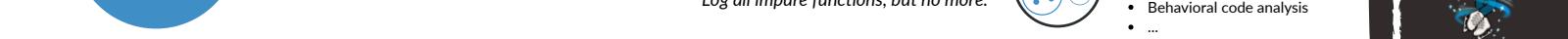
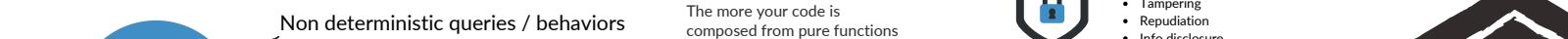
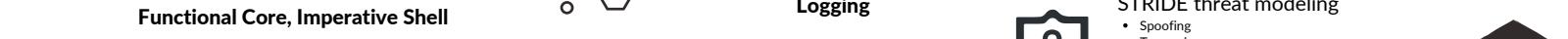
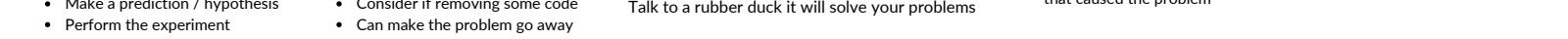
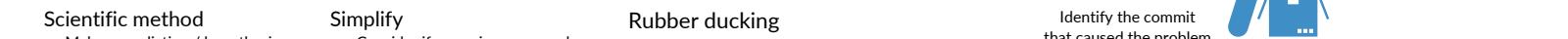
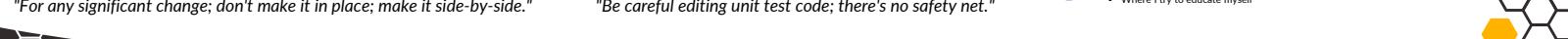
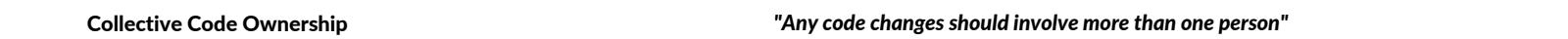
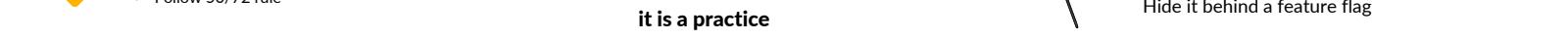
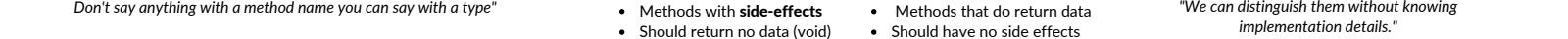
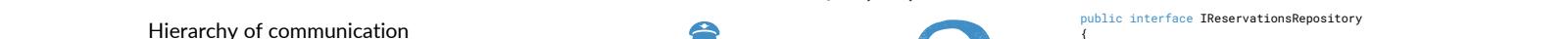
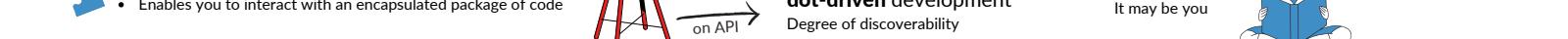
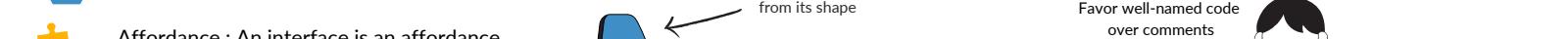
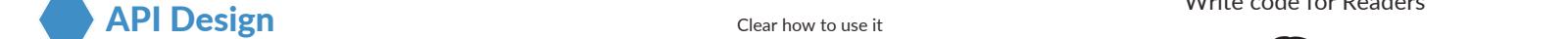
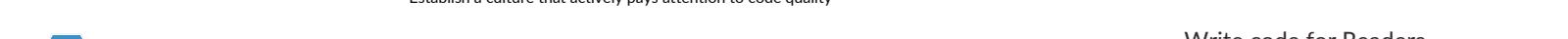
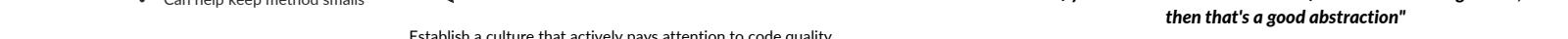
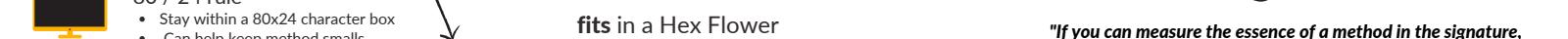
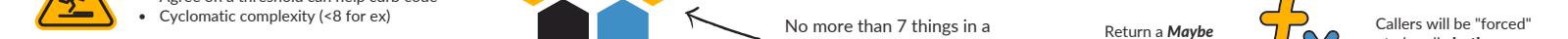
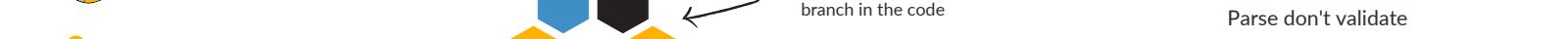
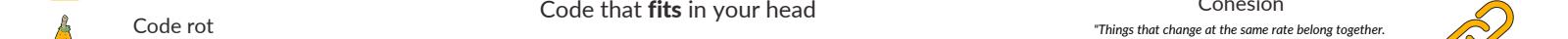
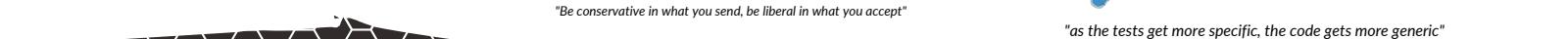
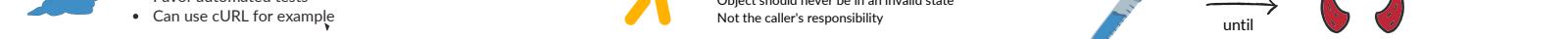
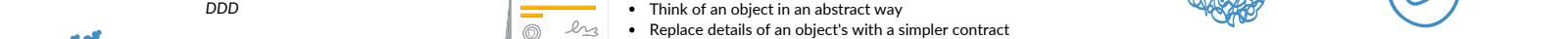
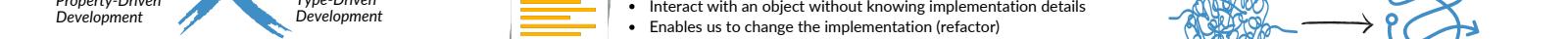
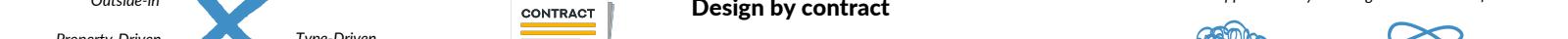
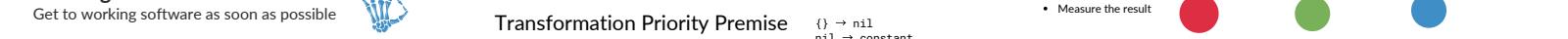


Code That Fits in Your Head

By Mark Seemann



CONTINUOUS DELIVERY

BY JEZ HUMBLE AND DAVID FARLEY

"CONTINUOUS INTEGRATION IS A SOFTWARE DEVELOPMENT PRACTICE WHERE MEMBERS OF A TEAM INTEGRATE THEIR WORK FREQUENTLY, USUALLY EACH PERSON INTEGRATES AT LEAST DAILY." - MARTIN FOWLER

CI PRINCIPLES

" ELIMINATE THE NEED FOR INSPECTION ON A MASS BASIS BY BUILDING QUALITY INTO THE PRODUCT IN THE FIRST PLACE.
- W. EDWARDS DEMING ,"



MAINTAIN A SINGLE SOURCE REPOSITORY

AVOID HAVING SOURCE CODE BEING SCATTERED ACROSS MULTIPLE LOCATIONS



KEEP THE BUILD FAST - FAIL FAST

- BUILD SHOULD NOT TAKE HOURS TO HAPPEN
- HAVE SHORT FEEDBACK LOOPS



AUTOMATE THE BUILD

A SINGLE COMMAND SHOULD HAVE THE CAPABILITY OF BUILDING THE WHOLE SYSTEM



TEST IN A CLONE OF THE PRODUCTION ENVIRONMENT



MAKE YOUR BUILD SELF-TESTING

- CONFIRM THAT IT BEHAVES AS EXPECTED
- COMPUTER CAN REPLACE HOURS OF MANUAL TESTING WITH JUST MINUTES OF AUTOMATED TESTING



EVERYONE CAN SEE WHAT'S HAPPENING

- SEND NOTIFICATIONS
- FEEDBACK EVERYWHERE
- REDUCE FEEDBACK LOOP



EVERYONE COMMITS TO THE CENTRAL REPOSITORY EVERY DAY

- AS OFTEN AS POSSIBLE - MICRO INCREMENTS
- GUARANTEE THE SUCCESS OF THE INTEGRATION

MAKE IT EASY FOR ANYONE TO GET THE LATEST EXECUTABLE VERSION

- MAKE IT EASY FOR ANYONE TO GET THE LATEST EXECUTABLE VERSION
- AVAILABLE TO STAKEHOLDERS AND TESTERS



EVERY COMMIT SHOULD BUILD ON AN INTEGRATION MACHINE

" OUR HIGHEST PRIORITY IS TO SATISFY THE CUSTOMER THROUGH EARLY AND CONTINUOUS DELIVERY OF VALUABLE SOFTWARE "

AUTOMATE DEPLOYMENT

- REDUCE THE DEPLOYMENT RISKS
- DEPLOYMENT AS A NON-EVENT



CD PRINCIPLES



REPEATABLE & RELIABLE PROCESS

- GET THE LIST OF DEPLOYED FEATURES
- NOTHING CAN BE INTRODUCED THAT HAS NOT BEEN TESTED



IF SOMETHING IS DIFFICULT,
DO IT MORE OFTEN



VERSION CONTROL EVERYTHING



DONE MEANS "RELEASED"



BUILD QUALITY IN



EVERYONE IS RESPONSIBLE

4 PRACTICES

BUILD ONCE, DEPLOY MANY

- EARLY AND OFTEN
- ARTEFACT IS ENVIRONMENT AGNOSTIC
- AUTOMATE EVERYTHING

USE PRECISELY THE SAME MECHANISM TO DEPLOY TO EVERY ENVIRONMENT



SMOKE TEST YOUR DEPLOYMENT

- "DOES CLICKING THE MAIN BUTTON DO ANYTHING?"
- DETERMINE THE STATE OF THE SYSTEM

IF ANYTHING FAILS
STOP THE LINE



CULTURE IS EVERYTHING

BY TRISTAN WHITE

THE STORY AND SYSTEM OF A START-UP THAT BECAME AUSTRALIA'S BEST PLACE TO WORK

1) DISCOVER THE CORE



CORE PURPOSE

Define yours : Inspiring / Valid in Time / Help to think Expansively / Help you Decide / Truly authentic to your company

"A CORE PURPOSE IS THE REASON AN ORGANISATION EXISTS"

CORE VALUES (3 TO 5)

- Inspire great behavior
- Make them short, sharp and memorable
- Each value should be an action statement



SHARE CORE VALUE STORIES TO REWARD / RECOGNIZE / REEDUCATE

- MVP Program
- Share stories of team members
 - Living core values
 - Celebrate their successes



2) DOCUMENT THE FUTURE

CREATE A TEN-YEAR OBSESSION THAT ACTS AS YOUR NORTH STAR



PAINTED PICTURES - 3 YEAR GOALS

- Broken down vision
- Make it : clear / specific / possible
- Communicate progress often
- Obsess over it
- Make it fun

"A STRONG CULTURE NEEDS A CLEAR VISION"

3) EXECUTE RELENTLESSLY

HAVE AN ENERGETIC DAILY HUDDLE

- Aligns everyone to the Painted Picture
- 12 minutes / day



ROBUST RECRUITMENT PROCESS

- Culture fit : examples of lived core values
- Passion for the work
- Passion for the company
- Key skills



"A STRONG CULTURE NEEDS EVERY TEAM MEMBER ALIGNED TO THE SAME VISION AND LIVING THE SAME VALUES."

4) SHOW MORE LOVE

MEMORABLE WELCOME EXPERIENCE

FACE-TO-FACE COMMUNICATION

PARTIES & CELEBRATIONS

GENUINE APPRECIATION / THKS



HAVE INTEREST FOR INFLUENCERS (Not on the payroll : Kids, Friends, Family, ...)



CULTURE BOOK

Story of your organization



19 STEPS

To build a GPTW

"CULTURE IS THE CEO'S RESPONSIBILITY : TOO IMPORTANT TO DELEGATE"

Dynamic Reteaming

The Art and Wisdom of Changing Teams

Dynamic Reteaming a.k.a. Team Change

People will **join** your team
Others will **leave**

Natural occurrence

Team



- At least two people working together
- Build something valuable for their customers
 - Shared work
 - Joint ownership of the outcome

When you **change** your team's composition, it:

- Creates a **new team social dynamic**
- Impacts the collective intelligence present on the team
- Brings **new learning** potential to the team as a whole
- Helps teams learn together and expand their skills

by Heidi Helfand

"In essence, team change is **inevitable**, so we might as well get good at it."



