Engineering with layers: How to supercharge your digital projects

Moderator:
Julia Wilson, Tessitura

Presenters:

Joel Enfield
Action Links

ANAHEIM
TLCC
2025

Distraction-free zone

Please keep keyboard and other distractions to a minimum

No recording

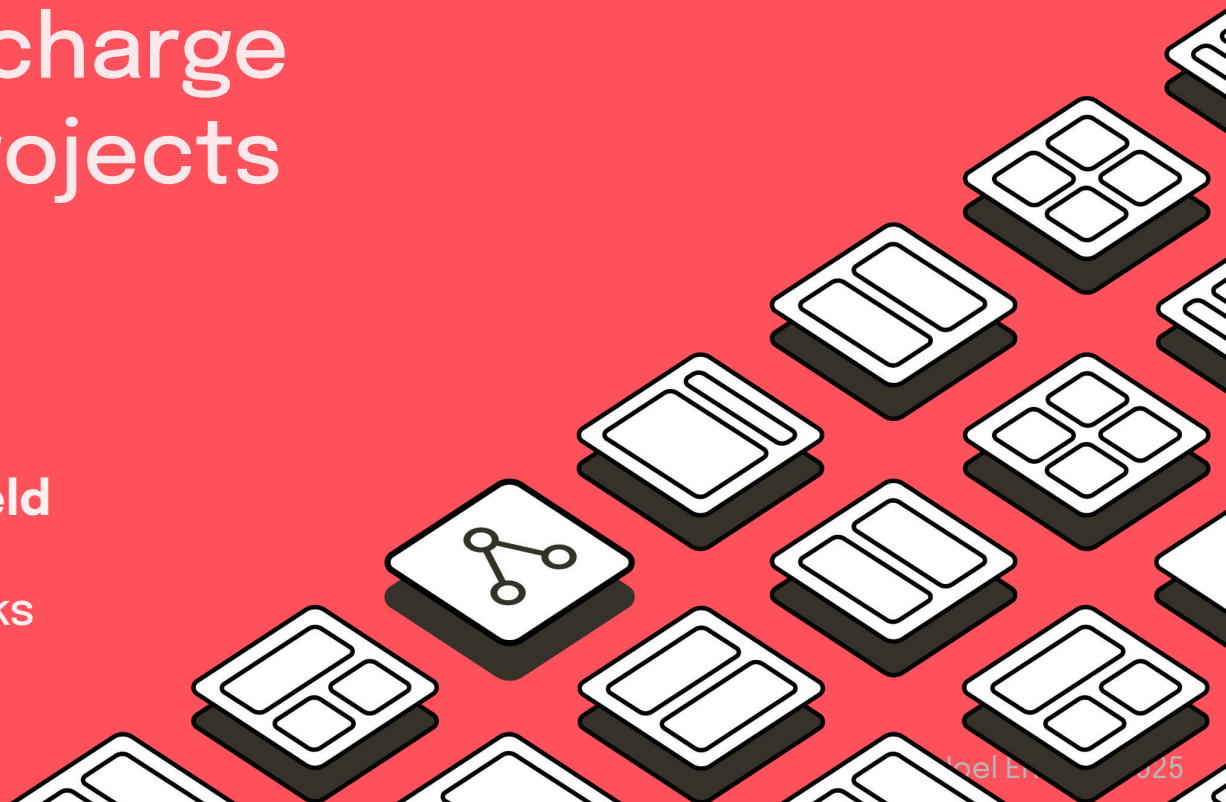
Do not record or broadcast concurrent sessions

ANAHEIM TLCC 2025

Engineering with layers

How to supercharge your digital projects

Joel Enfield
Creator of
Action Links



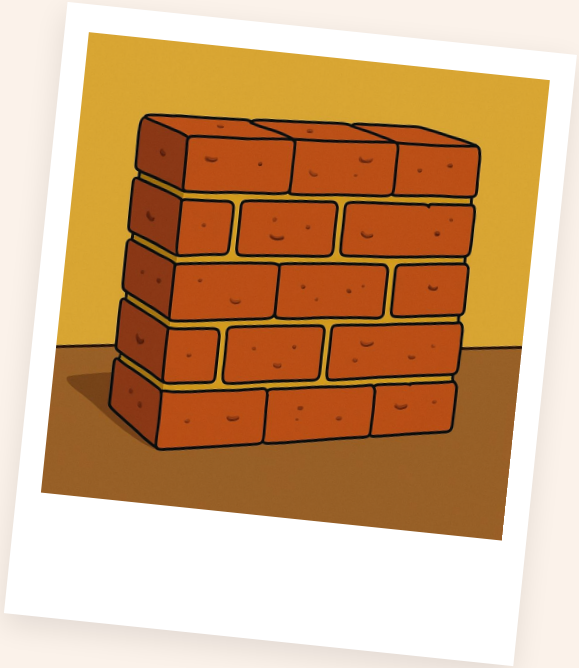
Hey, I’m Joel 👋

Worked in the cultural sector in the UK and North America for the past 15 years as a Product Manager, Technical Analyst, Digital Consultant, self-taught Developer, and now a Founder.

