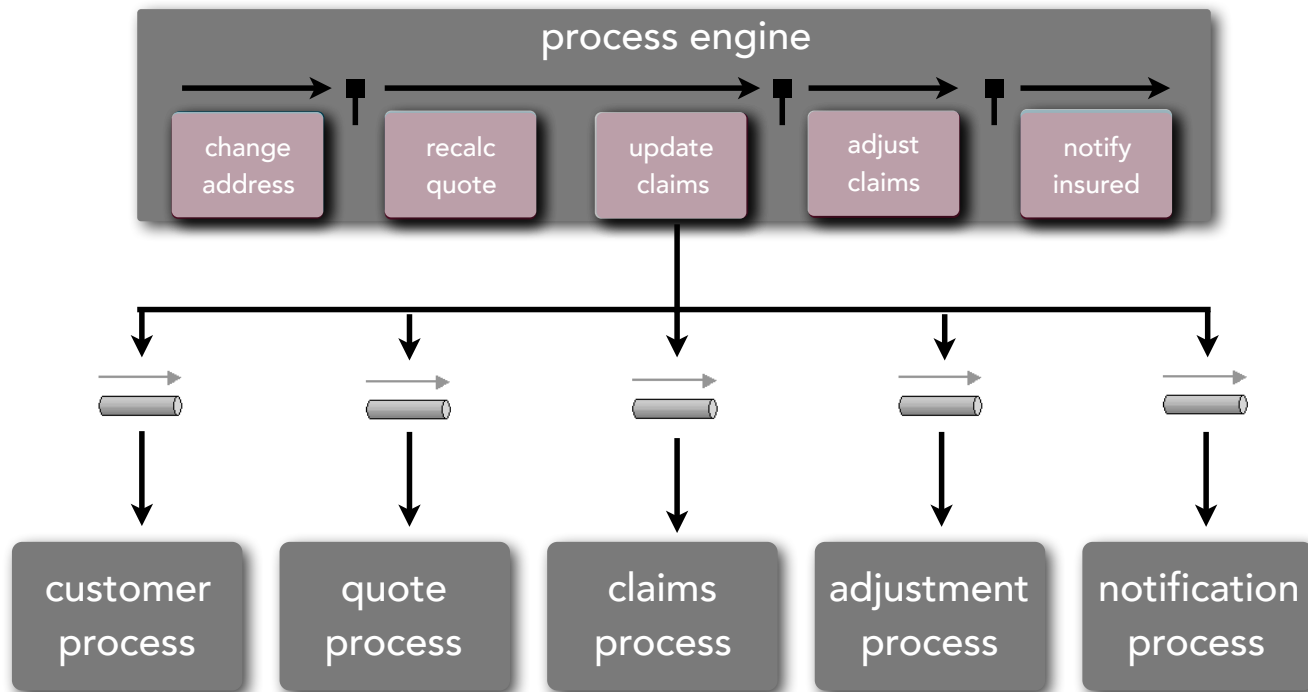
drift & scale

animations



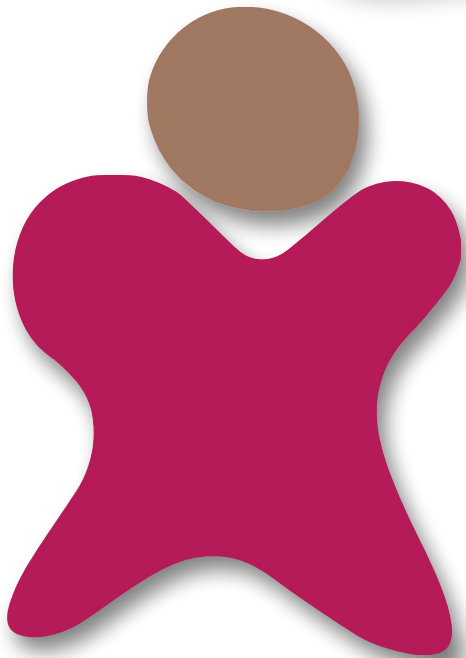
you move...

