Key Technical Decisions framework

Birds of a feather session

DrupalCon
BARCELONA2024
24-27 SEPTEMBER

25.09.2024

GLOBAL SCALE, LOCAL PRESENCE - WORKING AS ONE

Revenues
500M +

Growth rate
+19%
2018-2023 CAGR

Offices in
20+
Countries

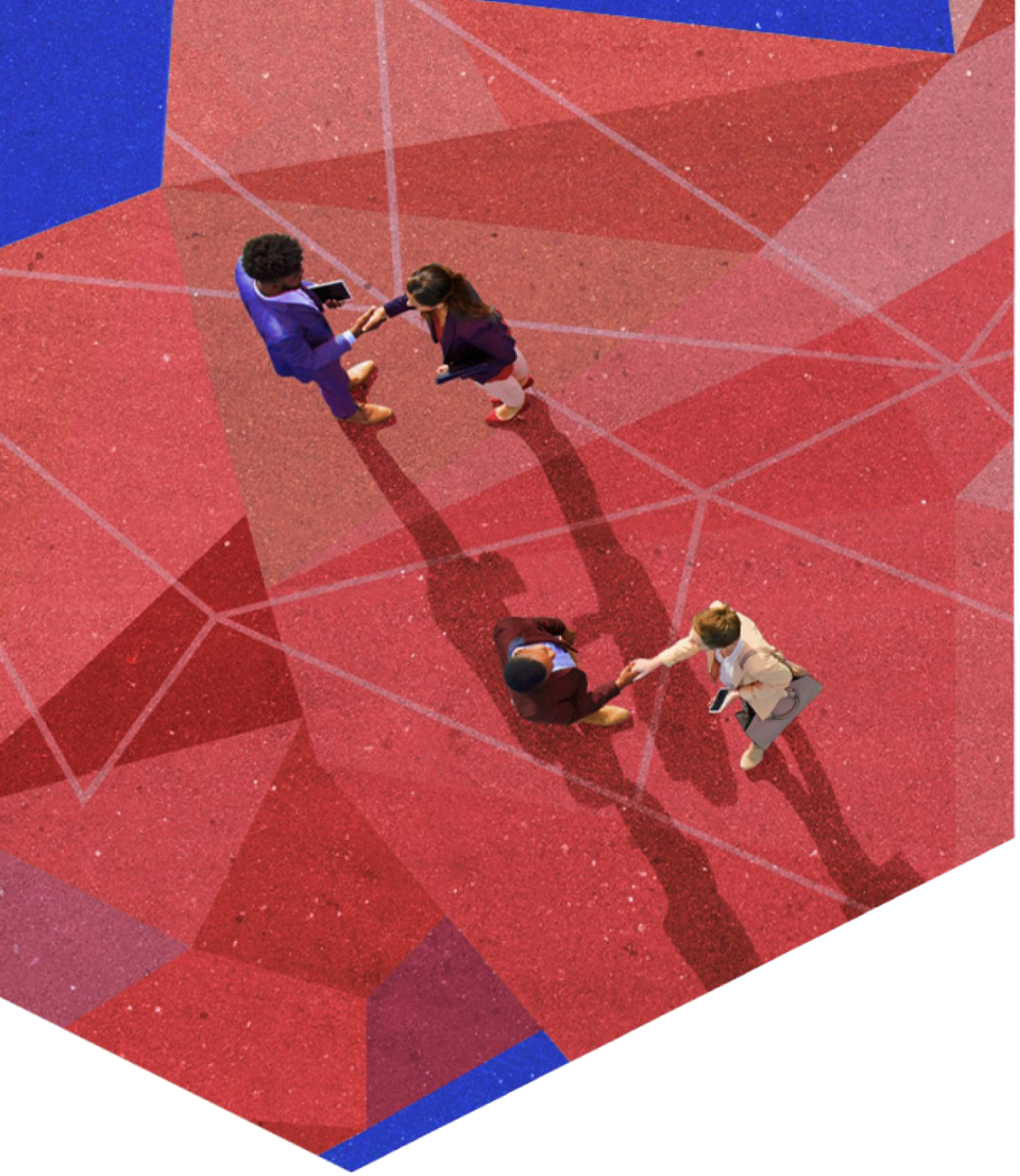
Clients
900

Active
Projects in
30+
Countries

We are also present in: Gabrovo - Veliko Tarnovo - Plovdiv - Gijon – Turin – Macerata – Naples – Bari – Rende – Catania – Vienna - Luxembourg – Düsseldorf – Riga

Jorge López-Lago
Solutions Architect

On Drupal.org for 19 years 7 months



Summary

1. Introduction - what is this about?
2. What is a KTD?
3. Anatomy of a KTD
4. Why KTDs?
5. What about you?
6. What now?

What is this about?

The average digital project has changed significantly.

From sizable web builds we are now landing contracts for multi-million dollar composable digital experience platforms.

With decoupled front-ends, elaborate system architectures and multiple third party integrations.

Infrastructure needs have changed too.

From a few servers with simple go to market mechanisms.

To containerised public clouds, distributed file systems and high availability databases, with complex integration and deployment pipelines.

Our ways of working have shifted also.

From a couple of developers and few photoshop files, finished before development started.