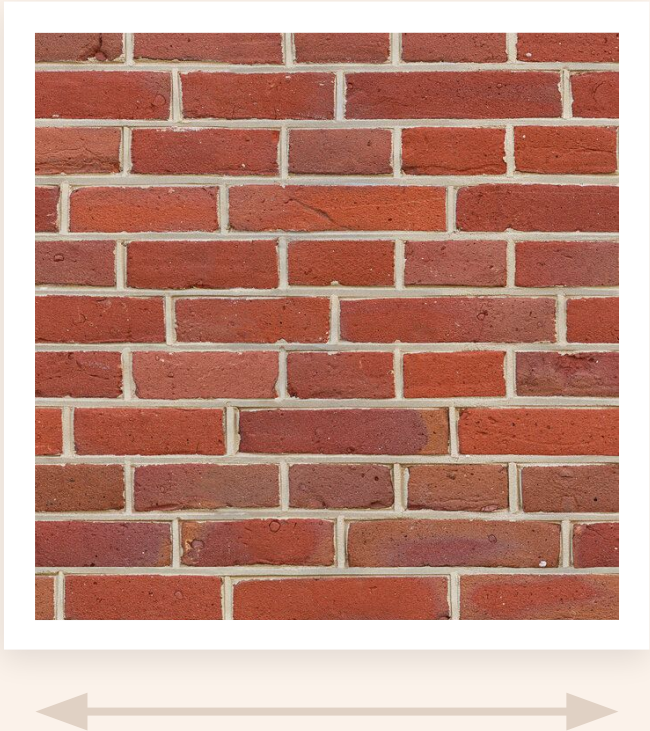
Started building Action Links in 2022 to provide solutions to some of the digital challenges faced by organisations in the cultural sector. Launched the first version of the product in November 2023.

Obsessed with **layers layers layers**
layers layers
layers
layers
layers

All the best things have layers...



Layers can come in both directions...



But too many layers can be a bad thing...

In this session:

Introducing ‘layered design’

- Layered design – what it is and how it can help you
- How to apply layered design to the different parts of a project

Product/Project Management with layered design

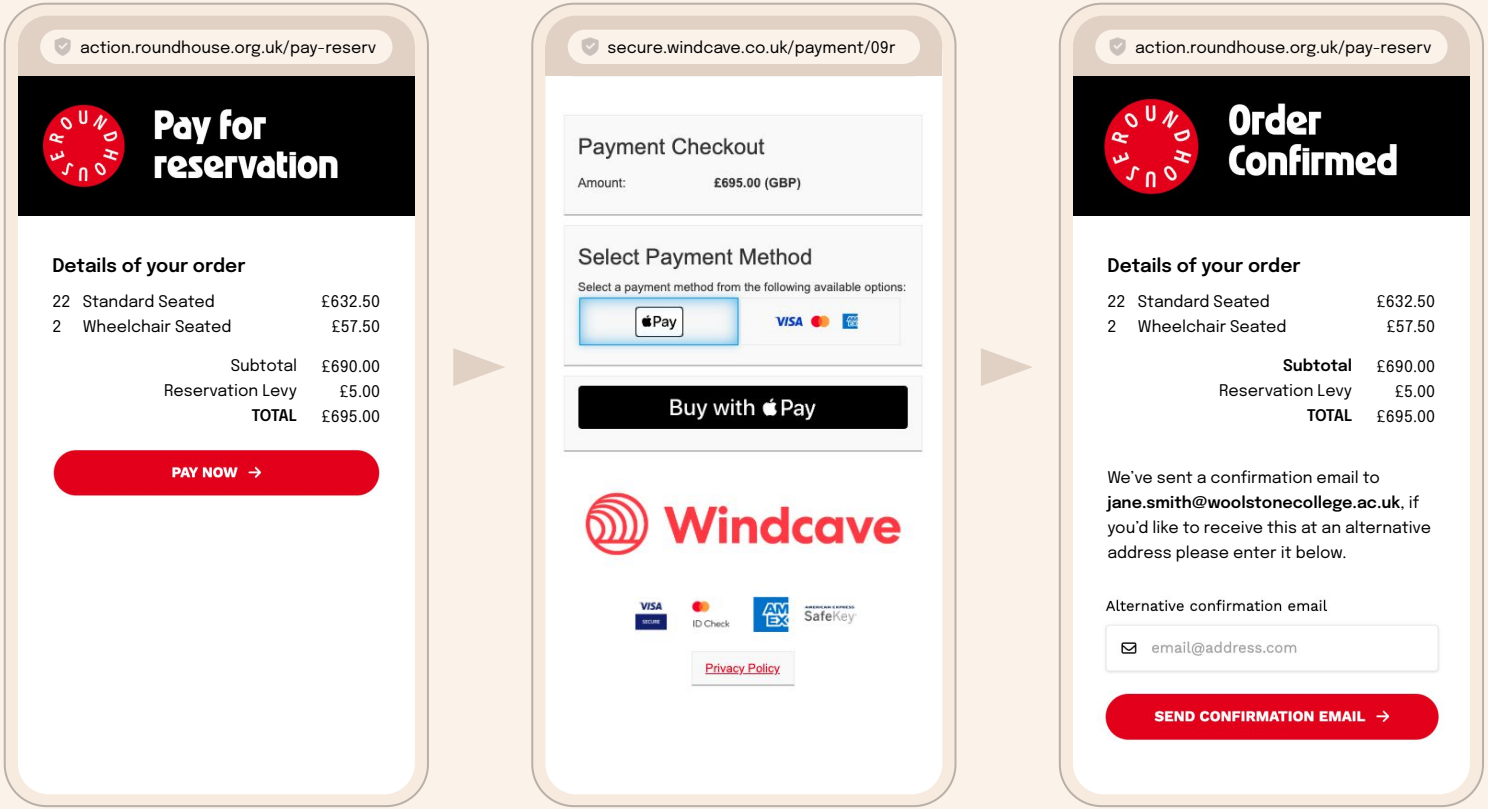
- How *agile* is *Agile*, really?
- Introducing *Maximum Potential Products*
- Hints & Tips toolkit