you
moved



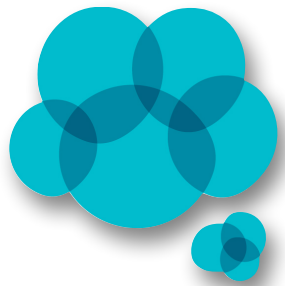


slides are ½ of the story

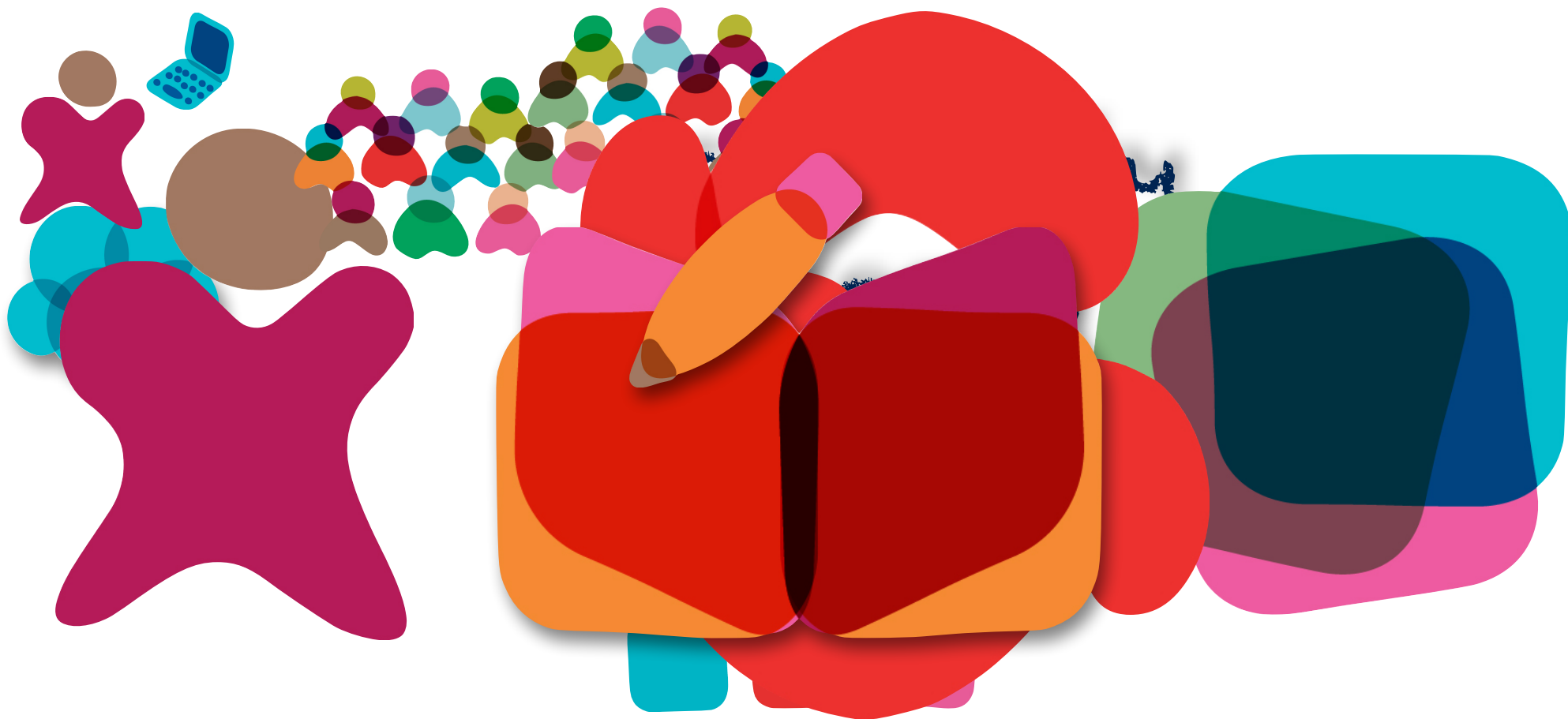


slides are $\frac{1}{2}$ of the story

(this slide intentionally
left blank)



creativity
needs
constraint



creativity needs constraint



ThoughtWorks

WE ARE THOUGHTWORKS

A community of passionate individuals whose purpose is to revolutionize software design, creation and delivery, while advocating for positive social change.

Game-changers »
A social experiment »

Our mission »
Leaders »

Our people & culture »
Our latest news »

Game-changers

We work with people and organizations who have **ambitious missions** - whether they are in the commercial, social or government sectors. We set up smart teams who love challenges and think disruptively to help our clients succeed. Our **Agile development tools** help our clients continuously improve and deliver quality software.

We are focused on helping our **industry improve**, and believe in sharing what we learn. We do this by **writing books, blogging, running events**, talking at conferences, and championing **open source**.