Collections of people assigned across different teams

Ex: Community Of Practice
To spread similar ways of working

The Social Dynamic of a Team

Own unique social dynamic / "feel"

Changes over time

High energy ----- chemistry / **high performance**

Low energy ----- lacks chemistry / **low performance**

How To ?

like Kanban recommendations



- Start **where you are**
- **Visualize** your team structures
- **Observe** and get to know them
- **Incremental** reflection / adjustment
- **Experiment** and learn

Politics of Team Assignment and Change

Reduces Risk and Encourage Sustainability

Decreases the Development of Knowledge Silos

RISK

- Within a team
 - Pair programming / TDD
 - Team-to-team level
 - Reduce the development of knowledge silos by reteam
 - Spreading knowledge out from one team to another

Reduces Team Member Attrition

Providing Career Growth Opportunities

Decreases Inter-Team Competition

Fostering a Whole Team Mentality

Less Freedom

- Someone "at the top" **put** them on the team
- Manager **put** them on the team without their input
- Manager **included** their input when assigning team
- Managers / leadership **arranged self selection events**
- Team members **trade** places / tell managers
- Team members **form** their own teams

More Freedom

"When People Leave You Have a **New Team...**"

Dynamic Reteaming Patterns

For company growth

Onboard New Team Members

- Make it **Known** That You are **Hiring** in New Team Members
- Plan and **Communicate** about the Arrival of the New Team Member
- **Get Things Together** for the New Person Before They Arrive
- Assign a specific **mentor** within their team (Pair Program)



"Too many" people : can become inefficient

Can also apply it to spread "best practices" across your organization by

- Conservatively adding in people to a stellar team
- Then splitting the team later
- When you feel all team members mastered the techniques you want to spread

Guidelines

1. Why are you splitting the team?
2. The **membership** on each of the resulting teams after the split should be made clear to everyone.
3. Try to **avoid sharing** team members between the two teams.
4. Let people **choose** which team they will move into.
5. The work of each of the split teams should be **separate**.
6. Don't let the team split **drag on forever** : choose a date on the calendar for "doing the split".
7. Consider coming up with new team names for each of the teams or engage the "new" teams.
8. Make sure any of your **tooling** is updated in advance of your team split event.
9. Determine the **facilities implications** for your team split.
10. Consider having "Team Liftoffs" or "Startups": discuss how you want to work together as a new team.
11. Get the team itself to "own the split", if possible".

Isolation



Creates beneficial silos by design
You form a team "off to the side" and give them process freedom

For the work

The **new work** is the inspiration for the team change

- **Isolation Pattern** for Pivoting & Innovation
- Form Teams and Reteam Around the Work
 - Ex: TRIAD (Product Manager, Engineering representative, UX)
- or when "**Overloaded**" with work
- If prioritization of work is not clear, people can suffer...

For the code

- **Spike**: research story that comes up from time to time in teams
- **Refactor**
- Share **Production Support**

"When you switch pairs, or teams for that matter, you are exposed to new people and new ideas.
You just learn more. That feels good to us as humans."

For Learning, Fulfillment, and Sustainability

- When you switch within a team or across teams
 - We switch to share knowledge with each other
 - The aim is to **spread out the knowledge** for learning and sustainability
 - We Want to be with other people and learn from them
- To **Support** a Feature
- Switching for Personal **Growth & Learning**
- Empower People to **Re-Role**
 - It can make your organization stickier and help you retain people

Engagement at work can happen when you are intellectually stimulated and are able to **continually learn** in your job.

Merging

teams that are **too small**

need more people to have collaboration opportunities like pair programming

Teams might merge as a strategy to combat dependencies across two or more teams

Switching

Switching

"**avoid tower of knowledge**"

"low bus count" problem -> members working a lot on their own

Incorporate reteaming to spread knowledge within or around your teams

Enables the Continuous Integration of Ideas

Mob Programming

Brings Consistency & Facilitates Reteaming

Other Reteaming Reasons

For Short-Term Events

1 hour of mob-style learning each day

To Find a Better Fit

To Liberate Silenced People Could Be a Sign for a Reteam Prisoners in meetings...

Cultivate Community

to Prime for Future Reteaming

Hold Educational Off-sites

to Sharpen Skills and Strategize

Bring Remote Workers

into the Office and Send Team Members to Them

Create Opportunities

to Get To Know Key Leaders in Different Departments: "coffee chats" with VPs, key POs, ...

Design Events

to Build Relationships Across the Organization

Retrospectives

Systemic Retrospectives

Retrospectives with Groups of Related Teams

Give Teams Budgets

to Create their Own Social Events

Reflect

on Team Compositions and How to Shift



HOW TO AVOID A CLIMATE DISASTER BY BILL GATES

Why zero ?

51

Billion Tons
of greenhouse gases to the atmosphere per year

We are here today

0

"near net zero"

What we need to aim for



Trouble getting clean water
Twice as many people

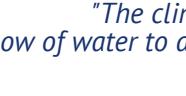


Corn production go down twice as much

2-degree rise wouldn't be 33 percent worse than 1.5. Could be 100 percent worse



Mosquitoes will start living in new places
Malaria



Heatstroke
Because of humidity



1°C increase since preindustrial times



Mid-century : between 1.5°C and 3°C



End of century : between 4°C and 8°C

By 2100 could be FIVE times as deadly than COVID 19

*"The climate is like a bathtub that's slowly filling up with water.
Even if we slow the flow of water to a trickle, the tub will eventually fill up and water will come spilling out onto the floor."*

Give a sense of how much is a lot / a little



How much of the 51 are we talking about ?

Convert numbers into a percentage of the annual total of 51 billion tons

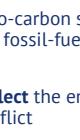


What's your plan for Cement ?

- A shorthand reminder that emissions come from 5 different activities
- We need solutions in all of them

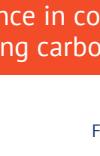


How much power are we talking about ?



How much space do you need ?

- How much space will be required to produce that much energy
 - Wind : 1-2 Watts per square meter
 - Fossil Fuels : 500-10,000 Watts per Square Meter



The difference between the 2 prices : GREEN PREMIUMS

It can be negative : green can be cheaper

"We need the premiums to be so low that everyone will be able to decarbonize."

Green Premiums

Difference in cost between a product that involves emitting carbon and an alternative that doesn't

5 types of activity

27 %

How We Plug In

- Electricity : A cheap source of energy always available
- Getting all the world's electricity from clean source won't be easy

Fossil fuels account for two-thirds of all electricity worldwide



Store electricity



Batteries

- Hard to improve on them
- Can improve by a factor of 3 but not by a factor of 50



Pumped hydro

- When electricity is cheap : pump water up a hill into a reservoir
- When demand goes up : let the water flow back down the hill



Thermal storage

- When electricity is cheap use it to heat up some material

16 %

How we get around

Bigest cause of emissions in the United States



Do less of it

Walking / biking / car-pooling



Use fewer carbon-intensive materials

in making-cars

4 ways to cut down on emissions from transportation



Use fuels more efficiently



Switch to electric vehicles

alternative fuels

96

Millions tons of cement produced every year in America

600 pounds for every person in the country

31 %

How we make things

Use tons of steel, cement, glass, plastic

Making 1 ton of steel



produces 1.8 tons of



CO₂

Offshore wind

- Putting wind turbines in an ocean or other body of water



Geothermal

- Deep underground : hot rocks that can be used to generate electricity
 - Amount of energy we get per square meter is quite low



How we grow things

70% agriculture / 30% deforestation

19 %

40% more people

- We'll need more than 40% more food too
- As people get richer, they eat more calories



Food thrown away

- 20% : Europe, Industrialized parts of Asia, Sub-Saharan Africa
- 40% in the US



We can cut down on meat eating while still enjoying the taste of meat :

- Plant based meat
- Artificial meats
- Cell based meat



7 %

How we keep cool and stay warm

Heating / cooling / refrigeration

Methane : main agriculture culprit

- 28 times more warming per molecule than CO₂ over the course of a century
- Nitrous oxide causes 265 times more warming



Wasting less of it

Behavior change

Stop deforestation

- Incentives to cut down trees are stronger than the ones to leave them alone
- We need political / economic solutions



4 tons over the course of 40 years

the CO₂ that a tree can absorb in its lifetime



The cruel injustice is that even though the world's poor are doing essentially nothing to cause climate change, they're going to suffer the most from it

“

A plan for getting to Zero

Science tells us that in order to avoid a climate catastrophe, rich countries should reach net-zero emissions by 2050

Electrofuels

Hydrogen produced without emitting carbon

Nuclear fusion

Grid-scale electricity storage

Zero-carbon plastics

Pumped hydro

Zero-carbon steel

Underground electricity transmission

Drought and flood-tolerant food crops

Advanced biofuels

Zero-carbon fertilizer

Zero-carbon cement

Plant and cell-based meat

Thermal storage

Technologies needed

Quintuple clean energy / climate-related R&D over the next decade

Make bigger bets on high-risk R&D projects



Match R&D with our greatest needs

Work with the industry from the beginning

Expand the supply of Innovation

To get these technologies ready soon

Accelerate the demand for Innovation



Put a price on carbon : eliminate Green Premiums



Change the rules

so new technologies can compete



Clean standards

- Electricity
- Fuel
- Product

Set standards in procurement programs for example

What each of us can do ?

"The market is ruled by supply and demand : we can have a huge impact on the demand side"

As a citizen

Personal action : important for the signals



VOTE

Elected officials will adopt specific plans if their voters demand it

- Make calls, write letters, attend town halls
- Run for office

As a customer



As a customer

Push your company to do its part :

- Sign up for a green pricing program : electricity utility
- Reduce your home's emissions
- Buy an electric car
- Try a plant-based burger

"We need to make it possible for low-income people to climb the ladder without making climate change worse."

As an employee or employer

As an employee or employer

Push your company to do its part :

- Set up an internal carbon tax
- Prioritize innovation in low-carbon solutions
- Be an early adopter
- Engage in the policy-making process
- Help early-stage innovators get across the valley of death

@yot88

La liberté du commandement

L'esprit d'équipage

Vice Amiral LOÏC FINAZ



- Mener des hommes au combat pour porter la mort
- Peut conduire à la recevoir

Commander

Diriger une entreprise



Partager une vision, Mobiliser l'intelligence

Structurer l'organisation



Préserver le patrimoine

Faire réfléchir et grandir



Générer de la valeur / innover

Manager



Commander 1 bâtiment de guerre
c'est aussi

Diriger 1 entreprise (manager)

Piliers de notre sagesse et de notre performance

Susciter l'initiative et accepter l'échec



AUTONOMIE ET SOLIDARITÉ

"Rassurez-vous, je suis là; si vous échouez, je corrigerai le tir; je suis là pour cela."

Des fonctions différentes, une même responsabilité



Fédérer Faire évoluer S'épanouir

FONCTIONS ET RESPONSABILITÉ

"La fonction fait l'homme tout autant que l'homme peut faire la fonction."

Hiérarchie importante pour prendre des décisions au combat



Intelligence collective pour trouver les solutions

Vis-à-vis de Soi-même (exemplarité) Ceux qui leur sont confiés

Culture participative très forte

Sans exigence 1 chef n'obtiendra / réussira rien



Sans bienveillance il détruira tout

HIÉRARCHIE ET PARTICIPATION

"Le système hiérarchique n'érigé pas la confiance, il utilise celle que fédère les chefs grâce à leur culture participative."

EXIGENCE ET BIENVEILLANCE

"[...] commander, diriger, est l'une des plus belles façons de servir ceux qui nous sont confiés."

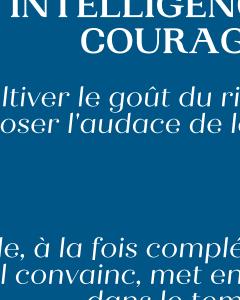
Le chef doit être une énergie : met en mouvement, convainc, fait durer, vivre et gagner



- La culture permet
 - De s'élever
 - De s'enrichir
 - De s'armer pour les luttes de l'existence

Besoin d'une cohérence entre ces 2 qualités

- Chef très intelligent et peu courageux, incapable de :
 - Décider
 - Agir



- Chef courageux et crétin :
 - Un maniaque
 - Ou 1 fou

INTELLIGENCE ET COURAGE

"[...] il faut cultiver le goût du risque et la capacité de l'assumer, oser l'audace de la solution originale."

C'est par la parole que l'action du dirigeant existe



"Par la parole, à la fois complément et expression de son énergie, il convainc, met en mouvement et s'inscrit dans le temps."

- Parole du chef adressée directement :
 - Suscite espoir et enthousiasme
 - Apaise les craintes
 - Remonte le moral (dans la crise ou la défaite)



PAROLE ET TEMPS

Apprenons à ne pas laisser de traces dans ce monde

Qui n'en vaillent pas la peine

"Rien n'irrigue plus les bonnes pratiques du management que les exigences du commandement."



QUE NOS PAS DEVIENTNENT SILLAGES



Porter nos regards sur l'horizon

AYONS TOUJOURS NOTRE REGARD SUR NOTRE LIGNE DE FOI



QUE NOS MAINS SACHENT ÉDIFIER



Tout le reste n'est que discours

COMMANDER C'EST AIMER

"Faites de vos équipes, de vos services, un équipage"



Leadership is language

The Hidden Power of What You Say and What You Don't - L. DAVID MARQUET



Redwork

Active production
"Prove"



Bluework

Thinking / Learning
"Improve"



"We are all both Redworkers and Blueworkers"

"A real danger to use old thinking in new situations"

A NEW PLAYBOOK

1) Control the clock : exiting redwork

Bluework allows us to adapt BUT you have no chance to do bluework if you don't control the clock



Make a pause possible : Invite a pause
Give the pause a name

Call a pause
Preplan the next one

"If you are on the team and see something unexpected, it's your responsibility to call a pause"



Vote first, Then discuss

Anonymous polling, Ask probabilistic questions
Use probability cards, Dot voting



"Before I tell you what I think we should do,
what would you do if I weren't there"

LEADERS SPEAK LAST

Invite dissent rather than drive consensus

Dissent cards



Give information, not instructions'

From "Park there" to "I see a parking spot there"

"A leader's obligation is to listen to the dissenters"

3) Commit

Commit to Learn, Not (just) Do

Develop hypothesis to test rather than making decisions to execute



Chunk it small
BUT do it all

Commit actions, not beliefs

Once the decision is made don't try to convince dissenters



4) Complete : the end of Redwork

Chunk work for frequent completes early

At the beginning of a project : shorter redwork periods

More frequent bluework periods to bias toward learning and improving

Celebrate FOR

"Good job" / "I'm so proud of you"

Transference of the reward to us rather than leaving it with the person

Celebrate WITH

Use descriptive statements : "I see", "I noticed", "It looks like"



Focus on Journey, Not destination

Invite people to tell their story

5) Improve : completing the cycle

"Employees with the autonomy to decide how to go about solving problems and achieving goals innovate"

Forward, Not Backward

"What do we want to do differently next time ?"



Process, not people

"How could this be done better ?"

Outward, not inward

Focusing on others instead of oneself

"What could we do better serve our customers ?"