I built a thing called Action Links

Action Links is a self-service SaaS product:

- A tool to enable cultural organisations to build their own solutions to digital problems
- Subscribers can create Links (i.e. pages) that can display and collect data from users
- Each Link can be integrated with one or more systems, including Tessitura
- Clients are able to create as many Links as they like

Example of Action Links – paying-off reservations



Other examples of use cases

Simple use cases

- Mailing list sign-up
- Register for a free event
- Manage contact preferences
- Discount ticket scheme sign-up
- Competition entry
- Gift Aid opt-ins
- Quick links (e.g. add to cart)

More complex use cases

- Multi-step event enrollment
- Paying-off reservations
- Upgrade tickets to memberships
- Ticket returns
- Group visit management
- Single-use promo codes
- Digital signage

Building complex products can be challenging.

What did I learn from building Action Links?

- Complex projects can be daunting, especially if you're having to learn things from scratch
- When you don't have hands-on support, breaking down concepts into layers can make them easier to understand
- When the scope of the project is huge, focusing on one thing at a time will eventually lead to results
- Thinking about a project in layers can help you explain what it is and how it works, both to yourself and to others

What is layered design?

It’s about framing what you do in layers so that it’s **easier to understand and take into account the structure of your projects as a whole**.

It’s also about trying to be **more flexible**, **working in a more *agile* way** and increasing the scope for **efficiency**, **ongoing improvement**, and **future innovation**.

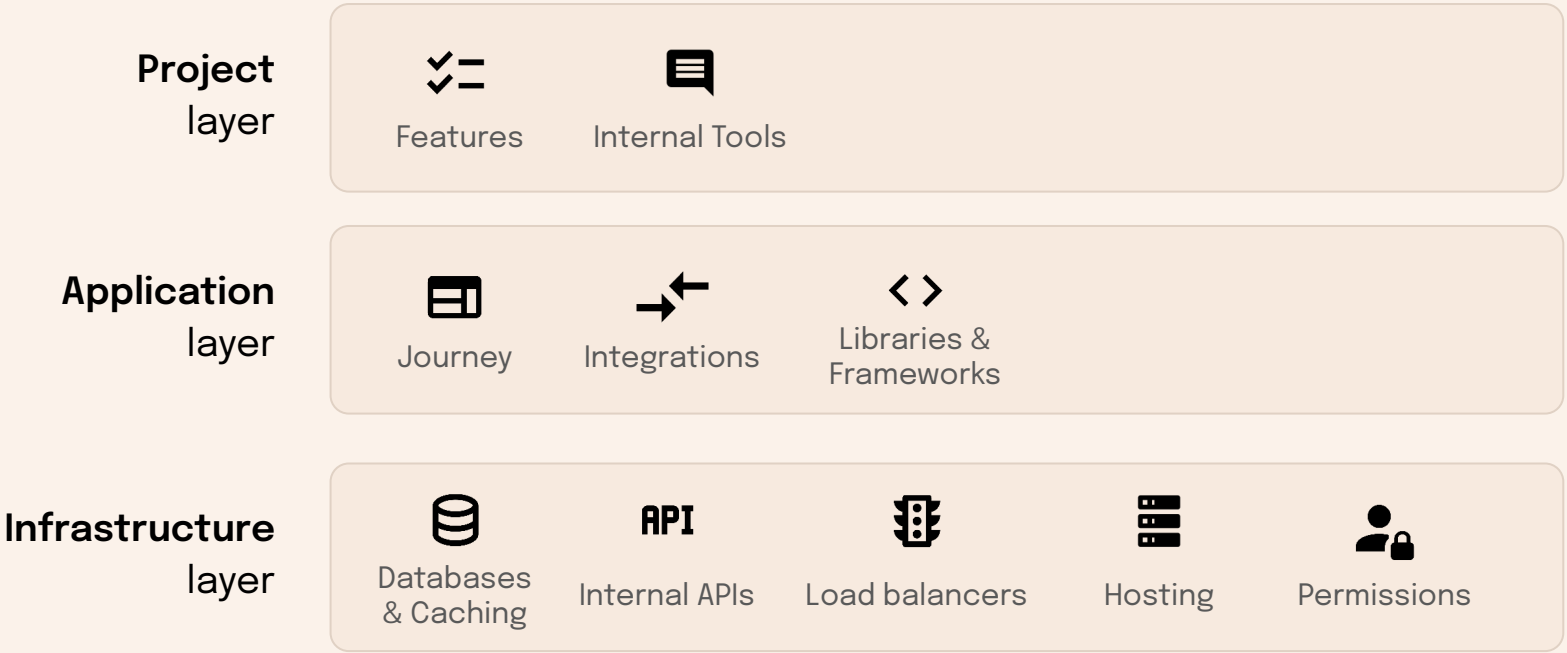
How do you apply ‘layered design’?

- Think about your project in three fundamental layers: the **Project** layer, **Application** layer, and **Infrastructure** layer.
- **Layers should be kept fairly broad**, so it’s possible to relate them back to the overarching project.
- Where possible you want to **remove layers that add complexity**.
- Think in the long-term – evaluate whether these layers ***unlock*** **rather than *undermine* future development**.
- **Allow for discoveries alongside development**, and be guided by those discoveries – **don’t be afraid to make significant u-turns**.