We are strong believers in the **power of software and technology as tools for social change**. Through our Social Impact Program, we collaborate with organizations with a humanitarian mission and broad reach, helping them use technology to make an impact.

4500 Employees
15 Countries
42 Offices

Our mission

Our mission is to better humanity through software and help drive the creation of a socially and economically just world. We bring together the most capable, driven and passionate people - we call them 'ThoughtWorkers' - to:

Run a sustainable
Champion software
Advocate passionately for

MARTIN FOWLER.COM

SyntheticMonitoring

Flávia Falé
Serge Gebhardt
25 January 2017

Synthetic monitoring (also called semantic monitoring [1]) runs a subset of an application's automated tests against the live production system on a regular basis. The results are pushed into the monitoring service, which triggers alerts in case of failures. This technique combines automated testing with monitoring in order to detect failing business requirements in production.

Find similar articles at these tags
continuous delivery
testing

during testing, many tests are run against the system in a staging environment

synthetic monitoring runs a few of these tests against the production system

staging

production

the tests join the regular traffic for monitoring

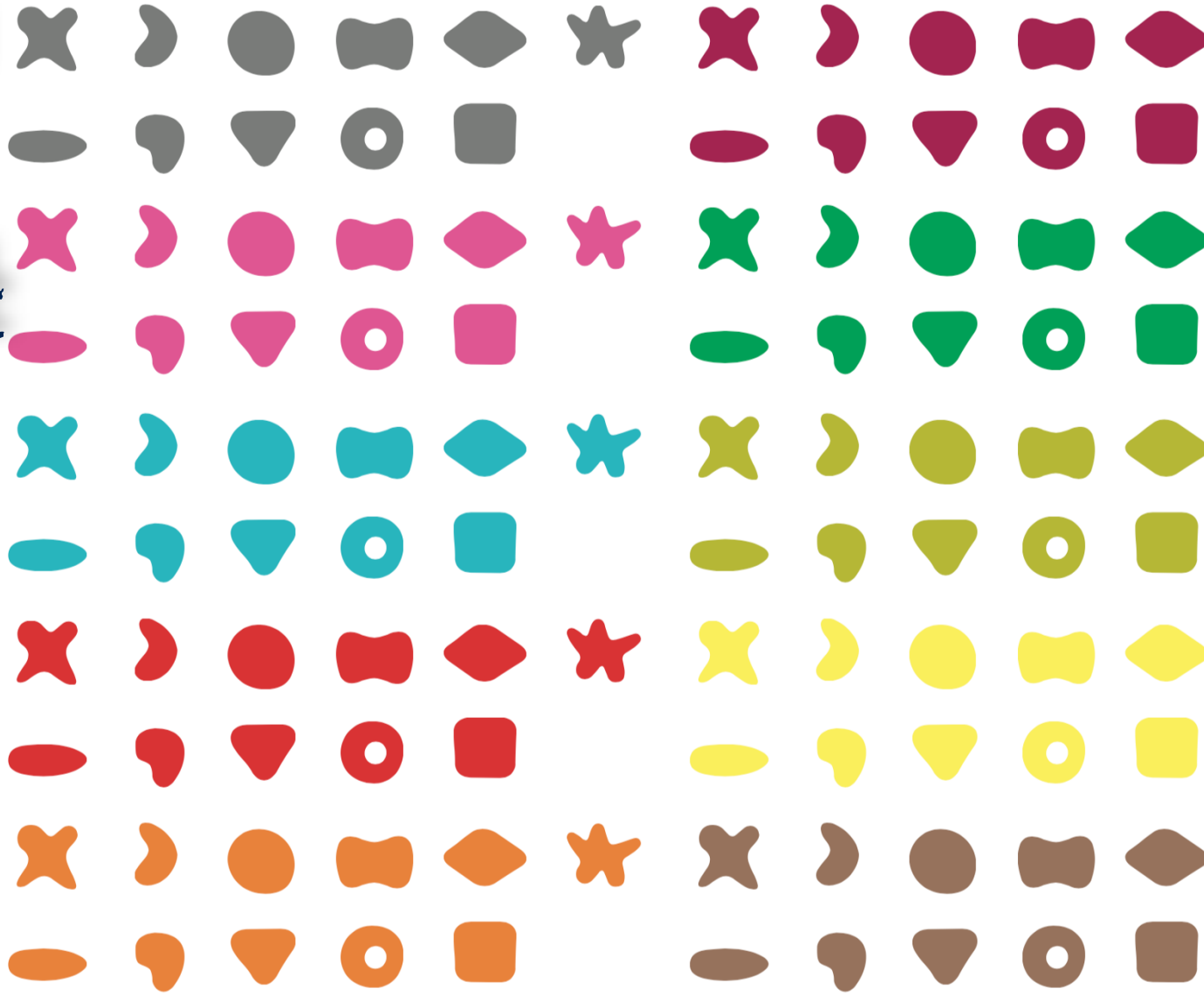
test monitoring is picked out for special display