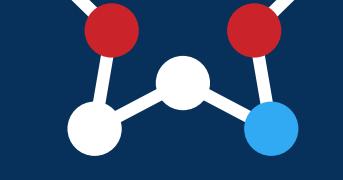
Achieve excellence, Not avoid errors

Flatten the power gradient

Amount of social distance between one person and another

Admit you don't know

Hard to connect with a Know-it-all



6) Connect : enabling play

Be vulnerable

"How is everyone feeling about this ?

I think I'm moving away from excited toward worried"

Trust first

What do we want to do differently next time ?

"Changed the way we communicated, changed the culture"

#sharingiscaring

by Yoan THIRION @yot88

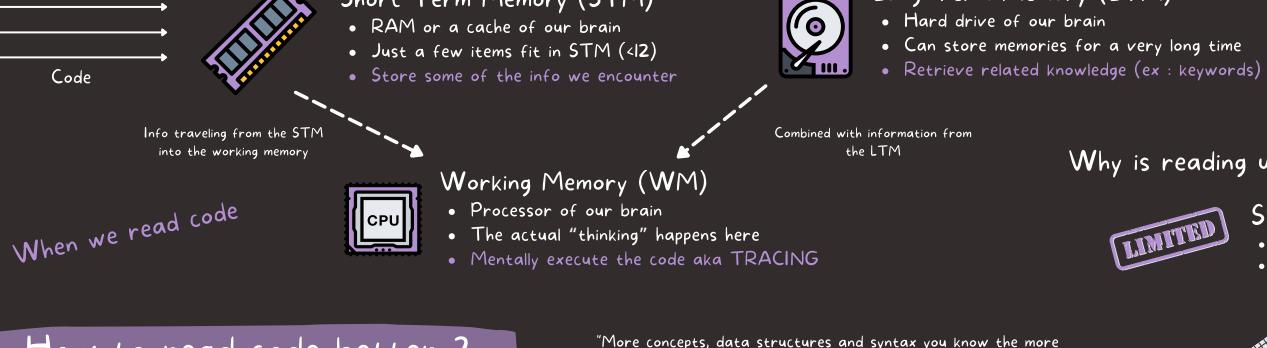
THE PROGRAMMER'S BRAIN

by Felienne Hermans

60%



of our time



Why is reading unfamiliar code hard?

LIMITED

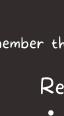
Short term memory

- Time : 30 seconds
- Size : 7 +/- 2 things

How to read code better ?

Learn programming syntax

Use Flashcards
• Front : prompt
• Back : corresponding knowledge



Remember syntax longer
• Retrieval : trying to remember something
• Elaboration : connecting new knowledge to existing memories

Read / Hide / Write code exercises

How to not forget things ?

Spaced repetition
• Practice regularly
• Best way to prevent forgetting



Revisit your Flashcards

- Once a month
- Each repetition strengthens your memory

DON'T FORGET

After 2 days, just 25% of the knowledge remains in our LTM

Read complex code easier

Reduce cognitive load

Refactoring code
Ex : replace unfamiliar language constructs



Dependency graph
• Circle variables
• Draw lines between occurrences

Cognitive load
• Capacity of our Working Memory
• Capacity : 2 to 6 "things"

State table
• Focuses on the values of variables
• 1 column / variable
• 1 line / step in the code



"Our ability to learn a natural language can be a predictor of your ability to learn to program."

Roles of variables
(Sajaniemi's framework)

- Fixed value : value does not change after initialization
- Stepper : variable stepping through a list of values
- Flag : has happened or is the case
- Walker : traverses a data structure (search index)
- Most-recent holder : holds the latest value encountered
- Most-wanted holder : holds the best value found so far

Understanding what types of information variables hold is key to being able to reason about and make changes to code.

"Many similarities between reading code and reading natural language"

Activating

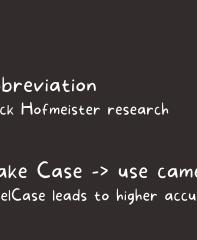
Actively thinking about code elements help our WM to find relevant information stored in the LTM

Monitoring

- Keep track of what we are reading and our understanding
- Ex : ticking the lines

Inferring

Inferring the meaning of variable names



Determining importance

Identify which parts of the code are likely to have the most influence on the program's execution



Questioning

- Asking ourselves questions while reading code
- Help us understand the code's goals and functionality
- Ex : What are the 5 most central concepts of the code?

Visualizing

List all operations in which variables are involved (dependency graph, state table,...)

Goal of the code: what is the code trying to achieve?
Most important lines of code
Most relevant domain concepts
Most relevant programming constructs
...

- Write a summary of code in natural language
- Help us gain a deeper understanding of what's happening in that code

Write better code

Search...



Avoid



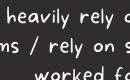
Abbreviation
Check Hofmeister research

Clear names help our LTM
LTM searches for related informations



Snake Case -> use camel Case
camelCase leads to higher accuracy

Methods that do more than they say



Methods that say more than they do
Identifiers whose name says that they contain less than what the entity contains

LTM can store different types of memory



Procedural / Implicit

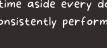
- How to do something
- Ex : How to run a bike

Memories

Declarative / Explicit

- Memories we are explicitly aware of
- Facts we can remember

"Experts heavily rely on episodic memory when solving problems / rely on solutions that have previously worked for similar problems."



Episodic

- Memories of experience
- Ex : meeting our wife / husband

Semantic

- Memories for meanings / concepts / facts
- Ex : $10 \times 10 = 100$



Getting better at solving complex problems

Automation

Create implicit memories

"Set some time aside every day to practice and continue until you can consistently perform the tasks without any effort"



Deliberate practice to improve skills

- Repeat a lot
- It frees up cognitive load for larger problems
- Ex : deliberately type 100 for loops when struggling with it

Deliberate practice : requires focused attention and is conducted with the specific goal of improving performance.

Study worked examples

Create episodic memories



Their occurrence in 7 open-source projects :

11% of setters also return a value

25% of methods : method name + comment = opposite descriptions

64% of identifiers starting with 'is' turned out not to be Boolean

20

% of developers time on interrupts

20

20

Better handle interruptions

Prepare for it



Help your "Prospective memory"

- Put TODO comments in the part of the code
- Remind you to complete / improve part of the code

15' to start editing code after an interruption

20

20

20

20

20

20

20

20

20

20

20

20

20

20

20

20

20

20

20

20

20

20

20

20

20

20

20

20

20

20

20

20

20

20

20

20

20

20

20

20

20

20

20

20

20

20

20

20

20

20

20

20

20

20

20

20

20

20

20

20

20

20

20

20

20

20

20

20

20

20

20

20

20

20

20

20

20

20

20

20

20

20

20

Refactoring at Scale



By Maude Lemaire

Refactoring

Restructure existing code
WITHOUT changing its external behavior



At Scale

- One that affects a substantial surface area of your systems
- Involves typically large codebases

Benefits

- Increase developer productivity
- Greater ease identifying bugs



Risks

- Serious Regressions
- Unearthing Dormant Bugs
- Scope Creep



Shift in Product Requirements

Performance issues

Using a new Technology



Code Complexity Hinders Development

Small Scope

For Fun or Out of Boredom

Because You Happened to Be Passing By



When You Don't Have Time

To Make Code More Extendable

When NOT ?

PLANNING

MEASURE OUR STARTING STATE



Measure Code Complexity

- Halstead metrics
- Cyclomatic Complexity
- NPath Complexity



Test Coverage Metrics

- Quantitatively : proportion of code under test
- Qualitatively : suitable test quality has been attained



Documentation

- Formal : everything you most likely think of as documentation
- Informal : Chat / email transcripts, Bug Tracking system, ...



Version Control

- Commit messages : keywords for given code
- Commits in Agg : change frequencies, authorship



Reputation

- Low-effort means of collecting reputation data
- Interview fellow developers



Build a Complete Picture

Pick one metric from every category

DRAFT A PLAN



Define your end state

Outline all starting metrics and target end metrics

Map the shortest distance

- Open a blank document technique
- OR Gather a few coworkers



Identify Strategic Intermediate Milestones

- 1) Does this step feel attainable in a reasonable period?
- 2) Is this step valuable on its own?
- 3) If something comes up, could we stop at this step and pick it back up easily later?



Dark Mode / Light Mode

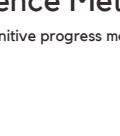
Compare pre-refactor and post-refactor behavior :

- Both implementations are called
- The results are compared

Light

The results from the OLD implementation are RETURNED

Choose a Rollout Strategy



Identify Strategic Intermediate Milestones

- Put in place an abstraction
- Enable dark mode
 - Monitor any differences between the 2 result sets
 - Track down and fix any potential bugs in the new implementation
- Enabling dark mode to broader groups of users
 - Continue logging any differences in the result sets
 - Opt groups of users into light mode
 - Until everyone is successfully processing results from the new implementation
 - Disable execution of both code paths
 - Remove the old logic



Clean Up Artifacts

- Feature Flags
- Dead Code
- Comments (TODOS)



Reference Metrics

Include definitive progress metrics



Share your plan

- Provide Transparency
- Gather perspective to strengthen it

"No refactor is complete unless all remaining transitional artifacts are properly cleaned up"

GET BUY-IN

Always remember



Aren't Coding

See the Risk

Are Evaluated Differently

Need to Coordinate

Using Conversational Devices

Persuade Them

(some techniques)



Rely on Evidence

Managers

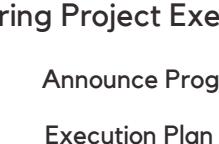
Play Hardball

2 Ways to Enlist Someone

BUILD THE RIGHT TEAM

Matchmaking

Match each expertise with one or more people



Subject matter experts (SMEs)

- Agreed to be available to talk through solutions with you
- Answer questions
- Can do some code review

EXECUTION

Active Contributor

- Heavily involved from day one
- Actively contributing to the effort by writing code
- Consulted for input on the execution plan

Stand-Ups

Everyone aligned at regular intervals



Retrospectives

Reflect on the latest iteration cycle

Weekly Syncs

- 1st part : accomplishments
- 2nd part : discuss any important topics

When Kicking Off

Single Source of Truth

Choose a platform to collect all documentation

Set Expectations

Draft a communication plan

During Project Execution



Announce Progress

Execution Plan

Living Version

Within Your Team

Outside Your Team

"To execute on a large refactoring effort successfully, we need our own Ocean's 11 [...] a team just the right size with just the right skills"

COMMUNICATION

PROGRAM PRODUCTIVELY

Early and often

Help move faster

Know your solution won't be perfect

Not spend too much time perfecting the details

Be willing to throw code away



Keep Things Small

- Commit small, incremental changes
- Makes it much easier to author great code



Test, Test, Test

- Confirm everything has remained unaffected
- Or pinpoint the precise moment at which the behavior diverged



Asking the "Stupid" Question

- Prioritize clarity
- Over maintaining an illusion of omniscience

MAKE THE REFACTOR STICK

Prototype

Foster Adoption through education

TUTORIAL

ACTIVE

PASSIVE

- Step-by-step tutorials
- Online courses, ...



Integrate Improvement into the Culture

To maintain a healthy codebase

- Continuous small refactoring
- Incrementally improve areas of the codebase



Hold design reviews

- Early in the feature development process



Encourage design conversations

Case Studies @Slack

- Redundant Database Schemas
- Migrating to a New Database



Technical Agile Coaching

with the Samman Method

by Emily Bache

A METHOD for people who want to make a difference and improve the way software is built

Wording



Samman : Swedish word for "together"
Describes this coaching method

Ensemble : French word for "together"
Describes Mob Programming



Focus on

Technical practices
How people write code



Foundation

Cultivate good relationships
Effective ways to learn from one another
Change behaviours for the long term

WHY ?

Build new features with
Shorter lead time
Higher quality
Attract skilled developers
Avoid drowning in technical debt
Increase business agility
and success

HOW ?

Ensemble working
Learning Hours



ON WHAT ?

Incremental / Iterative Development
Safe refactoring



Better unit tests
Continuous Integration

TIMELINE

10-20 coaching days / Team



LEVELING UP A WHOLE TEAM TOGETHER

Software development these days is a team sport and it doesn't work to only train individuals.

Samman coaching aims to **create a whole-team culture shift**.

EXPECTED OUTCOMES

1) AWARENESS ON
Good unit tests
Continuous Integration
Refactoring



2) NEXT
Successfully meet deadlines
Deliver High Quality Code

MEASURES

Attitudes
Deadlines met
Bugs reduction
Productivity



Friendly people collaborating
like musicians

ENSEMBLE WORKING

"All the brilliant minds working together on the same thing, at the same time, in the same space, and at the same computer - We call it 'Mob Programming' - Woody Zuill"

ROLES IN THE "ENSEMBLE"



TYPIST

Has the keyboard and mouse
Enter the code for the Ensemble



NAVIGATOR

Speaks for the Ensemble
Explains what code enter



COACH

Promote better ways of working
Spread Knowledge



TEAM-MEMBERS

Lead the work
Talk and make the decisions



FACILITATOR

Remind working agreements
Help to reflect and improve



OTHER ROLES

Researcher : Search for the Ensemble
Archivist : Log choices



LET THE ENSEMBLE GIVE YOU SUPERPOWERS

Learn as much from the team as they learn from you
Keep your technical skills sharp & up-to-date
Continue to write code every day



KINDNESS, CONSIDERATION AND RESPECT

Treat everyone with kindness, consideration, respect
Pay attention
Yes and ...
Call out bad behavior

COACHING BEHAVIORS IN THE ENSEMBLE

Teach

Breathing space

Coach

Retrospect

Mentor

Facilitate



Observe

Take Short Breaks

Turn up the good

A lot of great sample sessions
are described in the book

LEARNING HOURS



SHORT TRAINING SESSIONS

People practice coding skills
Learn new techniques



WHY 1 HOUR EVERY DAY ?