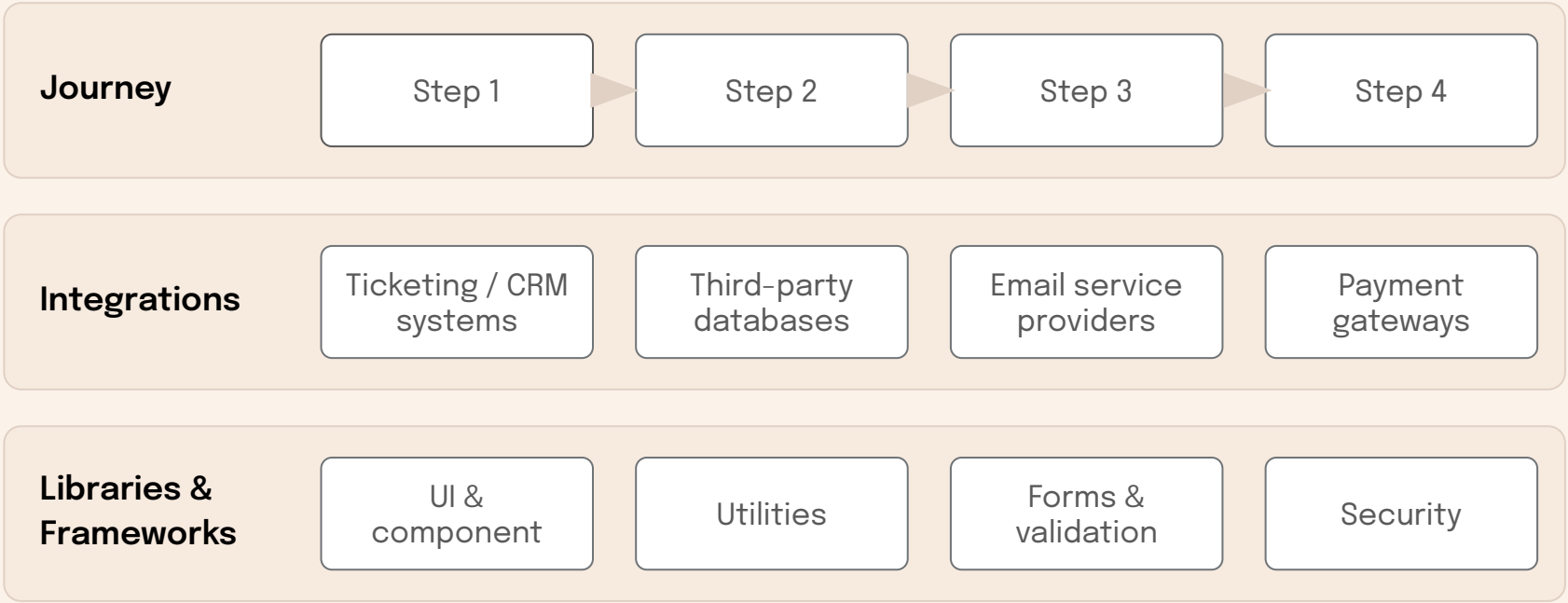
How can ‘layered design’ help us?

- Make complex projects more achievable
- Make sure your team are all on the same page
- Identify problems before they cause issues
- Make your activity more efficient
- Encourages more robust and secure products
- Encourage iteration and ongoing development into the future