In the age of small independent services and frequent deployments it's very difficult to test pre-production with the exact same combination of versions as they will later exist in production. One way to mitigate this problem is to extend testability from pre-production into production environments - the idea behind QA in production. Doing this shifts the mindset from a focus on Mean-Time-Between-Failures (MTBF) towards a focus on Mean-Time-To-Recovery (MTTR).

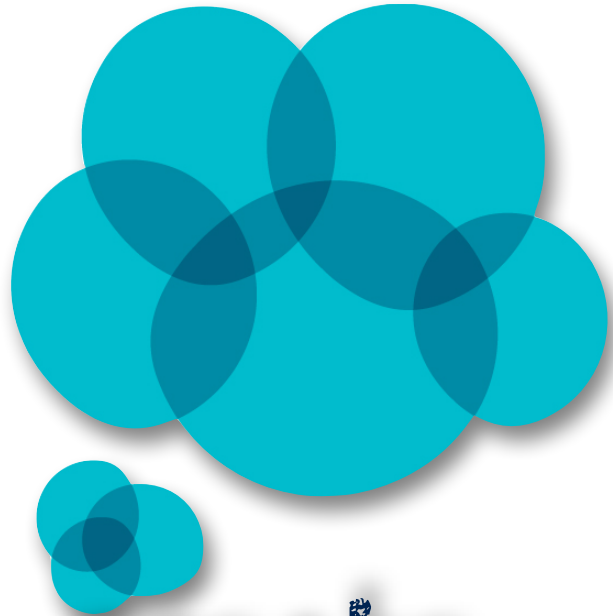
A technique for this is synthetic monitoring, which we used at a client who is a digital marketplace for cars with millions of classifieds across a dozen countries. They have close to a hundred services in production, each deployed multiple times a day. Tests are run in a ContinuousDelivery pipeline before the service is deployed to

MTTR > MTBF, for most types of F - John Allspaw

creativity
needs
constraint



cultivate



create



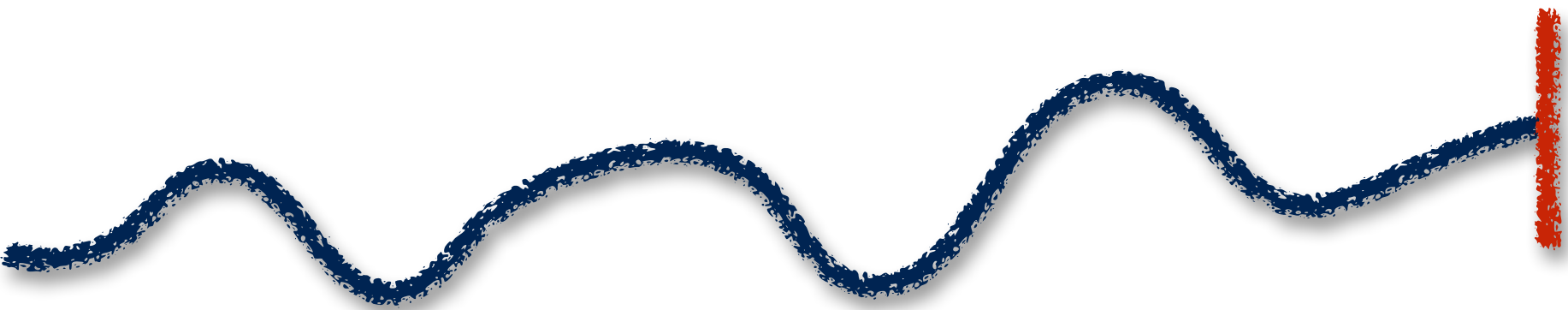
communicate

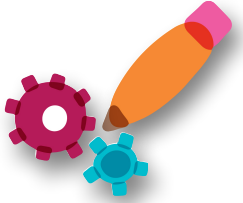


ideas need food



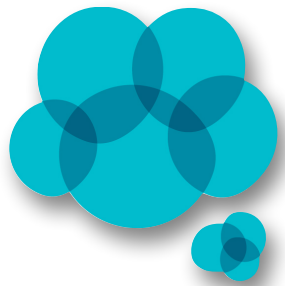
nothing useful gets done
without a deadline!