Become more productive and happier
Add up more than compensate the time you spent

For an organization to succeed in the modern world, it needs to be a learning organization

LEARNING OUTCOMES AND OBJECTIVES



What really matters :
What happens afterwards ?
Will they be able to apply what they've learnt ?
What is the outcome you're hoping to achieve ?
Start with the end in mind



4C LEARNING MODEL

by Sharon Bowman

Connect : Get people in the right head-space

Concept : Introduce the new skills you want the participants to learn

Concrete : Hands-on exercises to practice

Conclusions : An opportunity for people to consolidate



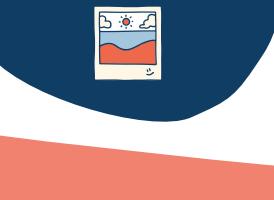
MOVEMENT TRUMPS SITTING

Learn more when you're feeling awake



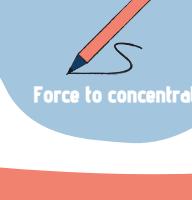
TALKING TRUMPS LISTENING

Talk reinforces your memory



IMAGES TRUMP WORDS

Tell stories that bring pictures in mind



WRITING TRUMPS READING

Force to concentrate



SHORTER TRUMPS LONGER

10-20 minute chunks



DIFFERENT TRUMPS SAME

Vary the format

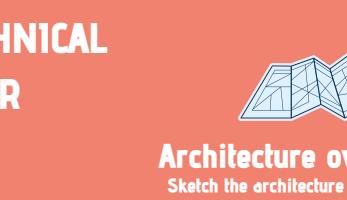


DESIGN LEARNING EXPERIENCES THAT FIT WITH HOW THE BRAIN WORKS



SAMMAN COACHING ENGAGEMENTS

IDEAS TO IMPROVE



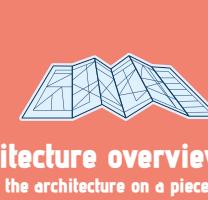
PREPARING FOR A TECHNICAL COACHING CAREER



CORE SKILLS

TDD / Refactoring
Software Design
Continuous Integration
Pair programming
Designing test cases

Presenting topics
Chairing a meeting
Facilitating a retrospective
Sketching / explaining a design
Live code



Architecture overview - 15'

Sketch the architecture on a piece of paper



Structured discussion - 30-40'

TRIZ, Lean Coffee, Speedboat



Issues in the codebase - 45'

Show me :

- Typical tests
- Code you would like to have tests for
- Well designed code
- Buggy code



Takeaways - 5'

Organize your notes by team



Attend formal training (SM or Agile Coach)

Join a meetup and organize events

Give presentations at an internal COP

SOFTWARE CRAFT

TDD, Clean Code and other essential practices

Cyrille Martraire
Arnaud Thiéfaine
Dorra Bartaguz

Fabien Hiegel
Houssam Fakih

Toolbox

Develop ?

"Working code is a low bar."

Software is never finished
It is always changing



It is read /understand the code
At least as much as writing it

Define the need
As difficult as writing the matching code



No hacker
Not a virtuous

Test-Driven Development (TDD)

Write a failing test



Make the test pass
as quickly as possible

Our tests



- Transcribe the business rule
- Easy to determine cause of failure

Should / When
Given / When / Then

DivideShould.Throw_an_invalid_operation_when-denominator_is_zero

Improve the code
Better readability

Reduce fears
Break down the problem

Take the time to make
the code readable
Refactoring

Help to define
an objective to achieve

Write as little code
as possible

Efficiently program complex features



Uncle Bob's 3 rules of TDD

- Only write code to make a test pass
- When writing a test, write the minimal to make a test fail; this includes your code not compiling
- Write the minimal amount of code to make a test pass



Techniques and Principles of Clean Code

"Any fool can write code that a computer can understand. Good programmers write code that humans can understand."
- Martin Fowler

Degradation in the urban environment
If not repaired: leads to further damage



Tendency to be sloppy
Faced with a degraded environment

Theory of the broken window



Comment sparingly



How to express the same knowledge through code?

Often a code smell
Concerns about cleanliness?



Don't comment bad code - rewrite it - Brian Kernighan

Report a subtlety
Optimization for example

Flag problems
TODO / FIXME

Legal information

Boy scout rule

KISS
Keep It Simple and Stupid

YAGNI
You Ain't Gonna Need It

Be exemplary

Storytelling

- Our code must tell a story
- Should be read as a table of contents

Cut out the functions

Focus on the service provided

f(x)
Few parameters

Must do only one thing

Only 1 level of abstraction

The 4 Rules of Software Design - Kent Beck

Passes the tests
work as intended



No duplication DRY

Reveals intention
be clear



Fewest elements
keep only what is important

Also applies to tests
Tests should be explicit
• Put as much or more effort into it

Split in a way to facilitate discovery + navigation



Properly format your code
Respect of the standard defined by the team
• Maximum indentation level limited (2 max)
• Lines of code not very wide



Contractual simplicity
Consider each parameter as immutable



Define variables as closely as possible to their use

Agile Specifications with Behavior-Driven Development (BDD)

3 Amigos

Discuss the features to be built



Represents the implementation
Technical feasibility



In the language of business



Expressing the need
Concrete details
Self-sufficient

Identify key scenarios

LIVE STREAMING
Increments of specifications

Short workshops
30 / 60%

Challenge the other 2
Identify failures in thinking
Problems in expressing needs

Acceptance criteria
to determine the progress of developments

Shared understanding among all

Non-regression tests
with good functional coverage

Living Documentation
evolutionary

Feature: Offer discounts to loyal customers
Offer a discount of 10% to all customers
As a book seller:
I want to offer discounts to loyal customers
Example: No discount table is sick of purchases
Discount table is sick of purchases
I want to offer a discount of 10%
When the customer checks out:
Then the total does not include any discount

Having conversations is more important than capturing conversations is more important than automating conversations - Liz Keogh

Improve the efficiency of collaboration between the specialists involved to build better software at lower cost.

Work with legacy code

"Code that does not have tests".

Keep your calm
Stay focused

Get an overview of the code

Assess the situation

Presence of tests?

Isolate dependencies
Seams

Make it testable

Failing Acceptance Test

ATDD and double loop

Failing Unit Test

ATDD Loop

Refactor

ATDD Loop

Acceptance Test Driven Development

Enrich them

Test code conforms to the behavior of production code

Testable code?

Business understanding approach

In front of a clear code

Code observation approach

In front of a cryptic code

Reliability

Mutation Testing

Build a Golden Master

Code coverage

Refactoring

Reliability

Test Doubles

Dummy Object

Puppet object

Imitate the desired behavior for a test

Test Spy

Ensure number of calls, args, ...

F.I.R.S.T.

Fast

Incentive to run tests

Repeatable

Ininitely executable

On different environments

Isolated/Independent

Do not depend on each other

Self verifying

Success or Failure

Timely/Thorough

Timely and accurate

Covers only 1 use case

Refactor the test code as well

Design techniques

SOLID Principles

Tell don't ask

Hollywood Principle
Don't call us, we'll call you

Hooks / Callbacks

Demeter's law

Only talk to immediate friends

Functional approach

Composition over inheritance

Low coupling high cohesion

Dependency from specific to generic

Fluent API

Builder

...

Object calisthenics

• Use only one level of indentation per method

• Don't use the else keyword

• Wrap all primitives and strings

• Use only one dot per line

• Don't abbreviate

• Keep all entities small

• Don't mix multiple levels of indentation

• Use first-class collections

• Don't use any getters/setters/properties

...

Introduce the Craft in your Context

Interact with others

Videos / Podcasts

Progress kata after kata

FizzBuzz

Tennis

RomanNumerals

Grided Rate

Treasure...

Individuellement

Within the organisation

Initiate a dynamic

Supported by sponsors

1 workshop per week

• Code kata

• Motivated people

No support needed

Intensify/generalize

Reaching out to those at a distance

Action examples

• Craft skills in job descriptions

• Recruit enthusiastic developers for craft

• Set aside time to practice

• Recruit / involve craft coaches

• ...

Convince sponsors

Economical

• Cycle time

• Reduction in the number of defects

SOFTWARE CRAFT

TDD, Clean Code et autres pratiques essentielles

Cyrille Martraire
Arnaud Thiéfaine
Dorra Bartaguz

Fabien Hiegel
Houssam Fakih

Boîte à outils

Développer ?

"Working code is a low bar."

- Un logiciel n'est jamais fini
Il change en permanence
- Définir le besoin
Aussi difficile que d'écrire le code correspondant



C'est lire et comprendre le code
Au moins autant que l'écrire



Pas de hacker
Ni de virtuose



Test-Driven Development

Clean Code



Behavior-Driven Development

Domain-Driven Design

Pair / Mob Programming

Legacy remediation

OO, FP, SOLID

Développement dirigé par les Tests (TDD)

Ecrire 1 test qui échoue

Améliorer le code
Meilleure lisibilité



Faire passer le test
le plus rapidement possible

Nos tests



- Retranscrit la règle de gestion
- Facile de déterminer cause de l'échec

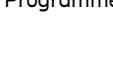
Should / When
Given / When / Then

DivideShould.Throw_an_invalid_operation_when-denominator_is_zero

Réduire les peurs
Décomposer le problème



Aide à définir 1 objectif
à atteindre



Prendre le temps de
rendre le code lisible
Refactoring

Ecrire le moins de
code possible



Programmer efficacement des fonctionnalités complexes

Les 3 règles - Uncle Bob

- On doit écrire 1 test qui échoue avant d'écrire n'importe quel code de production
- On ne doit pas écrire plus de tests que ce qui est nécessaire pour échouer (ou ne pas compiler)
- On ne doit écrire que le code suffisant pour que le test actuellement en échec réussisse

Techniques et principes de propreté de code

"Any fool can write code that a computer can understand. Good programmers write code that humans can understand."
- Martin Fowler

1 exercice de communication

Dégénération dans un environnement urbain
Si pas réparé : entraîne d'autres dégradations



Propension au laissez-aller
Face à 1 environnement dégradé



4 règles pour 1 design simple - Kent Beck

Passer les tests

fonctionner comme prévu



Eviter la duplication DRY

Révéler l'intention faire preuve de clarté



Rester petit

ne conserver que ce qui est important

S'applique également aux tests
Le meilleur des tests sera exemplaire
à mettre autour d'eux cette règle

Mettre l'accent sur le nommage

Exprimer l'intention
• Expliquer ce qu'on cherche à faire
• Pourquoi on veut le faire

s'appuyer sur le métier

Noms communs Verbes Renommage

Commentaires avec parcimonie



Comment exprimer la même connaissance à travers le code ?

Souvent 1 code small Souci de propriété ?

Don't comment bad code - revert it - Brian Kernighan

Signaler une subtilité Optimisation par exemple

Marquer des problèmes TODO / FUTURE

Mentions légales

Être exemplaire



Storytelling

- Notre code doit raconter une histoire
- Devrait se parcourir comme une table des matières

Découper les fonctions

Se focaliser sur le service rendu

Ne doit faire qu'une seule chose

Peu de paramètres

1 seul niveau d'abstraction

Découper de façon à faciliter découverte + navigation

Bien formater son code

- Respect du standard défini par l'équipe
- Niveau maximal d'indentation limité (2 max)
- Niveau de code pas très larges

Simplicité contractuelle

1 seul paramètre en sortie

Encapsuler l'input complexe dans 1 type dédié

Définir les variables au plus près de leur utilisation

Spécifications agiles avec le développement dirigé par le comportement (BDD)

3 Amigos

Discuter des fonctionnalités à construire

Product Owner

Représente le besoin Recherche de valeur

Product Manager

Représente la mise en oeuvre Feasibilité technique

Dev

Testeur

Testuse

Challenger les 2 autres Identifier failles de raisonnement Problèmes dans l'expression des besoins

Exemples concrets

Dans le langage du métier



- Exprimer le besoin
- Détails concrets
- Auto-suffisants

Identifier scénarios-clés



D'incréments de spécifications

Ateliers breifs 30 min / jour

Améliorer l'efficacité de la collaboration entre les spécialistes impliqués afin de construire de meilleurs logiciels au meilleur coût.



Représenter la mise en oeuvre Feasibilité technique



Testeur



Testuse



Challenger les 2 autres Identifier failles de raisonnement Problèmes dans l'expression des besoins



Représenter le besoin Recherche de valeur



Product Manager



Challenger les 2 autres Identifier failles de raisonnement Problèmes dans l'expression des besoins



Représenter le besoin Recherche de valeur



Product Manager



Challenger les 2 autres Identifier failles de raisonnement Problèmes dans l'expression des besoins



Représenter le besoin Recherche de valeur



Product Manager



Challenger les 2 autres Identifier failles de raisonnement Problèmes dans l'expression des besoins



Représenter le besoin Recherche de valeur



Product Manager



Challenger les 2 autres Identifier failles de raisonnement Problèmes dans l'expression des besoins



Représenter le besoin Recherche de valeur



Product Manager



Challenger les 2 autres Identifier failles de raisonnement Problèmes dans l'expression des besoins



Représenter le besoin Recherche de valeur



Product Manager



Challenger les 2 autres Identifier failles de raisonnement Problèmes dans l'expression des besoins



Représenter le besoin Recherche de valeur



Product Manager



Challenger les 2 autres Identifier failles de raisonnement Problèmes dans l'expression des besoins



Représenter le besoin Recherche de valeur

Product Manager

Challenger les 2 autres Identifier failles de raisonnement Problèmes dans l'expression des besoins

Représenter le besoin Recherche de valeur

Product Manager

Challenger les 2 autres Identifier failles de raisonnement Problèmes dans l'expression des besoins

Représenter le besoin Recherche de valeur

Product Manager

Challenger les 2 autres Identifier failles de raisonnement Problèmes dans l'expression des besoins

SOFTWARE DESIGN X-RAYS

Fix Technical Debt With Behavioral Code Analysis by Adam Tornhill



Technical Debt

- Explain the need for refactorings
- Communicate technical trade-offs



Apply at all levels (Micro and Macro)
Interest Rate Is a Function of Time

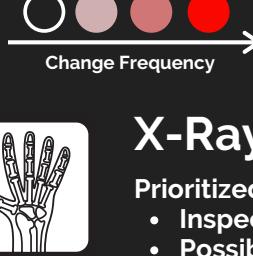
Bad Code is Technical Debt if you have to
PAY INTEREST ON IT

Identify Code with High Interest Rates

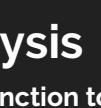
Prioritize Technical Debt with Hotspots

Complicated code that you have to work with often

- Change frequency of each file
- Lines of code as a simple measure of code complexity

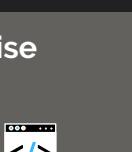


Hotspot



Evaluate Hotspots with Complexity Trends

- Complexity : indentation-based complexity
- Language agnostic



X-Ray analysis

Prioritized list of function to :

- Inspect
- Possibly refactor

Coupling in Time - A Heuristic for the Concept of Surprise

Change coupling - 2 (or more) files change

- Invisible in the code itself
- Mine it from code's history and evolution



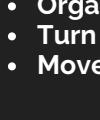
Is and Isn't Temporal Coupling
(ex : Unit Tests)

Neither good nor bad
all depends on context



"Change coupling can help us design better software as we uncover expensive change patterns in our code"

Refactor Congested Code with the Splinter Pattern

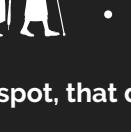


Break a hotspot into smaller parts

- Along its responsibilities
- Maintaining the original API for a transient period

"Parallel Development Is at Conflict with Refactoring"

How to ?