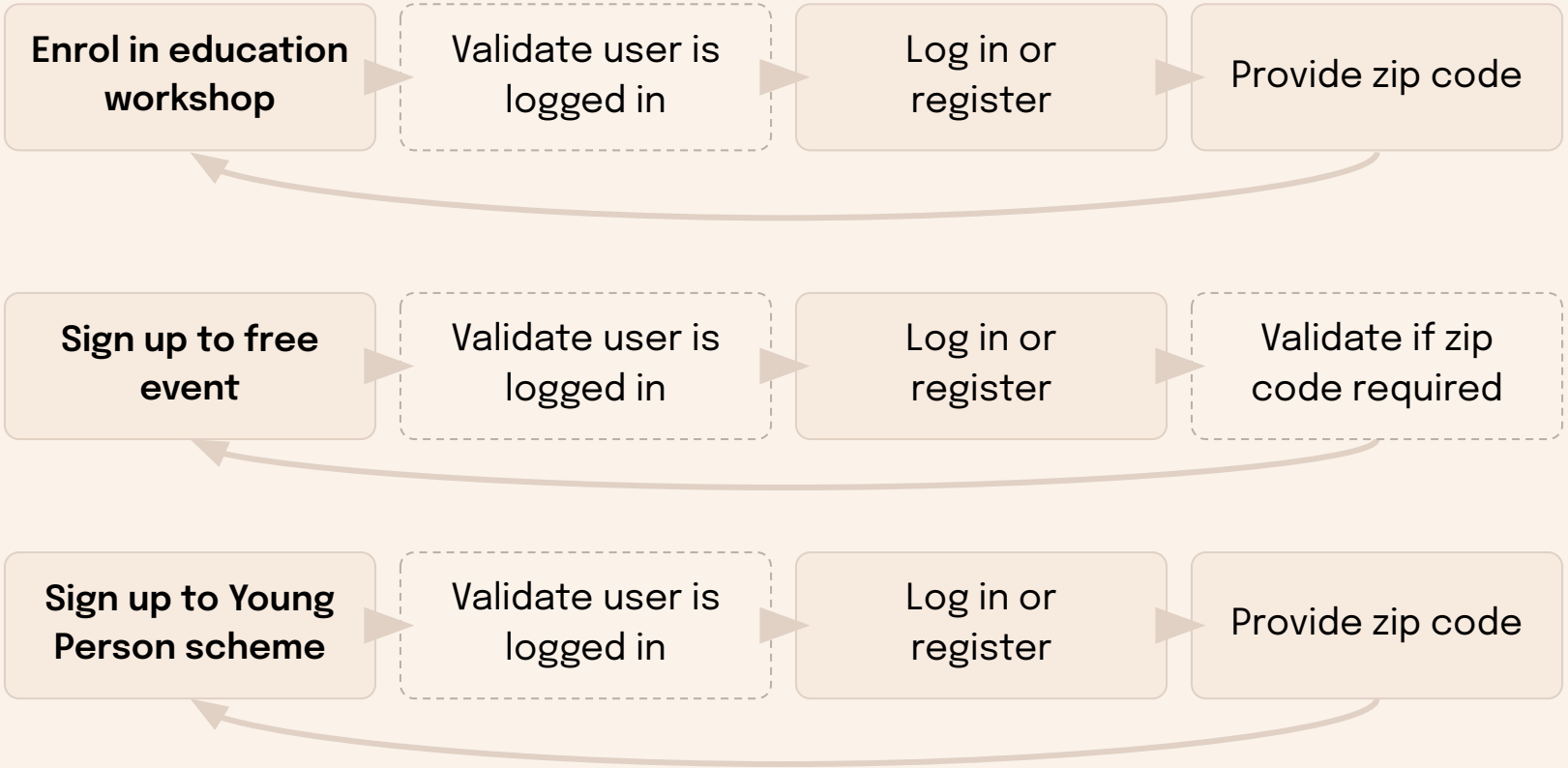
The layered project map



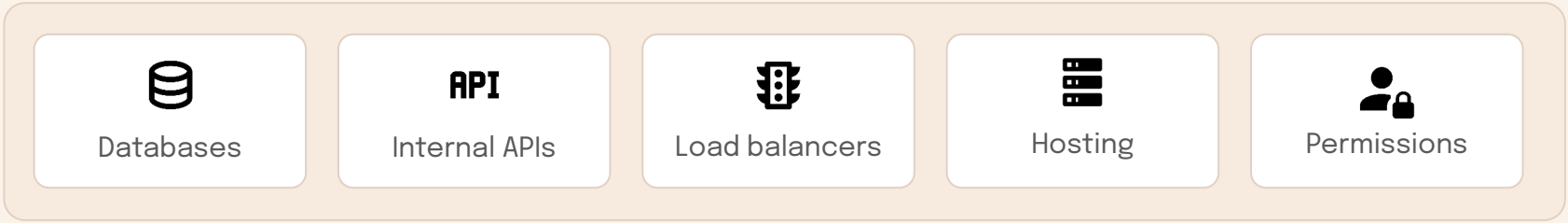
The *application* layer



The journey in your *application* layer



The *infrastructure* layer



Principles of the *infrastructure* layer

Keep the number of layers to a minimum – merge or remove layers where possible to:

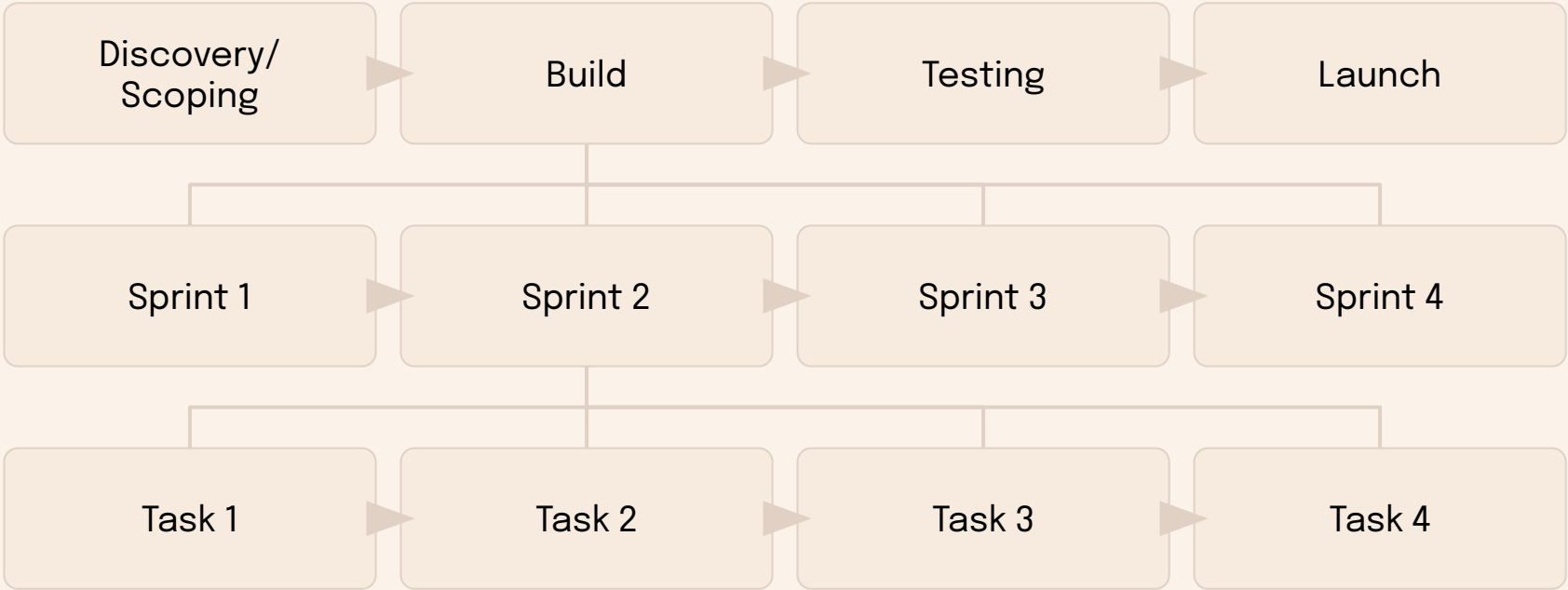
- **Make it easier to keep your application secure** – simplicity reduces the number of potential vulnerabilities, and makes it easier to manage permissions
- **Save on costs** – the fewer third-party services you use, the less it costs to run them
- **Reduce maintenance** – the greater the complexity, the harder it is to maintain and the greater the likelihood something will go wrong

