Agile ORLANDO
JULY 24-28 2023
PRESENTED BY Agile Alliance
JOIN US TODAY!
#AGILE2023
Andrés Joaquín, Hiroshi Hiromoto
Applying Technical Practices outside IT
ANDRÉS JOAQUÍN

Systems Engineer. Helps organization and teams as a consultant in Kleer. Collaborates with Argentine public education as a professor at UTN University.

Rosarino ⦿ Argentine ⦿ 16yr in Agile

HIROSHI HIROMOTO

Helps organizations to design more adaptable ecosystems that delivers high value to its customers and employees.

Nikkei ⦿ Peruvian ⦿ Part-time traveler ⦿ 12yr in Agile
WHY?
Organizational Performance

Profitability
Market Share
Customer Satisfaction
Software Delivery Performance

Organizational Performance

Software Delivery Performance

Stability
- Change Failure Rate
- Mean Time To Recovery

Speed
- Deployment Frequency
- Cycle Time

https://dora.dev/
Continuous Delivery Drivers

- Test Automation
- Deployment Automation
- Trunk-Based Development
- Shift Left on Security
- Loosely Coupled Architecture
- Empowered Teams
- Continuous Integration
- Version Control
- Test Data Management
- Monitoring
- Proactive Notifications

**Speed**
- Deployment Frequency
- Cycle Time

**Stability**
- Change Failure Rate
- Mean Time To Recovery
HOW?
The Explorers

- Explore and discover over predicting
- A sort of canvas, so it has a structure that helps you with the discovering.
- Based on the essence of the technical practices and guiding questions

Test Driven Development Explorer

<table>
<thead>
<tr>
<th>1.WHY</th>
<th>Why does team adopt this practice?</th>
</tr>
</thead>
<tbody>
<tr>
<td>TDD helps create products defensively and incrementally with higher quality and faster confidence in face of future changes.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>2. THE ESSENCE</th>
<th>What is the essence of this practice beyond the technical aspect?</th>
</tr>
</thead>
<tbody>
<tr>
<td>- Test First</td>
<td></td>
</tr>
<tr>
<td>- Minimum Viable</td>
<td></td>
</tr>
<tr>
<td>- Refactoring</td>
<td></td>
</tr>
<tr>
<td>Cycle</td>
<td></td>
</tr>
<tr>
<td>- Framework</td>
<td></td>
</tr>
<tr>
<td>- Fast</td>
<td></td>
</tr>
<tr>
<td>- Repeatable</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>3.THE PRODUCT</th>
<th>Describe the product which you want to build &amp; feature using TDD.</th>
</tr>
</thead>
<tbody>
<tr>
<td>It is possible to build a minimal increment of functionality that passes this test? If so, describe it.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>4.FEATURE</th>
<th>What part of the product do you want to build? We recommend starting small.</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>5. TEST FIRST</th>
<th>Think a little about design and choose a test that this feature should pass.</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>6. MINIMUM INCREMENT</th>
<th>Is it possible to build a minimal increment of functionality that passes this test? If so, describe it.</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>7. REFACTORING</th>
<th>After the test passes, is it possible to remove this test increment and make changes to improve it and keep the test passing?</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>8. FRAMEWORK</th>
<th>Is there any technology to automate this test? If so, describe it and setting up the technology.</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>9. FAST</th>
<th>Is the test execution fast? Can it be faster?</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>10. REPEATABLE</th>
<th>Is it repeatable? If you run the same test multiple times, (if using the product the result is the same)</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>10.CYCLE</th>
<th>If the feature is not finished repeat the cycle from 1 with another test.</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>NOTES</th>
<th>6, 9 and 10 have to do with the technical scalability of doing TDD with no chosen test.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>The more difficult it is to find stable tests, the more difficult it is going to be to do TDD for the chosen feature.</td>
</tr>
<tr>
<td></td>
<td>In any case, when TDD is partially used, it can add a lot of value.</td>
</tr>
</tbody>
</table>