1. Ensure tests cover the splinter candidate
2. Identify the behaviors inside your hotspot
3. Refactor for proximity
4. Extract a new module for the behavior with the most development activity
5. Delegate to the new module
6. Perform regression tests
7. Select the next behavior to refactor and start over at 4

Stabilize Code by Age



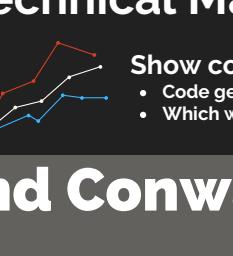
- Promotes long-term memory models of code
- Less cognitive load : less active code
- Prioritizes test suites to shorten lead times

"Always remember that just because some code is a hotspot, that doesn't necessarily mean it's a problem."

Divide and Conquer with Architectural Hotspots

Identify your architectural boundaries :

Often based on the folder structure of the codebase



Analyze the files in each architectural hotspot

Hotspot analysis on an architectural level :

- Identify the subsystems with the most development effort
- Visualize the complexity trend of a whole architectural component

Fight the Normalization of Deviance

- Each time you accept a risk, the deviations become the new normal
- Complexity trends as WHISTLEBLOWERS

"The more often something is changed the more important it is that the corresponding code is of high quality so all those changes are simple and low risk"

Communicate with Nontechnical Managers - Data buys trust



% of commits involving top hotspots

- Demonstrate importance of this code
- Support new features and innovations



Show complexity trends

- Code gets worse over time
- Which will slow us down



Coordination bottlenecks

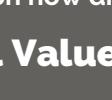
- Add people side to the presentation

Beyond Conway's Law



Quality Suffers with Parallel Development

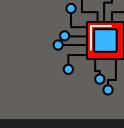
Increases risk of defects with the number of developers



Coordination needs

Number of authors behind each component

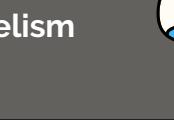
Rank Code by Diffusion



Calculate a fractal value

- How many different authors have contributed
- How the work is distributed among them

0 : Single author
1 : the more contributors there are



1 Color per Author

Module 1



Module 2

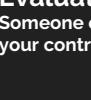
Module 1 : Many minor contributors
Higher risk for defects

RISK

Module 2 : 1 main developer
Reduced risks

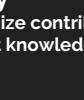
"Ranks all the modules in our codebase based on how diffused the development effort is"

Use Fractal Values to



Prioritize code reviews

Done right - a proven defect-removal



Focus tests

Identify the areas to focus extra tests



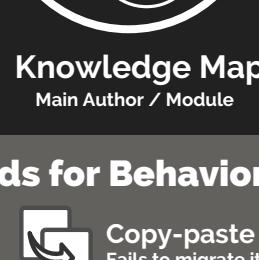
Replan suggested features

If high developer congestion

Lead by example
Model the behaviors you want to see in others



Redesign for increased parallelism
Candidate for splinter refactorings ?



Use Social Data

Evaluation
Someone else cares about your contribution

Gogogo...

Visibility
• Recognize contributions
• Present knowledge maps

Small Groups

Guide On and Off-boarding

Lead by example
Model the behaviors you want to see in others

Gogogo...

Visibility
• Recognize contributions
• Present knowledge maps

Identify the Experts
Find out who to communicate with

Measure Future Knowledge Loss

React to Knowledge Loss
Focus to maintain knowledge

Small Groups

Knowledge Map
Main Author / Module

TEAM TOPOLOGIES

by Matthew Skelton and Manuel Pais

TEAM AS THE MEANS OF DELIVERY



Team assignments
First draft of the architecture



Inverse Conway manoeuvre
Organize teams to match the architecture you want



- Not all communication / collaboration is good
- Restrict communication between teams
- Focus communication between specific teams

"Disbanding high-performing teams is worse than vandalism: it is corporate psychopathy."

— Allan Kelly, Project Myopia

TEAM FIRST-THINKING

S-9

Dunbar's number

Seven-to-nine MAX
> Trust will break down



Use Small, Long-Lived Teams

As the Standard
Autonomous



Owns the Software

"Continuity of care"
No shared ownership

Minimize Team Cognitive Load

Total amount of mental effort used in the working memory
Use good boundaries



Embrace Diversity

Produce more creative solutions

Reward the Whole Team

Not individuals

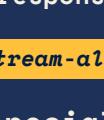


TEAM TOPOLOGIES THAT WORK FOR FAST FLOW

STREAM-ALIGNED TEAM

Team aligned to a single
valuable business
stream of work

Product or service



User Journey

Set of features

User Journey

User Persona



Primary type in an
organization
(80/90 %)

- Work on the full spectrum of delivery
- Requires clarity of purpose and responsibility

"Purpose of the other fundamental team topologies is to reduce the burden on the stream-aligned teams."

ENABLING TEAM

Help stream-aligned teams acquire
missing capabilities

HIRE SPECIALIST



Composed of specialists

In a given technical or product domain



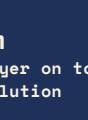
Collaborative nature

Focus on stream-aligned teams problems first
Not the solutions per se

"Do not exist to fix problems that arise from poor practices, prioritization choices, or code quality within stream-aligned teams."

COMPLICATED SUBSYSTEM TEAM

Reduce cognitive load of
stream-aligned teams that needs to
use the complicated subsystem



Responsible for building / maintaining

A part of the system

That depends heavily on specialist knowledge

Examples : Video processing codec, Mathematical model, Real-time trade, Reconciliation algorithm, Face-recognition, ...

"Prioritizes and delivers upcoming work [...] respecting the needs of the stream-aligned teams that use the complicated subsystem."

PLATFORM TEAM

Provide internal services to reduce
cognitive load of stream-aligned teams



Treat services as products

Reliable / Usable

Fit for purpose



Provision new server instance

Provide tools for access management

Thick platform

Combination of several inner platform teams

Providing a myriad of services



Thin platform
Could simply be a layer on top of
a vendor-provided solution



Responsible for building / maintaining

"A digital platform is a foundation of self-service APIs, tools, services, knowledge and support which are arranged as a compelling internal product."

Convert Common Team Types to the Fundamental Team Topologies

"Most organizations would see major gains in
effectiveness by mapping each of their teams
to one of the four fundamental topologies [...]
to adopt the purpose and behavior patterns
of that topology."



PLATFORM TEAM



PLATFORM TEAM



ENABLING TEAM

Or

PLATFORM TEAM



ENABLING TEAM

Or

PLATFORM TEAM

Component Teams

Or other

Architecture

Part time

Infrastructure Teams

Tooling Teams

Regulatory Compliance

Team Location

Risk

Change Cadence

Business Domain Bounded Context

Technology

Performance Isolation

Business Domain Bounded Context

Regulatory Compliance

Team Location

Risk

Performance Isolation

Business Domain Bounded Context

The Good Life

Ce que nous apprend la plus longue étude scientifique sur le bonheur et la santé

1938

2 générations

Une étude "longitudinale"
examiner des vies à travers le temps1300 descendants des
724 participants initiaux"Prospective"
interroger les participants sur leur vie telle qu'elle est

L'étude de Harvard sur le développement des adultes



Identifier ce qui compte pour la santé et le bonheur

- Quels investissements en valaient vraiment la peine ?
- Ce qui maintient les personnes heureuses et en bonne santé ?

Questionnaires

- Qu'est-ce qui compte pour cette personne en particulier ?
- Qu'est-ce qui donne un sens à ses journées ?
- Qu'avait-elle appris de ses expériences ?
- Que regrettait-elle ?

Entretiens attachés

- Étudier la façon dont les participants parlent l'un de l'autre
- Signaux non verbaux

Examiner leur bien-être

- Scanners cérébraux
- Analyses de sang
- Echantillons capillaires
- Poids
- Activité physique
- -

Autres données

- Nature de leur emploi
- Nombre d'amis proches ...

Qu'est-ce qu'une vie réussie ?



1 vie réussie c'est 1 vie compliquée

- Pour tout le monde
- Se forge à partir de ce qui la rend difficile



Le secret = la qualité des relations

Permettent de vivre plus heureux et en meilleure santé

Expérience de prévision affective

Imaginer un état émotionnel dans une situation future

Parler à un inconnu

vs

Rester dans son coin



Meilleur trajet en parlant à un inconnu

Des inconnus dans un train

De l'importance des relations

Mauvais en prévision

- Éviter les complications de la relation à autrui
- Surestimer les complications
- Sous-estimer les effets bénéfiques du lien humain

Culture : prédicteur de bonheur ?

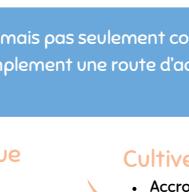
- Des injonctions culturelles permanentes
- Ex: l'argent est le fondement d'une vie réussie

Parfois, les pratiques et les messages culturels nous éloignent du bien-être et du bonheur

Liens sociaux forts et espérance de vie

148 études examinées

Pays du monde entier



Taux de mortalité des personnes ayant le moins de liens sociaux

♂ 2,3 ♀ 2,8

fois plus élevé que celui des personnes en ayant le plus

Espérance de vie > 10 à 15 ans

Hauts revenus

75 000\$ / an chiffre pivot

Chiffre pivot dépassé

- L'argent en plus
- Pas important pour atteindre le bonheur

L'argent fait-il le bonheur?

Etude de Angus Deaton et Daniel Kahneman

Etats-Unis en 2010

Pas la bonne question

Qu'est-ce qui me rend réellement heureux ?

Les relations et les virages de la vie

La vie est chaotique



Cultiver de bonnes relations



- Accroît le côté positif de ce chaos
- Augmente ainsi les chances de faire des rencontres bénéfiques

Milieu de vie = point d'inflexion

Entre 1 mode de vie égocentrique
Replié sur soi-mêmeEt un mode de vie plus généreux
Et tourné vers l'extérieur

Questionnements

Au milieu de notre vie



Est-ce que je m'en sors bien par rapport aux autres ?

Ai-je de bonnes relations avec mes enfants ?

Combien d'années me reste-t-il ?

Ma vie a-t-elle un sens, pour moi mais aussi pour le monde qui m'entoure ?

Quelles personnes et quelles objectifs me tiennent vraiment à cœur ?

Qu'est-ce que je voudrais encore faire d'autre ?

À la fin de notre vie

"Qu'aimeriez-vous avoir moins fait et au contraire davantage fait ?"

"J'aurais aimé ne pas avoir perdu autant de temps."

"J'aurais aimé ne pas avoir tant tergiversé."

"J'aurais aimé ne pas m'inquiéter autant."

"J'aurais aimé passer plus de temps avec ma famille."

Les adultes les plus heureux, ceux qui avaient réussi à transformer la question...

"Que puis-je faire pour moi ?"



"Que puis-je faire pour le monde qui m'entoure ?"

Notre propre étude de Harvard



"À quoi pensiez-vous alors ?"

"Qu'est-ce qui vous inquiétait ?"

"Qu'est-ce que vous abordiez avec confiance ?"

"Quels étaient vos projets ?"

"Avec qui passiez-vous votre temps ?"

"Qu'est-ce qui était le plus important pour vous ?"

"Quand vous pensez à cette époque, que regrettiez-vous ?"

1) Trouvez une photo de vous

À peu près la moitié de l'âge que vous avez aujourd'hui

2) Regardez-vous de près

- Essayez de vous replacer au moment où la photo a été prise
- Regardez vraiment : passez plusieurs minutes s'en imprégner

Prendre du recul de temps en temps.

Votre observatoire social

1) Qui fait partie de ma vie ?

Qui sont mes amis et parents les plus proches ?

Stimulant

Anime et vous exalte

Donne un sentiment de lien

Permet de se sentir mieux que si vous étiez seul

Rare

Ted

Fréquent

3) Allez du haut vers le bas

Haut

Réfléchir comment consolider / favoriser leurs aspects positifs

Dites à ces personnes combien vous les appréciez / pourquoi

Bas

Comment les tirer vers le haut ?

Épuisant

Provoque des tensions

Agacement ou de l'anxiété

La forme sociale - Comment muscler sa sociabilité ?

Les ravages de la solitude



Ses causes

- Une expérience subjective
 - Ce qui cause un sentiment de solitude chez un individu
 - Peut n'avoir aucun effet sur un autre
- Besoin d'amour, de lien et d'un sentiment d'appartenance
 - Comment répondre à ces besoins ?
 - Dans des environnements sociaux compliqués

La solitude augmente notre risque de mortalité autant que le tabagisme ou l'obésité.

Investir du temps / énergie sur nos relations



Le pouvoir de la générosité

Donner ce qu'on aimerait recevoir



La curiosité radicale

- Permet aux autres de se sentir compris et reconnus
- Crée un environnement bienveillant
- Poser une question, écouter la réponse, voir où cela mène

Spirales ascendantes

Chaque jour un peu plus d'attention



1) S'engager dans une communication active avec les autres

Les gens qui s'engagent sont plus heureux que les "passifs"

2) Après avoir surfé, sondez votre humeur

- Avez-vous l'impression d'avoir fait le plein d'énergie ?
- Vous sentez-vous éprouvé après un long voyage à travers Internet ?

Arrêtez de cliquer, discutez

3) Comment vos proches perçoivent votre utilisation ?

• Que pense votre conjoint de la façon dont vous utilisez votre téléphone ?