The Action Links *infrastructure* architecture

[Diagram illustrating the architecture – removed due to sensitivity]

Traditional digital *project* layers

Often when we plan a digital project we end up with something like this:



The Agile Manifesto principles

Some of the key principles behind the Agile Manifesto (2001):

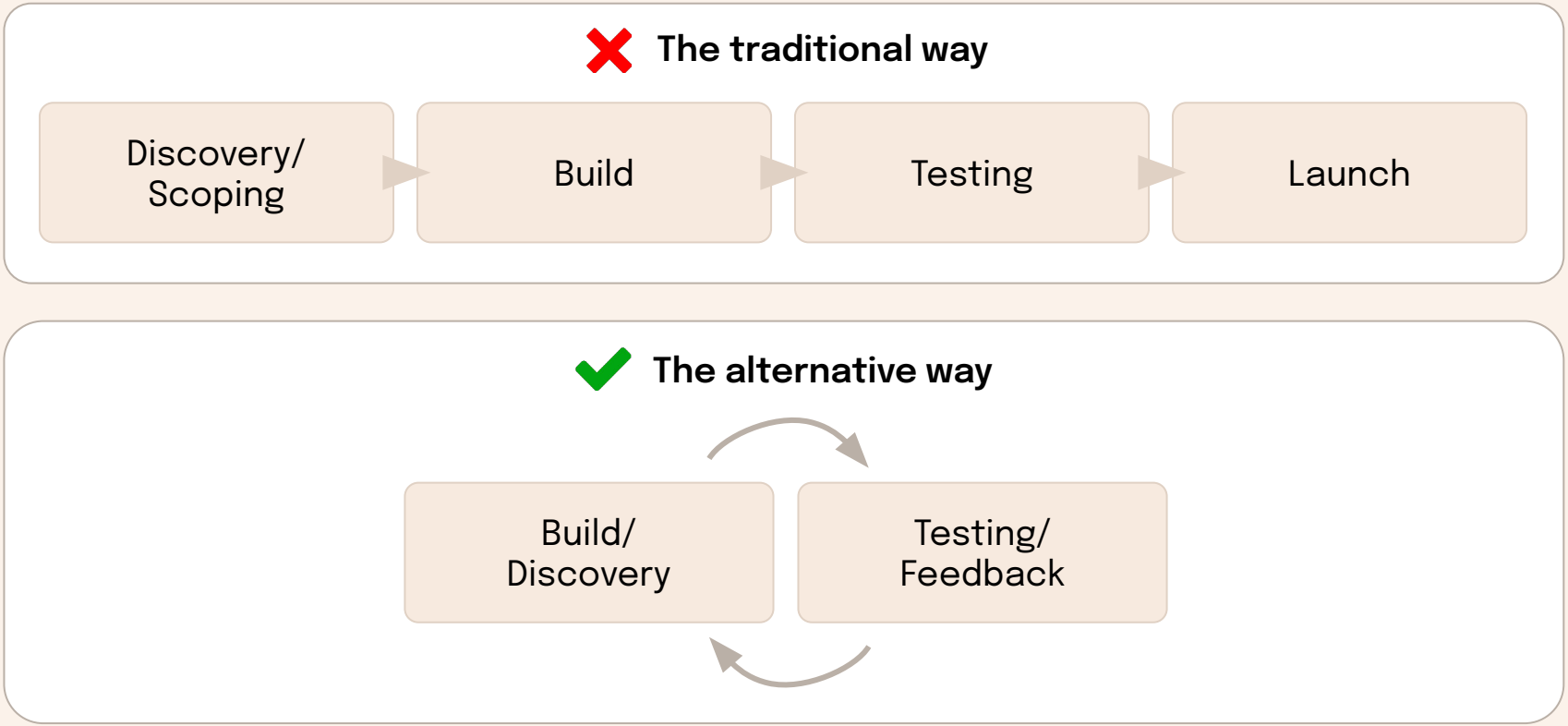
- Deliver working software frequently
- Working software is the primary measure of progress
- Welcome changing requirements, even late in development
- Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.

Are our projects truly Agile?