Andres Jason - Hiroshi Hirose
@andresjason - @h thorh

How to use: Complete the sections from number 3 onwards. The objective of each section is not necessarily to find the ideal scenarios but to find something that close the area as possible preserving a faultless communication. It has experience even the practice that you discover not exactly as the one in software, if it’s close enough to the essence will have a tremendous value in your context.
and we started with 4 practices

Based on our experiences we choose to start building explorers around this 4 practices:

- Test Driven Development
- Continuous Integration
- Modular Architecture
- Feature Flags

From which we extracted their essence and built a explorer to work around them.
Test Driven Development

“It allows to build products in small increments with higher quality and lower risk”
Given a feature, think a little about its design and choose a first test that this feature should pass.

Write the test.

Run the test.

It shouldn't pass because we didn't build anything yet. (Red)
Test Driven Development

Test First

Minimum Increment
Build the simplest possible solution that will make the test pass. If we run the test it will pass (Green).
Test Driven Development

**Test First**

Refactoring
If needed. We modify the feature to improve its technical quality (Refactor).
And we run the test to verify that is still working.

**TDD**

Minimum Increment
Test Driven Development

- Cycle
If the feature is not ready we choose another test and we repeat the cycle with this test.
Test Driven Development

We rely on Technology that provides us with:
- Low test construction cost
- Simple execution (automated)
- Simple result (red or green)

- Cycle
- Framework
Test Driven Development

- Fast and Repeatable
- Framework
- Cycle
- Test First

Minimum Increment

Refactoring

TDD

We look for tests to be fast and repeatable so that we can use them many times in short cycles.
Test Driven Development

- Test First
- Refactoring
  - Cycle
  - Framework
  - Fast and Repeatable

Minimum Increment
Test Driven Development Explorer
<table>
<thead>
<tr>
<th>1.WHY</th>
<th>3.THE PRODUCT</th>
<th>6. MINIMUM INCREMENT</th>
<th>8.FRAMEWORK</th>
<th>10.REPEATABLE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Why does teams adopt this practice?</td>
<td>Describe the product which you want to build a feature using TDD</td>
<td>Is it possible to build a minimal increment of functionality that passes this test? If so, describe it.</td>
<td>Is there any technology to automate this test? Is it relatively low cost? (the cost includes setting up the technology)</td>
<td>Is it repeatable? (if I run the same test multiple times without changing the product the result is the same)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>2.THE ESSENCE</th>
<th>4.FEATURE</th>
<th>7.REFACTORING</th>
<th>9. FAST</th>
<th>10.CYCLE</th>
</tr>
</thead>
<tbody>
<tr>
<td>What is the essence of this practice beyond the technical aspect?</td>
<td>What part of the product do you want to build? We recommend starting small.</td>
<td>After the test passes, is it possible to review the implementation and make changes to improve it and keep the test passing?</td>
<td>Is the test execution fast? Can it be faster?</td>
<td>If the feature is not finished repeat the cycle from 5 with another test.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>5. TEST FIRST</th>
<th>8. FRAMEWORK</th>
</tr>
</thead>
<tbody>
<tr>
<td>Think a little about its design and choose a first test that this feature should pass</td>
<td></td>
</tr>
</tbody>
</table>

**How to use:** Complete the sections from number 3 onwards. The objective of each section is not necessarily to find the ideal scenario but to find something as close to the essence as possible promoting a facilitated conversation. In our experience even the practice that you discover is not exactly as the one in software, if it's close enough to the essence will have a tremendous value in your context.
Wikispeed - IT Technical Practices Building a Car

Book by Paolo Sammicheli

Joe Justice @ Agile2012

https://wikispeed.com/
Why does teams adopt this practice?
TDD helps create products iteratively and incrementally with higher quality and greater confidence in face of future changes.

What part of the product do you want to build? We recommend starting small.
Crashworthiness of the car

What is the essence of this practice beyond the technical aspect?
- Test First
- Minimum Increment
- Refactoring
- Cycle
- Framework
- Fast
- Repeatable

Describe the product which you want to build a feature using TDD
A new Car

Think a little about its design and choose a first test that this feature should pass
Five-star crashworthiness score according to the official regulations.