• Vos habitudes en ligne l'affecte-t-il (ou elle) ?

• Qu'en est-il pour vos enfants ?

4) Prenez du temps sans écran

• Éteindre votre smartphone, votre ordinateur

• Pour vous révéler comment la technologie vous affecte

"Nous avons un avantage crucial sur tous les géants de la technologie : la conquête de notre attention se déroule sur notre propre terrain, littéralement dans notre esprit. Et c'est là que nous pouvons gagner la guerre."

Braver la tempête - s'adapter aux défis relationnels

Modèle W.I.S.E.R

• ralentir ses réactions dans n'importe quelle situation émotionnelle

• les examiner au microscope

W Observer

I Interpréter

S Choisir

E S'engager

R Réfléchir



• Notre "réalité" : pas forcément celle des autres

• Essayer de comprendre les enjeux

• Qu'est-ce qui m'échappe, ici ?

• S'engager dans la stratégie que vous avez choisie

• Soigner sa mise en œuvre

• Peut être utile de s'entraîner

Le contact aimant l'équivalent d'un

Expérience de Coan (IRM)

- Tenir la main d'un proche
- Réduit l'activité des centres de la peur
- Diminue l'anxiété
- Réduit l'intensité de la douleur

Flagrant délit de gentillesse

Dernière chose pour laquelle vous lui êtes reconnaissant ?

Remarquer les petits gestes / s'en souvenir peut avoir un effet positif

Tous les amis ont leurs avantages

L'amitié diminue notre perception de l'adversité :

- Épreuves apparaissent moins stressantes que si nous étions seuls
- Diminue l'impact et la durée d'un stress extrême

Ne

THE SOFTWARE CRAFTSMAN

BY SANDRO MANCUSO

WHAT ?

NOT A RELIGION

NOT A METHOD

WORKING CODE = THE MINIMUM FOR A PROFESSIONAL

GOOD SENIOR DEVELOPER CODE

80'S

NOW

NOBODY UNDERSTAND
THE CODE

- CLEAN
- HUMAN READABLE
- DOMAIN LANGUAGE

"CRAFTSMANSHIP OVER CRAP" - ROBERT C. MARTIN

MINDSET ?

BE PROUD TO BE A DEVELOPER

DEVELOPMENT IS A CRAFT



LEARNING FROM OTHERS

CONSTANTLY SHARING

A LONG JOURNEY TO MASTERY

CARING ABOUT WHAT THEY DO

RESPONSIBILITY / PROFESSIONALISM / PRAGMATISM / PRIDE

IDEOLOGY

LOWER THE COST OF QUALITY

WHAT MODERN DEVELOPERS DO

- DEVELOP
- TEST
- ANALYZE
- MAKE TECHNICAL CHOICES
- HELP CLIENT
- RECRUIT
- ...



AGILITY
HOW TO BUILD THE RIGHT THING

FOCUS ON THE PROCESS CUSTOMER CENTRIC
DOES NOT MAKE DEVELOPERS BETTER

CRAFTSMANSHIP
HOW TO BUILD THE THING RIGHT

FAILURE

ADVANCEMENT

SUCCESS

SUCCESS
ADVANCEMENT

ADVANCEMENT

MANIFESTO FOR SOFTWARE CRAFTSMANSHIP - 2008

1 NOT ONLY WORKING SOFTWARE, BUT ALSO WELL-CRAFTED SOFTWARE

WELL-CRAFTED = HIGH QUALITY CODE

- AUTOMATED TESTS
- BUSINESS LANGUAGE IN THE CODE
- SIMPLE DESIGN



"CODE QUALITY IS NOT A GUARANTEE OF SUCCESS
BUT CAN BE THE MAIN CAUSE OF FAILURE"

2 NOT ONLY RESPONDING TO CHANGE, BUT ALSO STEADILY ADDING VALUE

CONSTANTLY IMPROVE YOUR CODE

- TESTABLE
- EXTENDABLE
- REFACTOR



BOY SCOUT RULE

"ALWAYS LEAVE THE CAMPGROUND CLEANER THAN
YOU FOUND IT."



3 NOT ONLY INDIVIDUALS AND INTERACTIONS, BUT ALSO A COMMUNITY OF PROFESSIONALS

SHARE / MENTOR

- KNOWLEDGE
- IDEAS
- SUCCESSES AND FAILURES



CRAFTSMEN WANT TO WORK WITH
PASSIONATES & INSPIRING PROFESSIONALS
A.K.A OTHER CRAFTSMEN

4 NOT ONLY CUSTOMER COLLABORATION, BUT ALSO PRODUCTIVE PARTNERSHIPS

WE ARE NOT FACTORY WORKERS

- MUST HELP OUR CLIENTS
- MUST SAY NO FOR CLIENTS GOOD



SOME CLIENTS ARE NOT READY :
VERY DIFFICULT ENVIRONMENT FOR CRAFTSMEN

REDUCE THE GAP BETWEEN THE AGILE METHODOLOGIES AND THE TECHNICAL WORLD

ATTITUDE

PRACTICE THROUGH



CODE KATAS

OPEN SOURCE PROJECTS



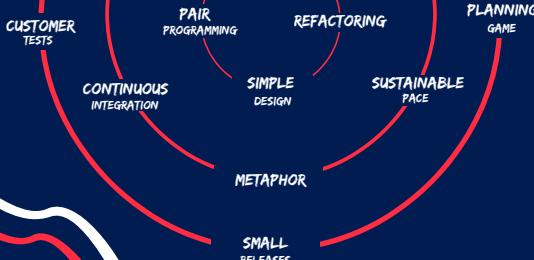
PAIR/MOB
PROGRAMMING

DISCOVERY

PRACTICES

EXTREME PROGRAMMING

WHOLE TEAM



INJECT PASSION

CREATE A
CULTURE OF IMPROVEMENT

IMPROVE

BOOK CLUB

LIGHTNING TALKS

BROWN BAGS

LEAN COFFEES

CODE REVIEWS

COMMUNITIES OF PRACTICE

POSITIVE
MOTIVATION TOWARDS A GOAL

"MAKE THIS PROJECT SUCCESSFULLY
AND YOU GET A BONUS"

"I REALLY WANT TO WORK ON
THIS PROJECT"

"DO THIS PROJECT OR
YOU ARE FIRED"

"I REALLY DON'T WANT TO WORK
ON THIS PROJECT"

NEGATIVE
MOTIVATION AWAY FROM SOMETHING

INTRINSIC
YOU WANT TO DO IT

CREATE A
CULTURE OF SHARING

CREATE A
CULTURE OF SHARING

IMPROVE

BROWN BAGS

LEAN COFFEES

CODE REVIEWS

COMMUNITIES OF PRACTICE

CREATE A
CULTURE OF SHARING

IMPROVE

BROWN BAGS

LEAN COFFEES

CODE REVIEWS

COMMUNITIES OF PRACTICE

CREATE A
CULTURE OF SHARING

IMPROVE

BROWN BAGS

LEAN COFFEES

CODE REVIEWS

COMMUNITIES OF PRACTICE

CREATE A
CULTURE OF SHARING

IMPROVE

BROWN BAGS

LEAN COFFEES

CODE REVIEWS

COMMUNITIES OF PRACTICE

CREATE A
CULTURE OF SHARING

IMPROVE

BROWN BAGS

LEAN COFFEES

CODE REVIEWS

COMMUNITIES OF PRACTICE

CREATE A
CULTURE OF SHARING

IMPROVE

BROWN BAGS

LEAN COFFEES

CODE REVIEWS

COMMUNITIES OF PRACTICE

CREATE A
CULTURE OF SHARING

IMPROVE

BROWN BAGS

LEAN COFFEES

CODE REVIEWS

COMMUNITIES OF PRACTICE

CREATE A
CULTURE OF SHARING

IMPROVE

BROWN BAGS

LEAN COFFEES

CODE REVIEWS

COMMUNITIES OF PRACTICE

CREATE A
CULTURE OF SHARING

IMPROVE

BROWN BAGS

LEAN COFFEES

CODE REVIEWS

COMMUNITIES OF PRACTICE

CREATE A
CULTURE OF SHARING

IMPROVE

BROWN BAGS

LEAN COFFEES

CODE REVIEWS

COMMUNITIES OF PRACTICE

CREATE A
CULTURE OF SHARING

IMPROVE

BROWN BAGS

LEAN COFFEES

CODE REVIEWS

COMMUNITIES OF PRACTICE

CREATE A
CULTURE OF SHARING

IMPROVE

BROWN BAGS

LEAN COFFEES

CODE REVIEWS

COMMUNITIES OF PRACTICE

CREATE A
CULTURE OF SHARING

IMPROVE

BROWN BAGS

LEAN COFFEES

CODE REVIEWS

COMMUNITIES OF PRACTICE

CREATE A
CULTURE OF SHARING

IMPROVE

BROWN BAGS

LEAN COFFEES

CODE REVIEWS

COMMUNITIES OF PRACTICE

CREATE A
CULTURE OF SHARING

IMPROVE

BROWN BAGS

LEAN COFFEES

CODE REVIEWS

COMMUNITIES OF PRACTICE

CREATE A
CULTURE OF SHARING

IMPROVE

BROWN BAGS

LEAN COFFEES

CODE REVIEWS

COMMUNITIES OF PRACTICE

CREATE A
CULTURE OF SHARING

IMPROVE

BROWN BAGS

LEAN COFFEES

CODE REVIEWS

COMMUNITIES

TU FAIS QUOI DANS LA VIE ?

par Joséphine Bouchez / Matthieu Dardaillon



80 000 HEURES

Temps moyen de nos vies à travailler



QU'ALLONS-NOUS EN FAIRE ?

L'URGENCE D'AGIR

Notre système actuel n'est pas durable

Fondé sur l'utilisation croissante de ressources naturelles présentes en quantités limitées



- 1970 : 29 décembre
- 1990 : 7 décembre
- 2019 : 29 juillet

26 personnes

Autant d'argent que la moitié la plus pauvre de la population mondiale



1 continent de plastique

Plus de 1.6 million de km² flotte dans l'Océan Pacifique



Chute des populations d'oiseau

Les populations d'oiseau ont chuté d'1/3 en 15 ans



300 000 SDF en France

"Qui pourra assumer face aux générations futures que nous avons cautionné et laissé faire ?"

Notre responsabilité ?

Nous sommes ce système

Chacun un rôle à jouer

Est-ce que je contribue activement à construire la société dans laquelle j'aspire à vivre ?



Quelles sont les causes qui vous touchent ?

Agriculture / alimentation

Permettre à tous de manger sainement

Santé

Permettre l'accès à des soins de qualité pour tous

Habitat

Permettre à chacun de se loger dignement



Education

Permettre une éducation de qualité et accessible

Energies

Permettre l'accès à une énergie propre

Inclusion & lien social

Permettre à chacun de trouver une place dans la société

Environnement

Préserver l'environnement, la nature, la biodiversité

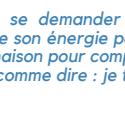
...

LE LEVIER DU TRAVAIL

"Nous avons besoin de tous les métiers dans tous les secteurs"

Travail
Semaine
Pour gagner sa vie

L'engagement
Soir / Week-end
"Quand on a le temps"



réconcilier les 2 avec

Des vies scindées

Intention

"J'ai choisi cette activité dans le but d'être utile à la société"



Impact

"les conséquences de mon travail ont un impact positif sur la société"

Engagement

"le temps que je dédie à ces activités représente au minimum 50% de mon temps et de ma rémunération"



Décisions

Critères d'impact (social, environnemental) ont au moins autant de poids que les critères économiques

LES CARRIÈRES À IMPACT



Baisse MAX de notre empreinte carbone individuel

Si chaque français adopte un comportement héroïque

25%



Etat + Entreprises



NOUS SOMMES AUSSI L'ETAT ET LES ENTREPRISES

CHANGER LE SYSTÈME



Education / formation

- Pas inviter à identifier nos talents
- Comment les mettre au service des enjeux de société ?



L'économie

- L'économie dirige le monde
- Possible de créer de la valeur :
 - économique, écologique ET sociale



L'emploi

- 85% des emplois de 2030 n'existent pas encore
- à nous de les créer

L'orientation

- Pas encouragés à nous engager dans des carrières à impact
- Choix d'orientation = souvent fait par défaut

Quid de valoriser l'utilité sociétal et l'épanouissement ?

"Ne doutez jamais qu'un petit groupe d'individus puisse changer le monde. En réalité c'est toujours comme cela qu'il a changé"
- Margaret Mead

Transformer votre emploi actuel



Réduire les impacts négatifs de votre activité



Connectez-vous à d'autres pionniers



Formez vous



Transformez pas à pas

Montrer la valeur que cela apporte à votre organisation

La rémunération

- Quels sont mes besoins ?
- De combien ai-je besoin pour être heureux aujourd'hui et préparer un futur désirable ?
- Qu'est ce qui me paraît juste par rapport à la structure ?

#sharingiscaring

PASSER À L'ACTION

Le plus ce n'est pas d'avoir envie, c'est de sauter le pas



3 stratégies



Trouver un nouvel emploi

- Identifiez vos talents / compétences
- Comment les utiliser différemment ?



Créer votre emploi

Quelque soit la thématique Trouvez un problème de société urgent, important, qui n'a pas d'alternative satisfaisante



Entourez-vous

Essentiel de travailler avec les bonnes personnes

Soyez obstinés par votre problème pour trouver la meilleure solution possible

@yot88

Yoan THIRION

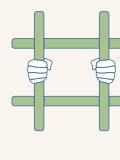
UNE VIE SUR NOTRE PLANÈTE

David Attenborough



Déclin accéléré de la biodiversité

Véritable tragédie de notre temps



Nous sommes tous coupables

- "Ce n'est pas notre faute"
- Nous sommes nés dans un monde humain qui n'est pas durable

Continuer

De vivre notre existence heureuse en ignorant la catastrophe à nos portes



Changer

Nous devons faire 1 choix



Encore temps d'arrêter le réacteur

Il existe une alternative viable

MON TÉMOIGNAGE



	en milliards	en Parties Par Million de Molécules d'air	monde sauvage subsistant	Observations
1937	2,3	280	66%	L'agriculture a changé notre rapport entre l'humanité et la nature. Apprivoisement d'une partie du monde sauvage.
1954	2,7	310	64%	Émission Zoo Quest Nature sauvage florissait. Personne n'avait conscience des problèmes qui se posaient déjà.
1960	3	315	62%	Comprendre le fonctionnement global de l'écosystème du Serengeti. Histoire d'interdépendance / écologie.
1989	5,1	353	49%	Le monde compte trois trillions d'arbres de moins qu'à début de la civilisation humaine.
1997	5,9	360	46%	L'humanité avait éliminé 90% des gros poissons dans tous les océans. Prétez les poissons au sommet de la chaîne trophique
2011	7	391	39%	Température moyenne de 0,8°C plus chaude qu'en 1926
2020	7,8	415	35%	Notre impact est vraiment mondial...