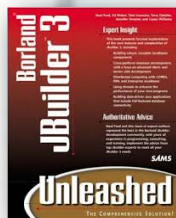
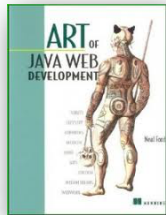
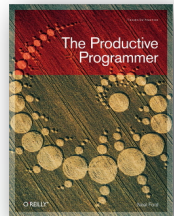
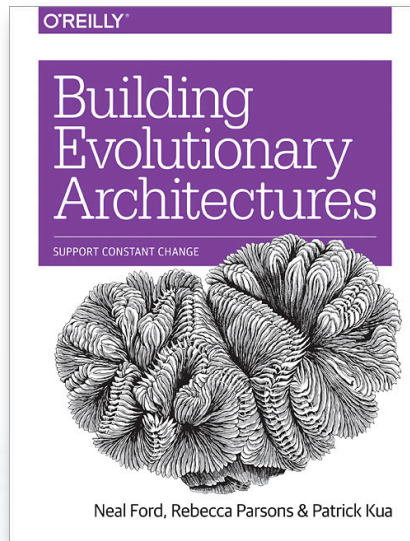
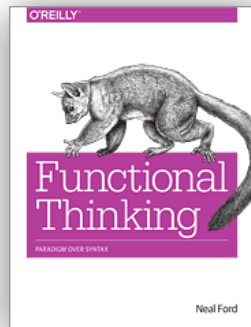
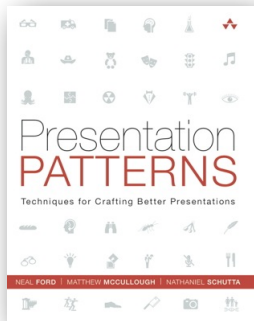


Cadence!

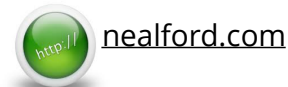




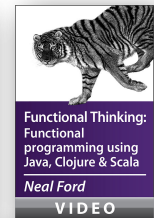
creativity
needs
constraint



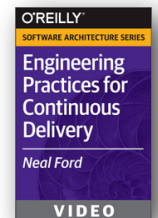
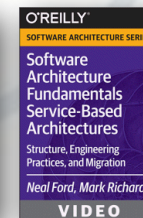
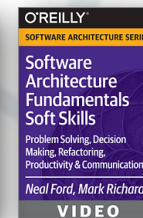
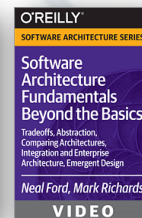
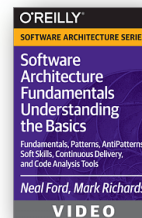
nealford.com/books



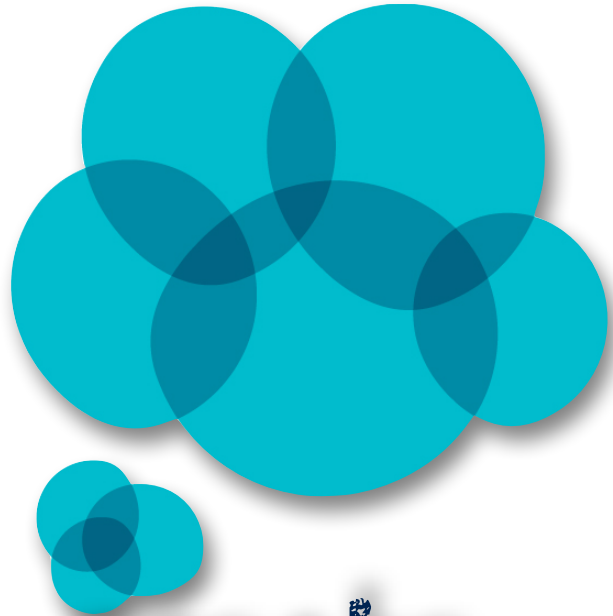
nealford.com/videos



www.oreilly.com/software-architecture-video-training-series.html



cultivate



create



communicate