Is it possible to build a minimal increment of functionality that passes this test? If so, describe it.
Yes. We can run the test with a version of the car that is not the final.

After the test passes, is it possible to review the implementation and make changes to improve it and keep the test passing?
Yes. The architecture of the car is optimized to be able to be modified in a simple way.

Is there any technology to automate this test? Is it relatively low cost? (the cost includes setting up the technology)
Real Crash Test, but it is expensive.

Is the test execution fast? Can it be faster?
Yes, is fast enough.

What is the essence of this practice beyond the technical aspect?
- Test First
- Minimum Increment
- Refactoring
- Cycle
- Framework
- Fast
- Repeatable

How to use: Complete the sections from number 3 onwards. The objective of each section is not necessarily to find the ideal scenario but to find something as close as possible promoting a facilitated conversation. In our experience even the practice that you discover is not exactly as the one in software, if it’s close enough to the essence will have a tremendous value in your context.

Andrés Joaquín - Hiroshi Hiromoto
@andrescjoaquin - @hhiroshi

DATE: 07/27/2023
1.WHY
Why does teams adopt this practice?

TDD helps create products iteratively and incrementally with higher quality and greater confidence in face of future changes.

2.THE ESSENCE
What is the essence of this practice beyond the technical aspect?

- Test First
- Minimum Increment
- Refactoring
- Cycle
- Framework
- Fast
- Repeatable

3.THE PRODUCT
Describe the product which you want to build a feature using TDD

Conference Session

4.FEATURE
What part of the product do you want to build? We recommend starting small.

Abstract

5. TEST FIRST
Think a little about its design and choose a first test that this feature should pass

If someone reads the title and we ask them what the session is about, they should describe what we have in mind for the abstract.

6. MINIMUM INCREMENT
Is it possible to build a minimal increment of functionality that passes this test? If so, describe it.

Yes. We can write only the title. The rest of the abstract could be just a draft, expressing the idea in a very general way.

7.REFACTORING
After the test passes, is it possible to review the implementation and make changes to improve it and keep the test passing?

Yes. It is easy to modify a title.

8.FRAMEWORK
Is there any technology to automate this test? Is it relatively low cost? (the cost includes setting up the technology)

Yes. We could ask ChatGPT

But it is also really easy to do a semi-automated test using a very simple google form with real people.

9. FAST
Is the test execution fast? Can it be faster?

Yes, is fast enough.

10.CYCLE
If the feature is not finished repeat the cycle from 5 with another test.

10.REPEATABLE
Is it repeatable?

(if I run the same test multiple times without changing the product the result is the same)

Yeah. Both with ChatGPT and with a form.

NOTES
8, 9 and 10 have to do with the technical feasibility of doing TDD with the chosen test.

The more difficult it is to find viable tests, the more difficult it is going to be to do TDD for the chosen feature.

In any case, even when TDD is partially used, it can add a lot of value.

How to use: Complete the sections from number 3 onwards. The objective of each section is not necessarily to find the ideal scenario but to find something as close the essence as possible promoting a facilitated conversation. In our experience even the practice that you discover es not exactly as the one in software, if it’s close enough to the essence will have a tremendous value in your context.
Continuous Integration

“The greatest and most wide ranging benefit of Continuous Integration is reduced risk”

Martin Fowler
Continuous Integration (CI)

**Single Source of Truth**
We maintain a single repository where the last integrated version of the product is.
Continuous Integration (CI)

Single Source of Truth

Daily integration (at least)
Everyone upload their advances and integrate it at least one time per day.
Continuous Integration (CI)

Single Source of Truth

Daily integration (at least)

**Integrity self-verified**
Every time the product is integrated
its coherence is self-verified
(structurally correct)
Continuous Integration (CI)

Fix errors immediately
The errors that are detected while integrated are fixed immediately.