To multidisciplinary agile teams, iterative component based design, decoupled JS front-end frameworks, third-party component libraries and multiple touchpoints and channels.

Digital does not have an end date.

Today everything is in product-mode.

Our work often is not governed by a beginning, a middle and an end.

We are working on products, platforms or services, with a fluid scope, flexible timelines and adjustable budgets.

Our work often needs refinement, refactoring, adaptation.

All of that comes with a whole set of challenges.

How do you make the **best technical decisions** in such a dynamic environment?

How do you know if an approach is good enough for now and **effectively manage scope**?

How do you quantify and manage the **technical debt** introduced?

How do you **document those decisions** so they can be revisited later on?

How do you **give voice to everyone** who has something valuable to contribute?

How do you **create consensus** within your team?

What is a KTD?

Key technical Decisions

Are decisions, that have profound strategic and architectural implications, for the organisation.

They enable or inhibit strategic goals: evolution, growth and value creation.

KTDs are not User Stories

User stories are about technical implementation details, functionality.

They still may require discussion over details, but the approach has already been given.

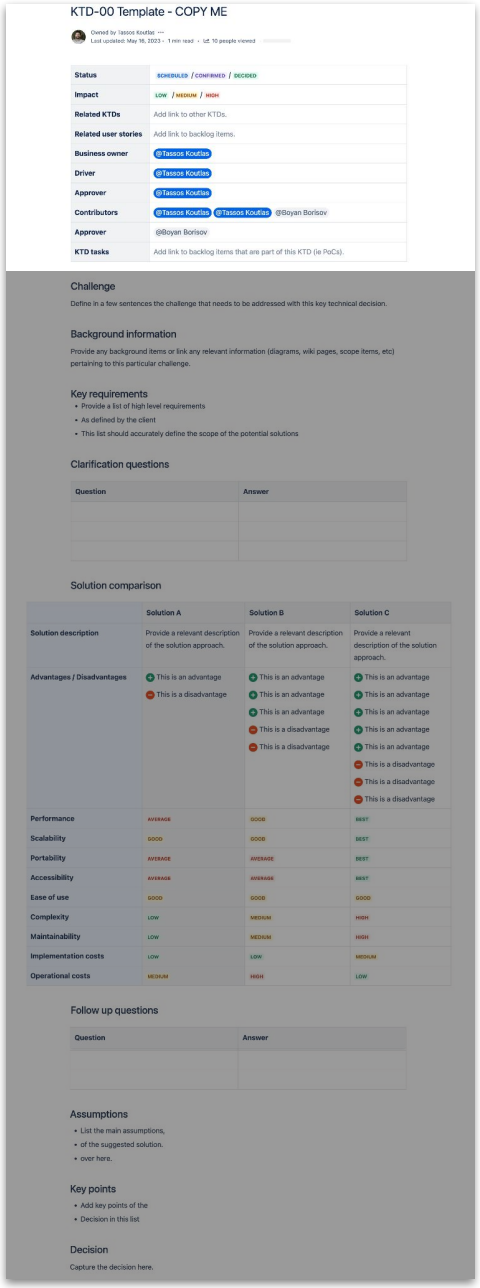
User Stories influence project delivery, not strategy.