> 80 millions de tonnes (par an) de fruits de mer prélevés

de fruits de mer prélevés

15 milliards d'arbres abattus par an

2020

$m(\text{humain}) + m(\text{élevage}) = 96\% m(\text{animaux})$

élevage : animaux que nous élevons pour les manger

" Nous avons remplacé le monde sauvage par un monde apprivoisé. Nous considérons la Terre comme NOTRE planète, gouvernée par l'humanité, pour l'humanité. "

CE QUI NOUS ATTEND



Monde du vivant en passe de s'effondrer

a déjà commencé à s'effondrer

Dégénération de la couche d'ozone

Changement climatique

Acidification des océans

Pollution atmosphérique

5 Limites planétaires dépassées

Erosion de la biodiversité

Changement d'utilisation des sols

Pollution chimique

Consommation d'eau

Usage d'engrais

" Nous vivons déjà hors de l'espace de fonctionnement sécurisé de notre planète "



2030

- -75% de la surface de la forêt amazonienne
- Pôle Nord : été libre de glace



2040

- Pergélisol fondu : 1400 GT de carbone stocké
- Glissements de terrains / inondations gigantesques



2050

- Acidité très élevé des océans
- Commencement de la fin pour la pêche



2080

- Engrais : sols stériles et épuisés
- Déclin des espèces d'insectes
- Affecter les 3/4 de nos cultures



Migrations forcées de populations



+0.9 m

du niveau de la mer



+4°C

Température de la Terre

2100



1/4

de l'humanité vivra > 29°C

Fin de la stabilité de l'Holocène (notre jardin d'Eden)

6ème extinction massive

UNE VISION POUR L'AVENIR



Personne

Profit

Planète



Croissance verte

Sans impact négatif sur l'environnement

Monde limité

rien ne peut grandir indéfiniment

Construire 1 modèle économique durable

3 P

Dépasser la croissance



Nous avons tout pris au vivant

sans songer aux dégâts

Ex : Cabo Pulmo

Nous avons tout pris au vivant

sans songer aux dégâts

Si notre principal critère pour juger nos actions est la renaissance du monde naturel nous ne pourrons manquer de prendre les bonnes décisions

Passer à l'énergie propre

Budget carbone

montant réduit de carbone pouvant être rejeté

Mettre fin à notre dépendance aux combustibles fossiles

Elever le prix des émissions au niveau mondial

Taxe carbone



Accélérerait la révolution durable dont nous avons besoin

Réensauvager les mers

Créer des zones interdites à la pêche

Permet aux poissons de devenir plus vieux et plus gros

Pêche durable à long terme

1/3 des océans en zones sans pêches suffirait

Favoriser la pêche durable

Les entreprises la pratiquant

Ex : Cabo Pulmo

Occupier moins d'espace

Une grande partie dépourvue de bétail

80% de la terre agricole consacrée à la production de viande / lait

Culture du soja pour nourrir le bétail

Viandes propres cultures cellulaires

Protéines alternatives

Remédier au gaspillage alimentaire

Produire + en cultivant moins de terre

Agriculture régénératrice revivifier les sols

Agriculture verticale étages de différentes plantes

Recréer des espaces sauvages

Ramènerait la biodiversité

Stabiliser la planète

Vers une vie plus harmonieuse

NOTRE PLUS GRANDE CHANCE

De l'Holocène à l'Anthropocène

Ere des êtres humains

Nous devons nous sauver nous-mêmes

Nous parlons souvent de sauver la planète

Avec ou sans nous, la vie sauvage reprendra ses droits

“ Nous sommes les premiers à comprendre vraiment le problème et les derniers à pouvoir encore y remédier ”

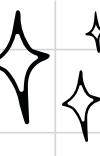
#sharingiscaring

by Yoan THIRION

@yot88

Unit Testing

Principles, Practices, and Patterns



by Vladimir Khorikov

Goal of Unit testing



Project without tests

- Quickly slows down
- Hard to make any progress



What makes a successful test suite?



- Integrated into the development cycle
- Targets most important parts of the code base
- Provides maximum value
 - With minimum maintenance costs

A tool that provides insurance against a vast majority of regressions

Not all tests are created equal



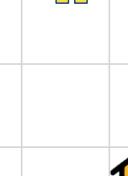
Bad tests : raise false alarms



- Unit tests are vulnerable to bugs
- Require maintenance

Tests are code too

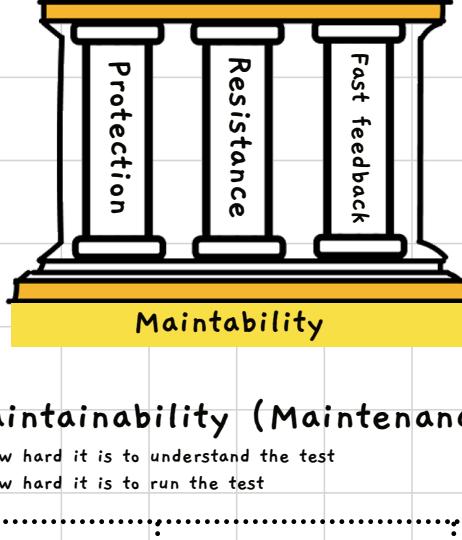
View them as part of your code base that aims at solving a particular problem: ensuring the application's correctness



Automated test that :

- Verifies a small piece of code (also known as a unit)
- Does it quickly
- And does it in an isolated manner.

What is a Unit Test ?



Protection against regressions

- A regression = a software bug
- The larger the code base → the more exposure to potential bugs
- Tests should reveal those regressions

Resistance to refactoring

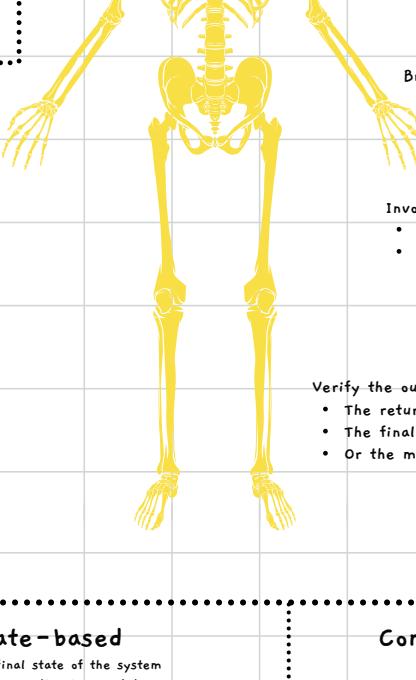
The degree to which a test can sustain a refactoring of the underlying application code without turning red (failing)

Fast feedback

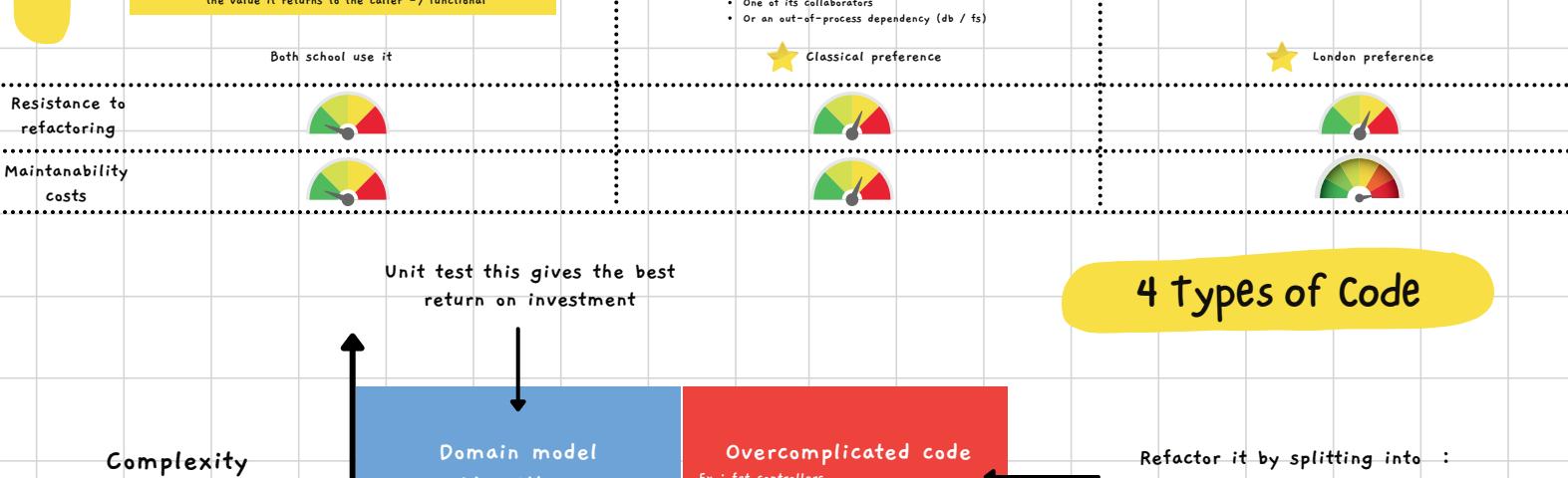
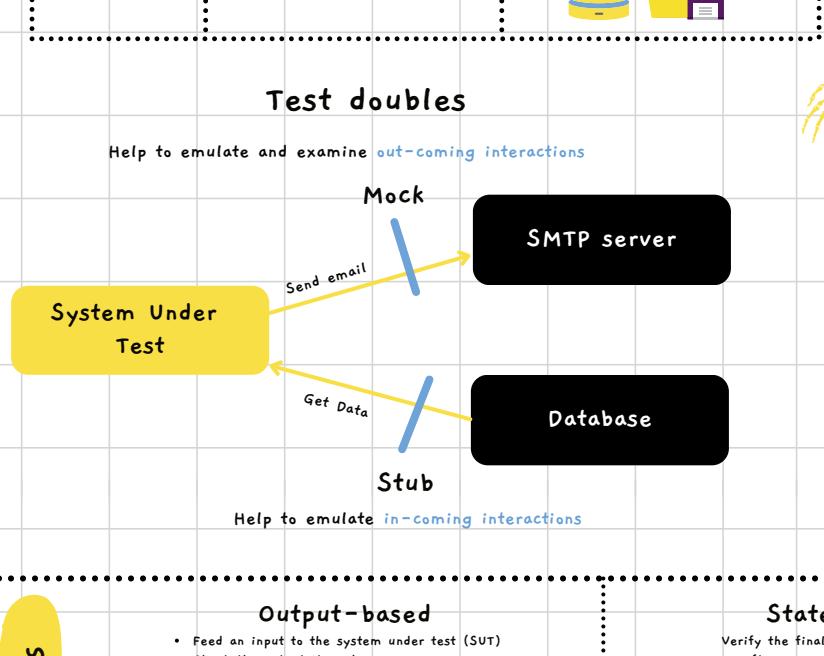
The more of them you can :

- Have in the suite
- Run them → shorten the feedback loop

Anatomy



3 Styles of tests



4 Types of Code

LEADERSHIP STRATEGY and TACTICS

by Jocko Willink

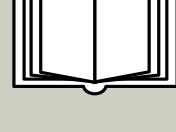
"A good leader has nothing to prove, but everything to prove."

STRATEGIES



Detach

Mentally from the problem



Humility

Always learn



Leaders tell the TRUTH

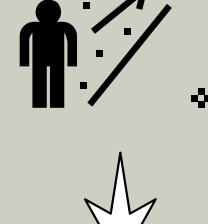


Control Yourself

Don't overreact



Earn Influence & RESPECT



Self Discipline



No Yes-men

Favor challenging people



Pride

Drives positive behavior

SKILLS to be a Good Leader



Simple Communication
Confidence
Charisma
Read People

Acknowledge Strengths/Weaknesses

The Power of Relationships

basis of all good leadership



HOW TO SUCCEED AS A NEW LEADER ?

BEHAVIORS

Take Ownership

Of failures and mistakes



Get the Job DONE

Of failures and mistakes

Pass Credit

For success up and down

Treat People with Respect

Take care of them / will take care of you.

SELF-BEING

Build

Build trust



Listen

Ask for advice and heed it

Don't Act Like you Know Everything

You don't... Ask smart questions

RELATIONSHIPS

Be Balanced

Extreme actions / opinions
are not good.



Work Hard

Work harder than anyone

Be Decisive

When it is time to make a decision
make one

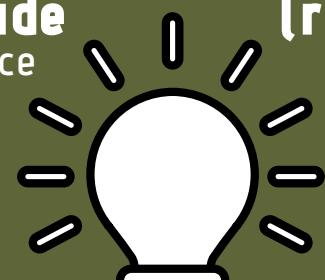


Be Humble

An honor to be in a
leadership position

Have Integrity

Do what you say; say what you do.



(re)Building Confidence

Fixing negative attitude
Maybe not at the right place

Building high-level
team players

Put junior in charge

Teaching Humility
Fix overconfidence

by Yoan THIRION

RÉALISER SES RÊVES CA S'APPREND !

PAR THOMAS GIBOT



CONVICTION

Réaliser ses rêves est une compétence s'acquiert et se développe

Manifestation de ce que nous sommes vraiment

- au plus profond de nous-mêmes
- détachée de notre ego / peurs
- représente notre essence véritable



Le rêve comme une stratégie (un moyen)

- Pourquoi est-il si important pour moi ?
- Qu'est-ce qu'il me donne l'occasion de vivre, de ressentir ?

Tor des Géants

330 km 24 000 D+

"Je rêvais mon rêve"

En parler > agir pour lui
Evoquer mon rêve = le vivre

Evidence PHASE DU KIF

Fluidité Joie Energie



Comment vais-je m'organiser pour faire avancer mon objectif ?