Single Source of Truth

Daily integration (at least)

Integrity self-verified
Continuous Integration (CI)

Single Source of Truth

Daily integration (at least)

Integrity self-verified

Fix errors immediately

Transparency
Everyone can see the current state of the product
Continuous Integration (CI)

- Single Source of Truth
- Daily integration (at least)
- Integrity self-verified
- Fix errors immediately
- Transparency
- Easy access
  Make it Easy for Anyone to Get the Latest Version
Continuous Integration (CI)

Single Source of Truth
Fix errors immediately

Daily integration (at least)
Transparency

Integrity self-verified
Easy access
Continuous Integration (CI)

**Single Source of Truth**
We maintain a single repository where the last integrated version of the product is.

**Daily integration (at least)**
Everyone upload their advances and integrate it at least one time per day.

**Integrity self-verified**
Every time the product is integrated its coherence is self-verified (structurally correct)

**Fix errors immediately**
The errors that are detected while integrated are fixed immediately.

**Transparency**
Everyone can see the current state of the product

**Easy access**
Make it Easy for Anyone to Get the Latest Version
Continuous Integration Explorer
## Continuous Integration Explorer

<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>1. WHY</strong></td>
<td>Why does teams adopt this practice?</td>
</tr>
<tr>
<td><strong>2. THE ESSENCE</strong></td>
<td>What is the essence of this practice beyond the technical aspect?</td>
</tr>
<tr>
<td><strong>3. THE PRODUCT</strong></td>
<td>Describe the product which you want to continuous integrate it.</td>
</tr>
<tr>
<td><strong>4. SINGLE SOURCE OF TRUTH</strong></td>
<td>Is it possible to have a single place where the most updated version of the product is available? If so, describe it.</td>
</tr>
<tr>
<td><strong>5. EASY ACCESS</strong></td>
<td>Is the single repository accessible to everyone in the team? Is it easy to access? If so, describe how can people access to it.</td>
</tr>
<tr>
<td><strong>6. DAILY INTEGRATION</strong></td>
<td>Describe how will you promote that the team integrates the advances of the product at least one time per day.</td>
</tr>
<tr>
<td><strong>7. INTEGRITY SELF-VERIFIED</strong></td>
<td>Is it possible that every time a change is integrated, the integrity and coherence of the product is self-verified? If so, describe how it will be implemented.</td>
</tr>
<tr>
<td><strong>8. FIX ERRORS IMMEDIATELY</strong></td>
<td>Describe how will you ensure that every time something fails while integrating, the issues arises are fixed.</td>
</tr>
<tr>
<td><strong>9. TRANSPARENCY</strong></td>
<td>Does everyone can see the state of the product at any given time? If so, describe how can people see that state without needing other people.</td>
</tr>
</tbody>
</table>

### NOTES

The sections 4, 6 and 8 are the core of the practice and without them the practice loses its value. Ideally the section number 7 is automated using some kind of software.

### How to use:

Complete the sections from number 3 onwards. The objective of each section is not necessarily to find the ideal scenario but to find something as close the essence as possible promoting a facilitated conversation. In our experience even the practice that you discover is not exactly as the one in software, if it's close enough to the essence will have a tremendous value in your context.
### 1. WHY
Why does teams adopt this practice?

On the whole the greatest and most wide ranging benefit of Continuous Integration is reduced risk. At all times you know where you are, what works, what doesn't, the outstanding issues you have in your product.

### 2. THE ESSENCE
What is the essence of this practice beyond the technical aspect?

- Single source of truth
- Daily integration (at least)
- Integrity self-verified
- Fix errors immediately
- Transparency
- Easy access

### 3. THE PRODUCT
Describe the product which you want to continuous integrate it

Building a new Car to participate on an innovation competition

### 4. SINGLE SOURCE OF TRUTH
Is it possible to have a single place where the most updated version of the product is available? If so, describe it.

Team member uploads a new 3d drawing to Dropbox, Box.net, Windows SkyDrive, or any of the file sharing technologies in use, to a single shared drive.

### 5. EASY ACCESS
Is the single repository accessible to everyone in the team? Is it easy to access? If so, describe how can people access to it.