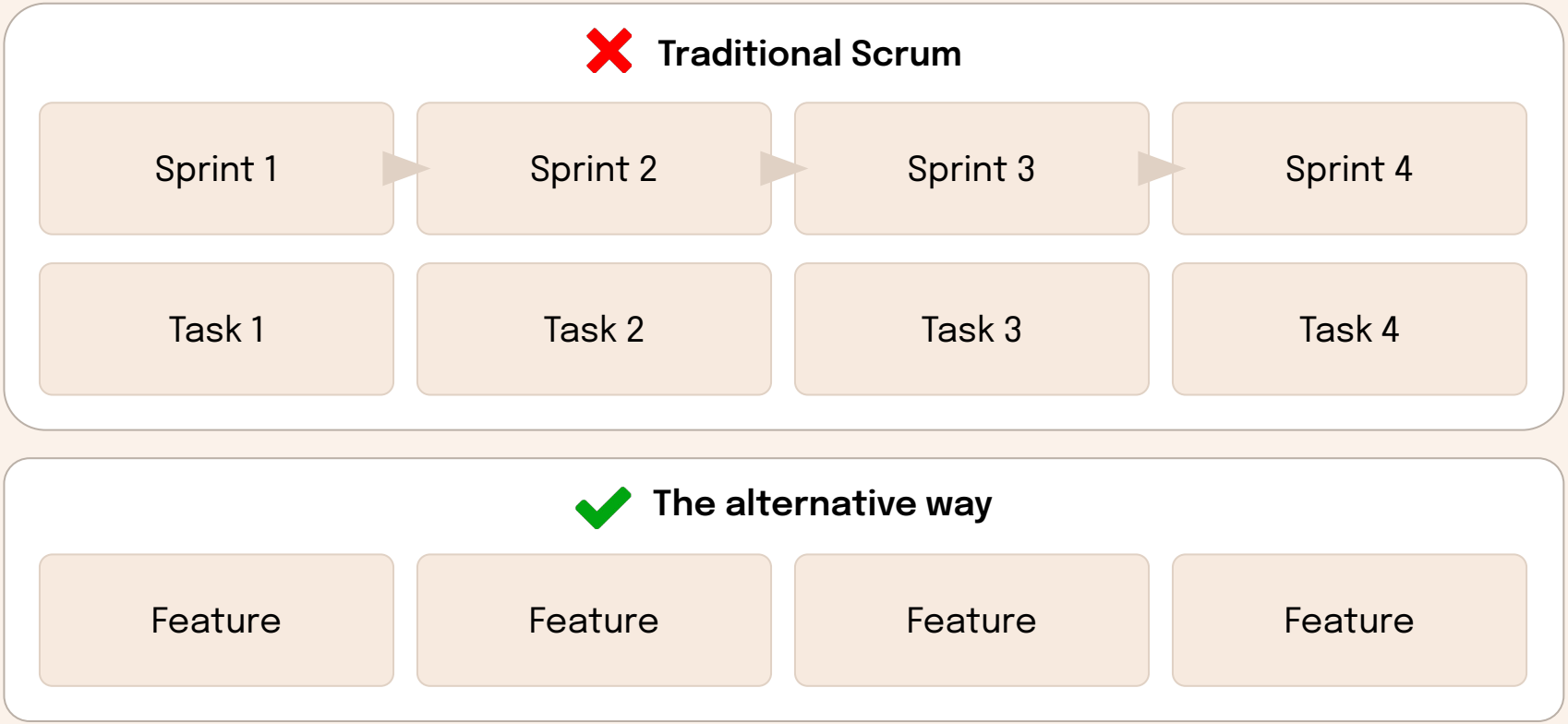
We have a fixation on setting deadlines for large projects. Our approach to projects generally:

- Relies on fixed timelines and go-live dates
- Have inflexible budgets
- Have inflexible project scopes, limiting our ability to respond to new discoveries

The layered design approach to projects



The layered design approach to projects



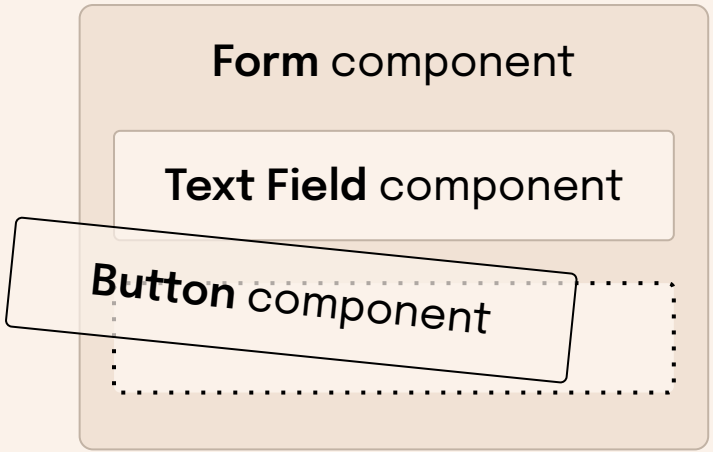
What does it mean to be truly agile?

This is my methodology:

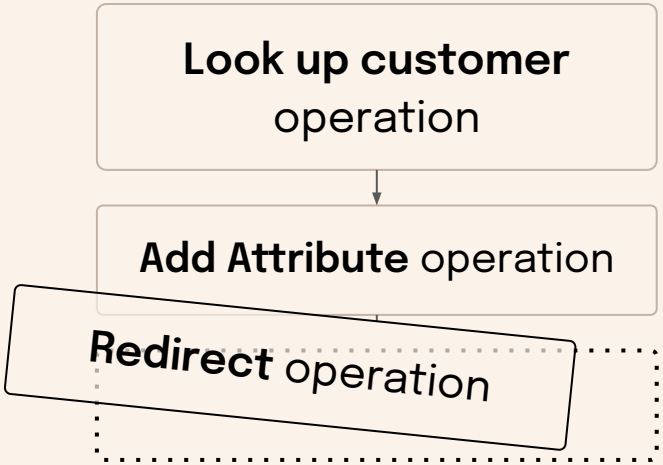
- You don't have to launch an entire project all at once – get it live sooner and iteratively improve it
- Avoid set deadlines, be fluid in how your prioritise – continually reassess what to work on and in which order
- Focus on the long-term rather than the short-term – make sure your products are fit for the future
- Combine the build with discovery – allow for new features and evolving user needs, and test as you go along

Don’t be afraid of a u-turn

Components
What the user sees when they look at your Link



Operations
What happens when the user interacts



Before we finish,
some other hints & tips...