Réfléchir au processus

"Faire de mon rêve un projet"

PHASE DU TAF

Besogneux

"Ce qui nous fait passer de quelqu'un qui rêve ses rêves à quelqu'un qui se donne une chance de les vivre."

"JE NE SAIS PAS RÊVER !"

SE PRÉPARER POUR SES RÊVES

CRÉER SON SYSTÈME EST LE MOYEN D'ACCOUCHER DE NOS RÊVES.

ASTRONOME DU RÊVE

Sommel Courir Matériel Pour l'autre Pour soi

Maximiser nos chances de réaliser nos rêves !

- Augmenter notre niveau de discernement
- Identifier leviers pour nous permettre de le réaliser

"Ma conviction est qu'en améliorant « le Faire », nous avançons fort.

En ayant une plus grande conscience de nous-mêmes, de « l'Être », nous avançons juste."

LA CONSTELLATION DU RÊVE

INTITULÉ DE NOTRE RÊVE
ou centre de la constellation



Vision juste de ce qu'il représente et implique pour nous



Tout ce qui pourrait nous permettre de changer notre rapport à nos enjeux

inspirations extérieures

TED

Livres

Podcasts

Vidéos

Formations

problèmes besoins questionnements

ENJEUX

composent chacun des axes

ACTIONS

Identifier pour chacun des enjeux :

- actions concrètes / réalisables
- responsables = moi

COMBATTRE LE JUGE INTÉRIEUR

- Cette voix qui nous gâche la vie
- "Je suis nul, les autres sont tellement meilleurs que moi..."

NOUS MAINTIENIR DANS UN STATU QUO

sa mission principale

NOUS MAINTIENIR DANS UN MALHEUR ACCEPTABLE

ne jamais sortir de sa zone de confort

OSER VIVRE LA VIE QUI NOUS INSPIRE VRAIMENT :

Etape 1 : Identifier les enjeux

- Identifier les déclencheurs de votre juge intérieur / ses jugements
- Ex : Evocuer mon rêve à des trailers très expérimentés
- Attributs : "Je suis préférentiel..."

Etape 2 : Identifier mes schémas habituels de résistance

- Identifier les schémas de résistance qui concernent mon juge intérieur
- Je m'interroge : "je me souviens à mon juge intérieur"

Etape 3 : Identifier les attributs

- L'exploration : explorer les jugements jusqu'à les rendre réalisables
- L'attribution : décider consciemment de ne pas les écouter
- L'aggrégation : regrouper les attributs
- La pose à l'épreuve : je pose à la vie : "je m'en fous" et de voir si l'attribution est réalisable

"Si tes rêves ne te font pas peur, c'est qu'ils ne sont pas assez grands." - Mike Horn ,

Est-ce que les autres vont trouver ça bien?
Vais-je être valorisé et reconnu en poursuivant ce rêve?

Comment tous ce que l'entreprise est-il un moyen de me prouver que je m'engage pour ce qui est important pour moi ?

MA RECONNAISSANCE PLUTÔT QUE LA RECONNAISSANCE

Comment toutes ces personnes sont-elles pour moi ?

Est-ce que les autres vont trouver ça bien?

Vais-je être valorisé et reconnu en poursuivant ce rêve?

Comment toutes ces personnes sont-elles pour moi ?

Est-ce que les autres vont trouver ça bien?

Vais-je être valorisé et reconnu en poursuivant ce rêve?

Comment toutes ces personnes sont-elles pour moi ?

Est-ce que les autres vont trouver ça bien?

Vais-je être valorisé et reconnu en poursuivant ce rêve?

Comment toutes ces personnes sont-elles pour moi ?

Est-ce que les autres vont trouver ça bien?

Vais-je être valorisé et reconnu en poursuivant ce rêve?

Comment toutes ces personnes sont-elles pour moi ?

Est-ce que les autres vont trouver ça bien?

Vais-je être valorisé et reconnu en poursuivant ce rêve?

Comment toutes ces personnes sont-elles pour moi ?

Est-ce que les autres vont trouver ça bien?

Vais-je être valorisé et reconnu en poursuivant ce rêve?

Comment toutes ces personnes sont-elles pour moi ?

Est-ce que les autres vont trouver ça bien?

Vais-je être valorisé et reconnu en poursuivant ce rêve?

Comment toutes ces personnes sont-elles pour moi ?

Est-ce que les autres vont trouver ça bien?

Vais-je être valorisé et reconnu en poursuivant ce rêve?

Comment toutes ces personnes sont-elles pour moi ?

Est-ce que les autres vont trouver ça bien?

Vais-je être valorisé et reconnu en poursuivant ce rêve?

Comment toutes ces personnes sont-elles pour moi ?

Est-ce que les autres vont trouver ça bien?

Vais-je être valorisé et reconnu en poursuivant ce rêve?

Comment toutes ces personnes sont-elles pour moi ?

Est-ce que les autres vont trouver ça bien?

Vais-je être valorisé et reconnu en poursuivant ce rêve?

Comment toutes ces personnes sont-elles pour moi ?

Est-ce que les autres vont trouver ça bien?

Vais-je être valorisé et reconnu en poursuivant ce rêve?

Comment toutes ces personnes sont-elles pour moi ?

Est-ce que les autres vont trouver ça bien?

Vais-je être valorisé et reconnu en poursuivant ce rêve?

Comment toutes ces personnes sont-elles pour moi ?

Est-ce que les autres vont trouver ça bien?

Vais-je être valorisé et reconnu en poursuivant ce rêve?

Comment toutes ces personnes sont-elles pour moi ?

Est-ce que les autres vont trouver ça bien?

Vais-je être valorisé et reconnu en poursuivant ce rêve?

Comment toutes ces personnes sont-elles pour moi ?

Est-ce que les autres vont trouver ça bien?

Vais-je être valorisé et reconnu en poursuivant ce rêve?

Comment toutes ces personnes sont-elles pour moi ?

Est-ce que les autres vont trouver ça bien?

Vais-je être valorisé et reconnu en poursuivant ce rêve?

Comment toutes ces personnes sont-elles pour moi ?

Est-ce que les autres vont trouver ça bien?

Vais-je être valorisé et reconnu en poursuivant ce rêve?

Comment toutes ces personnes sont-elles pour moi ?

Est-ce que les autres vont trouver ça bien?

Vais-je être valorisé et reconnu en poursuivant ce rêve?

Comment toutes ces personnes sont-elles pour moi ?

Est-ce que les autres vont trouver ça bien?

Vais-je être valorisé et reconnu en poursuivant ce rêve?

Comment toutes ces personnes sont-elles pour moi ?

Est-ce que les autres vont trouver ça bien?

Vais-je être valorisé et reconnu en poursuivant ce rêve?

Comment toutes ces personnes sont-elles pour moi ?

Est-ce que les autres vont trouver ça bien?

Vais-je être valorisé et reconnu en poursuivant ce rêve?

Comment toutes ces personnes sont-elles pour moi ?

Est-ce que les autres vont trouver ça bien?

Vais-je être valorisé et reconnu en poursuivant ce rêve?

Comment toutes ces personnes sont-elles pour moi ?

Est-ce que les autres vont trouver ça bien?

Vais-je être valorisé et reconnu en poursuivant ce rêve?

Comment toutes ces personnes sont-elles pour moi ?

Est-ce que les autres vont trouver ça bien?

Vais-je être valorisé et reconnu en poursuivant ce rêve?

Comment toutes ces personnes sont-elles pour moi ?

Est-ce que les autres vont trouver ça bien?

Vais-je être valorisé et reconnu en poursuivant ce rêve?

Comment toutes ces personnes sont-elles pour moi ?

Est-ce que les autres vont trouver ça bien?

Vais-je être valorisé et reconnu en poursuivant ce rêve?

Comment toutes ces personnes sont-elles pour moi ?

Est-ce que les autres vont trouver ça bien?

Vais-je être valorisé et reconnu en poursuivant ce rêve?

Comment toutes ces personnes sont-elles pour moi ?

Est-ce que les autres vont trouver ça bien?

Vais-je être valorisé et reconnu en poursuivant ce rêve?

Comment toutes ces personnes sont-elles pour moi ?

Est-ce que les autres vont trouver ça bien?

Vais-je être valorisé et reconnu en poursuivant ce rêve?

Comment toutes ces personnes sont-elles pour moi ?

Est-ce que les autres vont trouver ça bien?

Vais-je être valorisé et reconnu en poursuivant ce rêve?

Comment toutes ces personnes sont-elles pour moi ?

Est-ce que les autres vont trouver ça bien?

Vais-je être valorisé et reconnu en poursuivant ce rêve?

Comment toutes ces personnes sont-elles pour moi ?

Est-ce que les autres vont trouver ça bien?

Vais-je être valorisé et reconnu en poursuivant ce rêve?

Comment toutes ces personnes sont-elles pour moi ?

Est-ce que les autres vont trouver ça bien?

Vais-je être valorisé et reconnu

SUCCEEDING WITH OKRS IN AGILE

BY ALLAN KELLY <https://www.allankellyassociates.co.uk/>



OBJECTIVES



AVOID BOXING YOURSELF
INTO A SPECIFIC APPROACH OR SOLUTION



MAKE THE VALUE THAT BRINGS OBVIOUS
SO THAT...

RETOOL THE DELIVERY PIPELINE TO FACILITATE CONTINUOUS DELIVERY



INCREASE ROI BY REDUCING TIME TO MARKET WITH
A NEW DELIVERY PIPELINE AND CONTINUOUS DELIVERY PRACTICES

KEY RESULTS



FIGHT AGAINST DOMINOS

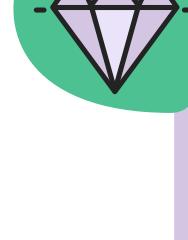
DON'T ACCEPT DEPENDENCIES

SMALLER GOALS THAT BUILD TOWARDS THE OBJECTIVE



EACH ONE MUST DELIVER VALUE

ALL ABOUT DELIVERING OUTCOMES THAT ADD VALUE



KEY RESULTS TRICKS



EXPERIMENTS

SAFER FOR THE TEAM TO TAKE ON RISK

SUCCESS = DOING THE EXPERIMENT ITSELF AND ABSORBING THE LEARNING

USE SURVEY

MAKE CHANGES TO PEOPLE

TAKE SURVEY

TEST IT WITH SURVEY

TIME-BOXES

EXPERIMENT SOMETHING FOR N WEEKS



HYPOTHESIS-DRIVEN DEVELOPMENT

WE BELIEVE <THIS CAPABILITY>
WILL RESULT IN <THIS OUTCOME>
WE WILL HAVE CONFIDENCE TO PROCEED WHEN <WE SEE A MEASURABLE SIGNAL>

"if you aren't failing, you aren't trying"

WHY ?

FILL A NEED AT THE MID-TERM
PLANNING LEVEL



LATER
LOOK MONTHS / YEARS INTO THE FUTURE



SOON : OKRS
LOOK TO THE NEXT FEW MONTHS

NOW : SPRINT PLANNING
FEW WEEKS INTO THE FUTURE



CREATE FOCUS
TELLS YOU WHEN TO STOP



TRUE NORTH

GUIDE AND FIGHT TO STAY ON COURSE
DON'T STICK BLINDLY TO OKRS AS THE WORLD AROUND CHANGES

OKRS ENHANCE COMMUNICATION



- EASIER TO COMMUNICATE WHAT A TEAM IS DOING
- A MEANS OF COMMUNICATING STATUS AND PROGRESS
- SUCCESS MOTIVATES CONTINUATION

HOW TO ?

OBJECTIVE VALUE > Σ (KEY RESULTS VALUES)



BOTTOM UP

DON'T IMPOSE OKRS FROM ABOVE
TEAM RESPONSIBLE FOR SETTING THEIR OWN OKRS AND DELIVERING THEM



LIMIT THEIR NUMBER

3 OBJECTIVES
3 KEY RESULTS PER OBJECTIVE



LEADERS

BUILD PSYCHOLOGICAL SAFETY / MAKE FAILURE AN OPTION
MAKE COMPLETELY CLEAR WHAT THE PRIORITIES ARE



ALL OKRS ARE NOT EQUALS

SOME MIGHT BE HIGHER PRIORITY



WHAT NOT TO DO

EVERYTHING THAT IS NOT IN THE OKRS IS LOWER PRIORITY



STRATEGY

WHAT ARE THE STRATEGIC PRIORITIES FOR THE NEXT QUARTER ?
WHAT DOES THE TEAM AIM TO DO ?
WHAT TARGETS WILL THE TEAM SET FOR ITSELF ?

TEST DRIVEN APPROACH



DECIDE WHAT YOU WANT : OBJECTIVE



SET A SERIES OF ACCEPTANCE CRITERIA : KEY RESULTS

EACH KEY RESULT SHOULD BE MEASURABLE

GET ON AND DEVELOP



DON'T CONSIDER YOURSELF DONE UNTIL

YOU CAN PASS THE TESTS

YOU MEET THE OBJECTIVES

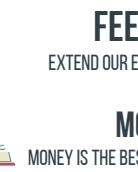
"As with agile, you need to find your own way to OKRs [...] be prepared to experiment."

OKRS AND BACKLOG

BACKLOG FIRST

SUCCESS : BURN DOWN THE BACKLOG

OKRS : ONE OF SEVERAL INPUTS



OKRS FIRST

SUCCESS : DELIVER OKRS

OKRS ARE EVERYTHING

TIMELINE

SET OKRS A FEW WEEKS BEFORE NEXT QUARTER

2 OR 3 SHOULD BE FINE



REVIEW AT THE END OF EACH QUARTER

CULTURE

"if you aren't failing, you aren't trying"

DELIVERY CULTURE

VALUE DELIVERY (WORKING PRODUCTS USED BY CUSTOMERS)

NOT HOURS WORKED, NOT PARTIALLY DONE WORK



SUPPORTIVE CULTURE

PSYCHOLOGICAL SAFETY

FAILURES WILL HAPPEN

IF MONEY ATTACHED

- PEOPLE FEEL COMPELLED TO CHASE 100% SUCCESS
- EASIEST WAY = REDUCE THE TARGET

#SHARINGISCARING