All team members has access to the drive that contains the 3d designs.

### 6. DAILY INTEGRATION
Describe how will you promote that the team integrates the advances of the product at least one time per day

Team members uploads a 3d drawing everytime they have a new design

### 7. INTEGRITY SELF-VERIFIED
Is it possible that every time a change is integrated, the integrity and coherence of the product is self-verified? If so, describe how it will be implemented.

WIKISPEED can simulate crash tests and stress tests on the part using FEA and a software package like LS Dyna247. Can simulate airflow, aerodynamics, fluid flow, heat transfer, and electrical propagation using CFD

### 8. FIX ERRORS IMMEDIATELY
Describe how will you ensure that every time something fails while integrating, the issues arises are fixed.

Whenever a new CAD shows up and write out a 1-page report with a list of red or green lights. Green lights mean the test is the same or better than the current version or passes an explicit test for that part or module.

### 9. TRANSPARENCY
Does everyone can see the state of the product at any given time? If so, describe how can people see that state without needing other people.

It’s easy to know what the current best part is; the version of record is whatever part in CAD has passed all tests with the most green lights.

### NOTES
The sections 4, 6 and 8 are the core of the practice and without them the practice loses its value.

Ideally the section number 7 is automated using some kind of software.

**Andrés Joaquín - Hiroshi Hiromoto**
@andrescjoaquin - @hhiromoto

**How to use:** Complete the sections from number 3 onwards. The objective of each section is not necessarily to find the ideal scenario but to find something as close the essence as possible promoting a facilitated conversation. In our experience even the practice that you discover es not exactly as the one in software, if it’s close enough to the essence will have a tremendous value in your context.
### 1. WHY
Why does teams adopt this practice?

On the whole the greatest and most wide ranging benefit of Continuous Integration is reduced risk. At all times you know where you are, what works, what doesn't, the outstanding issues you have in your product.

### 2. THE ESSENCE
What is the essence of this practice beyond the technical aspect?

- Single source of truth
- Daily integration (at least)
- Integrity self-verified
- Fix errors immediately
- Transparency
- Easy access

### 3. THE PRODUCT
Describe the product which you want to continuous integrate it

Marketing and Legal copies of the digital channels of a bank.

### 4. SINGLE SOURCE OF TRUTH
Is it possible to have a single place where the most updated version of the product is available? If so, describe it.

All the text are in a wiki page

### 5. EASY ACCESS
Is the single repository accessible to everyone in the team? Is it easy to access? If so, describe how can people access to it.

All team members including the legal and marketing team has access to the wiki platform.

### 6. DAILY INTEGRATION
Describe how will you promote that the team integrates the advances of the product at least one time per day

Everytime there is a new version of any copy to review, people upload it.

### 7. INTEGRITY SELF-VERIFIED
Is it possible that every time a change is integrated, the integrity and coherence of the product is self-verified? If so, describe how it will be implemented.

Basic text formatting, spell checking and length is verified in each integration.

### 8. FIX ERRORS IMMEDIATELY
Describe how will you ensure that every time something fails while integrating, the issues arises are fixed.

Integrity has real-time feedback and after reviews an automated mail for fixing errors are sent. So team member upload a fix as soon as possible.

### 9. TRANSPARENCY
Does everyone can see the state of the product at any given time? If so, describe how can people see that state without needing other people.

Anyone in the team can see the progress status of any copy.

### NOTES
The sections 4, 6 and 8 are the core of the practice and without them the practice loses its value.

Ideally the section number 7 is automated using some kind of software.