The KTD process

The KTD process is a lightweight but powerful methodology for driving key technical decisions when implementing digital products. Organisations are able to accommodate change in dynamic environments and create **alignment**, **consensus** and **trust** with stakeholders.

The KTD process can work alongside waterfall or agile methods. For **agile** methods, the sprint cadence needs to be combined with KTD process so one informs the other during project delivery. For **waterfall** the workflow is straightforward, make all the decisions prior to technical project delivery.

Anatomy of a KTD



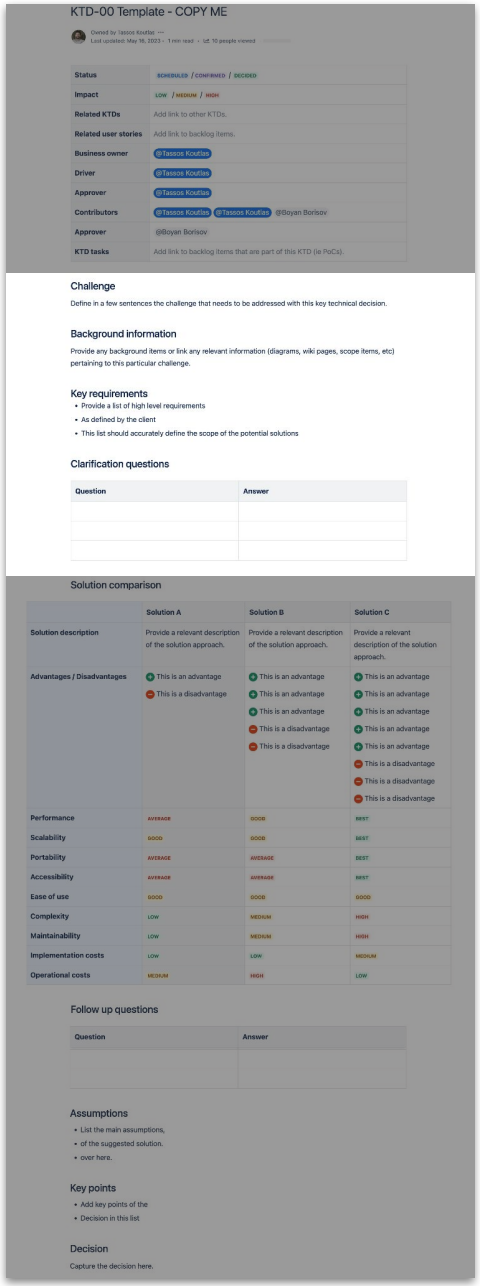
Header section

Contains basic information about the decision governance and how it fits within the current project.

KTD-00 Template - COPY ME

Owned by Tassos Koutlas •••
Last updated: May 16, 2023 • 1 min read • 10 people viewed

Status	SCHEDULED / CONFIRMED / DECIDED
Impact	LOW / MEDIUM / HIGH
Related KTDs	Add link to other KTDs.
Related user stories	Add link to backlog items.
Business owner	@Tassos Koutlas
Driver	@Tassos Koutlas
Approver	@Tassos Koutlas
Contributors	@Tassos Koutlas @Tassos Koutlas @Boyan Borisov
Approver	@Boyan Borisov
KTD tasks	Add link to backlog items that are part of this KTD (ie PoCs).



Background section

Contains information and context on the challenge and provides a structured way to clarify any questions that may come up.