The most important question is *why*…

- Never define ***what*** the project needs to deliver before you can define ***why*** it needs to deliver it.
- Never procure third-party support before you understand why you need them.
- Keep asking *why* at every stage of the project – the answer should evolve over time.
- ‘Discovery’ shouldn’t be about discovering *why*, it should be about discovering *why else*.

Throw the roadmap away

- Roadmaps force you to plan ahead as if nothing is ever going to change on the road.
- To be truly agile is to not make assumptions about the future, but to allow for the unknowns.
- Don’t make promises you can’t keep – you’ll always regret it.
- Building successful products is less like a road and more like a journey through space.

Users are unreliable witnesses

Don’t believe everything they say…

- They often want to please – sometimes they’ll tell you what they think you want to hear.
- They may be accessing your product in a way they’re not used to.
- Narrating what you’re doing makes you more aware of what you’re doing, it becomes less subconscious.
- Don’t focus on *what* they say they want, consider *why* they say they want it.

Scope creep (is/can be) a good thing…

… as long as you know why the scope has changed.

- It means you're continuing to ask *why*.
- It can be necessary to improve the product in response to feedback.
- The earlier you can identify a major issue, the less the impact it will have.

Challenge the status quo

- *Siloed teams are a bad thing*
  - ⇨ Involve as few people as possible in the day-to-day of the project (too many cooks)

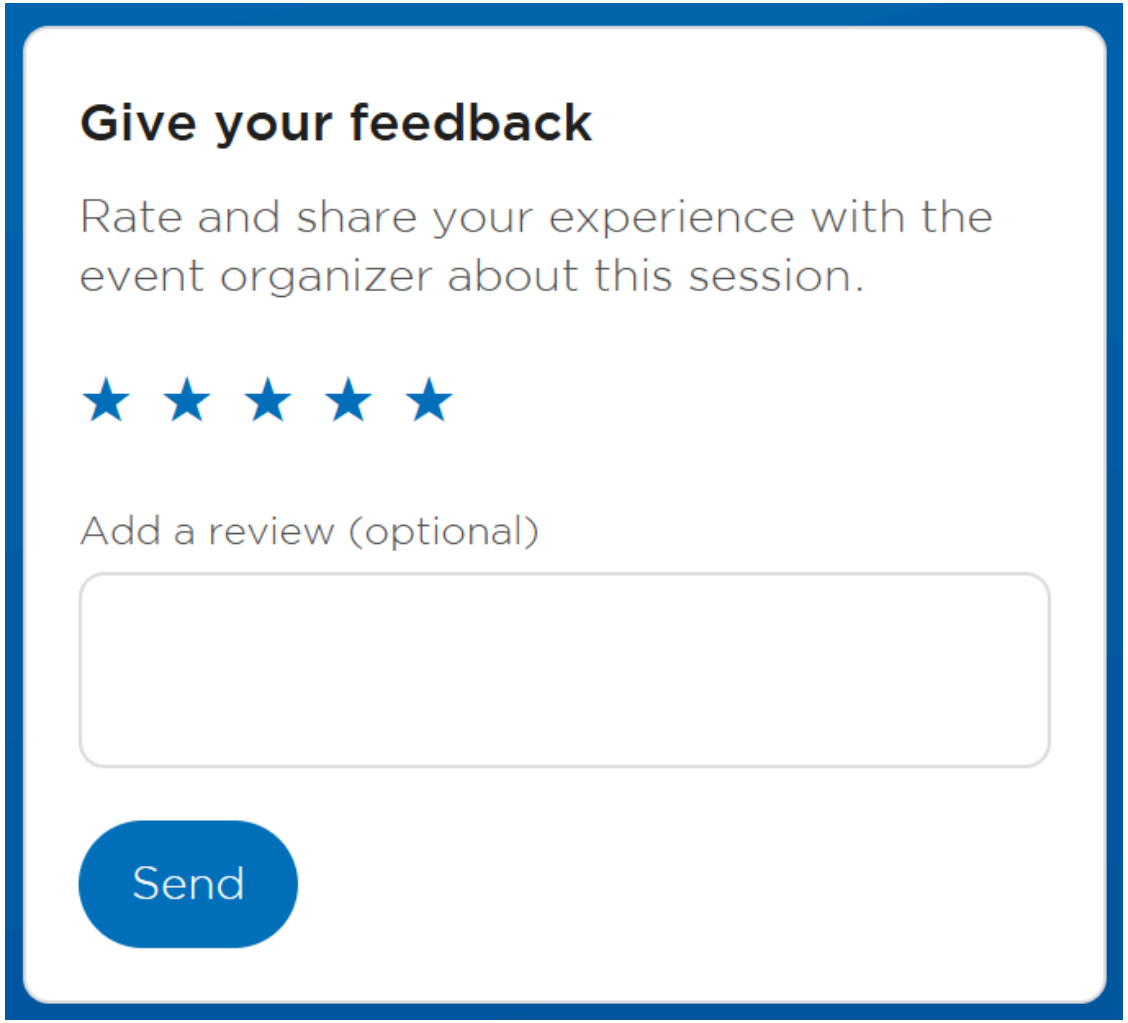
- *A bad workman always blames his tools*
  - ⇨ Few things can limit your potential for success than using the wrong technology

Any questions?

Your opinion matters!

★★★★★

Complete the short in-app survey



ANAHEIM TLCC 2025

Q&A

Please use a microphone so that everyone
in the room can be part of the conversation

ANAHEIM
TLCC
2025