Andrés Joaquín - Hiroshi Hiromoto
@andrescjoaquin - @hhiroshi

How to use: Complete the sections from number 3 onwards. The objective of each section is not necessarily to find the ideal scenario but to find something as close the essence as possible promoting a facilitated conversation. In our experience even the practice that you discover es not exactly as the one in software, if it's close enough to the essence will have a tremendous value in your context.
Modular Architecture

“It produces more adaptable products, where different parts of the product can evolve with a high level of independence from the rest of the parts”
Modular Architecture

A Product has a Non-Modular Architecture (or Design) when it includes a lot of internal components with a lot of interdependencies between them.
Modular Architecture

With a Modular Architecture we design and group components on less dependent (Loosely Coupled) modules.
Modular Architecture

With a Modular Architecture we design and group components on less dependent (Loosely Coupled) modules.
Modular Architecture

Loosely Coupled Modules

Contract First Design
Each module is designed thinking first on the contract (API) with other modules
Modular Architecture

Loosely Coupled Modules

Contract First Design

Automated Tests
Contracts have automated tests
Modular Architecture

Loosely Coupled Modules

Contract First Design

Automated Tests

Minimum Version

Contracts can begin to be implemented with a minimum version. You don’t need the perfect versión to start.
Modular Architecture

Loosely Coupled Modules

Contract First Design

Minimum Version

Automated Tests

Emergent Design

Initial designs can evolve without having to start from 0 in each evolution.
Modular Architecture

- Loosely Coupled Modules
- Contract First Design
- Automated Tests
- Minimum Version
- Emergent Design
Modular Architecture
Explorer
<table>
<thead>
<tr>
<th><strong>1.WHY</strong></th>
<th>Why does teams adopt this practice?</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>2.THE ESSENCE</strong></td>
<td>What is the essence of this practice beyond the technical aspect?</td>
</tr>
<tr>
<td>- Loosely Coupled Modules</td>
<td></td>
</tr>
<tr>
<td>- Contract First Design</td>
<td></td>
</tr>
<tr>
<td>- Minimum Version</td>
<td></td>
</tr>
<tr>
<td>- Emergent Design</td>
<td></td>
</tr>
<tr>
<td>- Automated Tests</td>
<td></td>
</tr>
<tr>
<td><strong>3.THE PRODUCT</strong></td>
<td>Describe the product which you want to build</td>
</tr>
<tr>
<td><strong>4. LOOSELY COUPLED MODULES</strong></td>
<td>Can it be divided into modules with low dependency on each other? How many? Can they be more?</td>
</tr>
<tr>
<td><strong>5.CONTRACT FIRST DESIGN</strong></td>
<td>Can these modules be designed starting by agreeing their contracts with other modules?</td>
</tr>
<tr>
<td><strong>6.MINIMUM VERSION</strong></td>
<td>Can you start with minimal versions that fulfill the contracts?</td>
</tr>
<tr>
<td><strong>7.EMERGENT DESIGN</strong></td>
<td>Can this version evolve without necessarily having to start from 0 each time?</td>
</tr>
<tr>
<td><strong>8.AUTOMATED TESTS</strong></td>
<td>Are there tests that allow you to test changes in a module in an automated way?</td>
</tr>
</tbody>
</table>

**NOTES**

If in general the answers to these questions is yes, then we are getting closer to what we need in terms of architecture.

**How to use:** Complete the sections from number 3 onwards. The objective of each section is not necessarily to find the ideal scenario but to find something as close to the essence as possible promoting a facilitated conversation. In our experience even the practice that you discover is not exactly as the one in software, if it's close enough to the essence will have a tremendous value in your context.

Andrés Joaquín - Hiroshi Hiromoto  
@andrescjoaquin - @hhiroshi
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Why does teams adopt this practice?</td>
<td>What is the essence of this practice beyond the technical aspect?</td>
<td>Describe the product which you want to build</td>
<td>Can it be divided into modules with low dependency on each other? How many? Can they be more?</td>
<td>Can these modules be designed starting by agreeing their contracts with other modules?</td>
<td>Can you start with minimal versions that fulfill the contracts?</td>
<td>Can this version evolve without necessarily having to start from 0 each time?</td>
<td>Are there tests that allow you to test changes in a module in an automated way?</td>
<td>If in general the answers to these questions is yes, then we are getting closer to what we need in terms of architecture.</td>
</tr>
<tr>
<td>It produces more adaptable products, where different parts of the product can evolve with a high level of independence from the rest of the parts. It makes it easier to change the product.</td>
<td>- Loosely Coupled Modules</td>
<td>Car</td>
<td>Yes. 8 modules.</td>
<td>Yes. The API between these modules can be defined at the beginning.</td>
<td>Yes</td>
<td>Sometimes yes and other times no. In this sense it is important to leave free physical space for growth when possible.</td>
<td>Yes</td>
<td></td>
</tr>
</tbody>
</table>