Challenge

Define in a few sentences the challenge that needs to be addressed with this key technical decision.

Background information

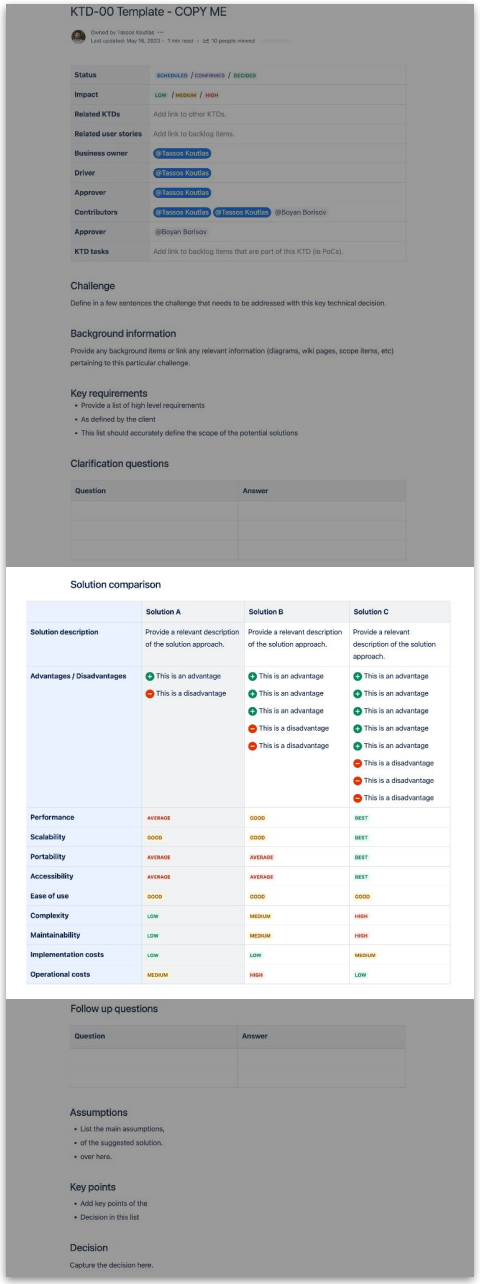
Provide any background items or link any relevant information (diagrams, wiki pages, scope items, etc) pertaining to this particular challenge.

Key requirements

- Provide a list of high level requirements
- As defined by the client
- This list should accurately define the scope of the potential solutions

Clarification questions

Question	Answer



Solutions section

Contains solution information and allows comparison between different approaches for the given context.