**How to use:** Complete the sections from number 3 onwards. The objective of each section is not necessarily to find the ideal scenario but to find something as close as possible promoting a facilitated conversation. In our experience even the practice that you discover is not exactly as the one in software, if it's close enough to the essence will have a tremendous value in your context.

Andrés Joaquín - Hiroshi Hiromoto
@andrescjoaquin - @hhiroshi
Feature Flags

“Allows teams to unlock dynamic control without rebuilding the product.”
Feature Flags

Flags Objective
We choose the goal of the feature flags. It could be hide a feature for a release, experiment with something new, enable something to a user group or test different options for a single feature.

[Diagram showing the relationship between a new feature, feature flags, and customers]
Feature Flags

Flags Objective

Toggle Points
We design how the toggle will work and the options that we will apply the flags on if applicable
Feature Flags

When we test different options within different target audience we need to design those target audience.
Router
We use a router that determines the flag state (on/off) without modifying the product.
Feature Flags

Flags Objective

Toggle Points

Target Audience

Monitoring
Options are monitored in relation to their usage and performance in order to make decisions about them
Feature Flags

Flags Objective

New feature

Feature Flags

Customers

Target Audience

Toggle Points

Monitoring

Router
Feature Flags

Flags Objective
We choose the goal of the feature flags. It could be hide a feature for a release, experiment with something new, enable something to a user group or test different options for a single feature.

Toggle Points
We design how the toggle will work and the options that we will apply the flags on if applicable.

Users Groups
When we test different options within different user groups we need to design those user groups.

Router
We use a router that determines the flag state (on/off) without modifying the product.

Monitoring
Options are monitored in relation to their usage and performance in order to make decisions about them.
### Feature Flags Explorer

<table>
<thead>
<tr>
<th>1. WHY</th>
<th>3. THE PRODUCT</th>
<th>5. FLAGS OBJECTIVE</th>
<th>7. TARGET AUDIENCE</th>
<th>9. MONITORING</th>
</tr>
</thead>
<tbody>
<tr>
<td>Why does teams adopt this practice?</td>
<td>Describe the product you are going to work on</td>
<td>What is the goal of this Feature Flag?</td>
<td>Which is the target audience to which we will present the different options? If they are not fixed groups describe the selection mechanism</td>
<td>How are you going to monitor of the performance of the different options to make decisions later about them?</td>
</tr>
</tbody>
</table>

Feature Flags allows teams to experiment, test hypothesis and add flexibility to products thru unlock dynamic control of features without rebuilding the product. When we mention dynamic control it means for example turning on or off a specific feature or showing different options of a feature to different user groups.

<table>
<thead>
<tr>
<th>2. THE ESSENCE</th>
<th>4. FEATURE</th>
<th>6. TOGGLE POINTS</th>
<th>8. ROUTER</th>
<th>NOTES</th>
</tr>
</thead>
<tbody>
<tr>
<td>What is the essence of this practice beyond the technical aspect?</td>
<td>What feature do you want to use Feature Flags on?</td>
<td>How are you going to turn on/off the feature? What are the options available? (if applicable)</td>
<td>What's the mechanism you will use to determine when a flag is on or off? How it will activates or deactivates the flags?</td>
<td>The section 5 will determine what kind of flag you will implement, so the subsequent sections will depend on that decision. The section 6 is the core of the practice. The section 8 is also a core of the practice when we have different options.</td>
</tr>
</tbody>
</table>

- Flags Objective
- Toggle Points
- User Groups
- Router
- Monitoring

**How to use:** Complete the sections from number 3 onwards. The objective of each section is not necessarily to find the ideal scenario but to find something as close the essence as possible promoting a facilitated conversation. In our experience even the practice that you discover is not exactly as the one in software, if it's close enough to the essence will have a tremendous value in your context.

Andrés Joaquín - Hiroshi Hiromoto
@andrescjoaquin - @hhiroshi
### Feature Flags Explorers

<table>
<thead>
<tr>
<th>Date: 07/27/2023</th>
</tr>
</thead>
</table>

#### 1. WHY
Why does teams adopt this practice?

Feature Flags allows teams to experiment, test hypothesis and add flexibility to products thru unlock dynamic control of features without rebuilding the product.

When we mention dynamic control it means for example turning on or off a specific feature or showing different options of a feature to different user groups.

---

#### 2. THE ESSENCE
What is the essence of this practice beyond the technical aspect?

- Flags Objective
- Toggle Points
- User Groups
- Router
- Monitoring

---

#### 3. THE PRODUCT
Describe the product you are going to work on

Car

---

#### 4. FEATURE
What feature do you want to use Feature Flags on?

Rear-heated seats

---

#### 5. FLAGS OBJECTIVE
What is the goal of this Feature Flag?

Enable the capability of heating the rear seats of the car based on the subscription type of the car owner.

---

#### 6. TOGGLE POINTS
How are you going to turn on/off the feature? What are the options available? (if applicable)

The capability of heating the rear seats are built on the car but an electronic component turn them on and off.

---

#### 7. TARGET AUDIENCE
Which is the target audience to which we will present the different options? If they are not fixed groups describe the selection mechanism

The customer with a Premium Subscription has the feature on.

---

#### 8. ROUTER
What's the mechanism you will use to determine when a flag is on or off? How it will activates or deactivates the flags?

The main car panel controller validates the user subscription type and interacts with the electronic component of the heating seat system to turn it on or off.

---

#### 9. MONITORING
How are you going to monitor of the performance of the different options to make decisions later about them?

We collect data about the usage of the heating seat system that is collected daily from the car.

---

### NOTES

- The section 5 will determine what kind of flag you will implement, so the subsequent sections will depend on that decision.
- The section 6 is the core of the practice.
- The section 8 is also a core of the practice when we have different options.

---

**Andrés Joaquin - Hiroshi Hiromoto**
@andrescjoaquin - @hhiroshi

**How to use:** Complete the sections from number 3 onwards. The objective of each section is not necessarily to find the ideal scenario but to find something as close the essence as possible promoting a facilitated conversation. In our experience even the practice that you discover es not exactly as the one in software, if it's close enough to the essence will have a tremendous value in your context.
Let’s explore!
In groups of 2 or 3

- Think about a context in your organization (outside IT) where the use of technical practices could be of benefit. It can be any domain/business unit you’re working with. It can be a Product but it can also be a Service. **Choose one.**

- **Choose one of the explorer** that you have available at the table and **complete it** using the context selected.

- If you want a digital copy you can find it here (Gmail account required)

- If you have any doubts while completing the explorer, **just call us out!**
What’s next?

• You can use this presentation (available in the conference site) to **present the technical practices and the explorer** to teams working outside IT.

• The explorers can be used directly by those teams or you can use it to **guide a conversation** with them.

• If you use the explorers and found them useful or have feedback, **please reach out** so we can continue iterating over this idea (our contact information is in a following slide).
Resources to deep dive

Recommended session

Today at 3:45 PM - 5:00 PM
Coastal Ballroom B,4,5
¡Thanks!

ANDRÉS JOAQUÍN
@andrescjaoquin
https://www.linkedin.com/in/andresjaoquin
andres.joaquin@kleer.la

HIROSHI HIROMOTO
@hhiroshi
https://www.linkedin.com/in/hhiroshi
hiro@ment.la
Join Agile Alliance today!

Become an Agile Alliance member and help support our non-profit mission, while gaining access to valuable benefits like online events, in-person conference discounts, and event session videos.