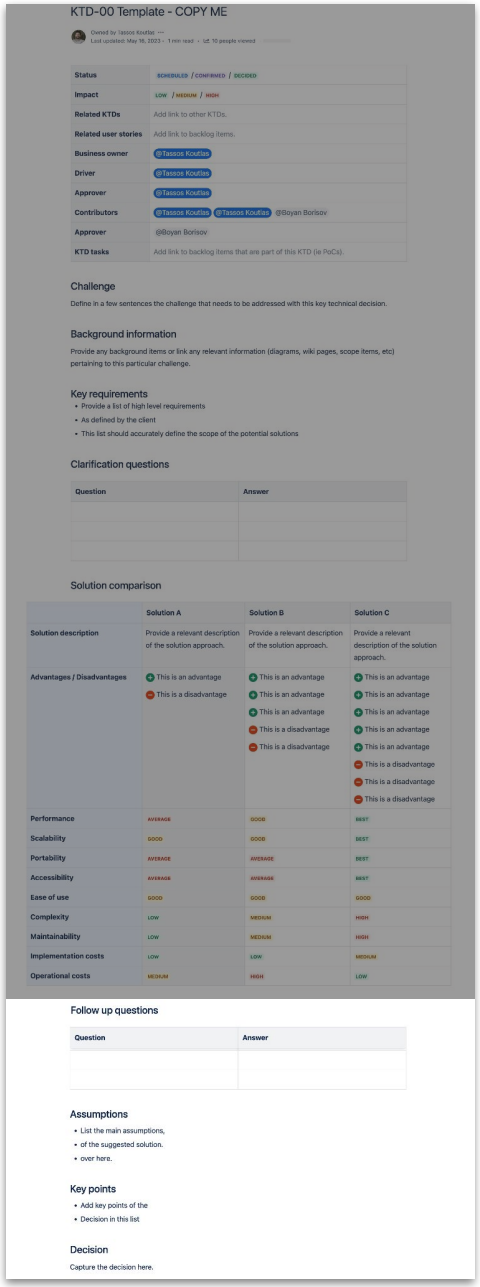
Solution comparison

	Solution A	Solution B	Solution C
Solution description	Provide a relevant description of the solution approach.	Provide a relevant description of the solution approach.	Provide a relevant description of the solution approach.
Advantages / Disadvantages	⊕ This is an advantage ⊖ This is a disadvantage	⊕ This is an advantage ⊕ This is an advantage ⊕ This is an advantage ⊖ This is a disadvantage ⊖ This is a disadvantage	⊕ This is an advantage ⊕ This is an advantage ⊕ This is an advantage ⊕ This is an advantage ⊕ This is an advantage ⊖ This is a disadvantage ⊖ This is a disadvantage ⊖ This is a disadvantage
Performance	AVERAGE	GOOD	BEST
Scalability	GOOD	GOOD	BEST
Portability	AVERAGE	AVERAGE	BEST
Accessibility	AVERAGE	AVERAGE	BEST
Ease of use	GOOD	GOOD	GOOD
Complexity	LOW	MEDIUM	HIGH
Maintainability	LOW	MEDIUM	HIGH
Implementation costs	LOW	LOW	MEDIUM
Operational costs	MEDIUM	HIGH	LOW

Pro Tip

Use specific project KPIs to make sure the traffic light section is relevant to business stakeholders.

That's an easy way to outline the chosen solution and build trust.



Discussion & Decision section

Enables discussion with client of preferred approach and creates a record of the decision taken.

Follow up questions

Question	Answer

Assumptions

- List the main assumptions,
- of the suggested solution.
- over here.

Key points

- Add key points of the
- Decision in this list

Decision

Capture the decision here.

Why KTDs?

Why KTDs?

We devised them as a process to bring consensus and clarity to complex projects in highly regulated industries.

They are ~~battle~~ skirmish tested.

KTDs offer a transformative way to consult, decide and create consensus.

Generally a very well accepted method by teams and clients alike.

Pros and cons.

Trust makes decision making easier.

Teams defining together deliver better.

A decision must be followed through.

Single decision per KTD, don't cram decisions.

Scope decisions for context, timeframes, budgets.

The collaborative nature of the process is not easy for everyone to follow, usually at the beginning.

From KTD to User Story it's a long way.

Good KTDs don't guarantee good delivery.

Don't use KTD as an excuse for over analysing.

What about you?

What is the process you follow?

Architecture Decision Records?

User Stories + JIRA?

Email chains?

Documented meetings?

Formal signed contracts?

Shared spreadsheets?

Just iterate in sprints and the documentation is the code itself?

Outside your pay range?

What now?

We are open sourcing the KTDs framework.

We **have**:

Framework documentation. ✅

KTD templates (Confluence, docx, odt, markdown, plain text). ✅

A website almost ready. 😬

Github repository. ✅

We **need**:

Help with translations.

Help with implementation examples.

Help with templates.

Help improving the framework.

If you are interested, join us to start in October.

key.technical.decisions@gmail.com

Stop and say hi at booth #1!

Connect with me at jorge.lopezlago@jakala.com

JAKALA
TOGETHER TO GET THERE

Thank you!

Disclaimer lorem ipsum dolor sit amet

DrupalCon
BARCELONA2024
24-27 SEPTEMBER