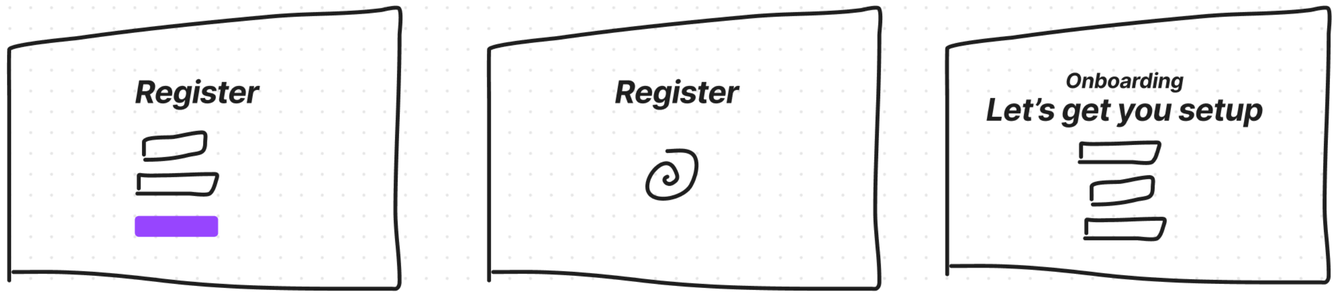
The
SOFTWARE
ESSENTIALIST

Metaphysical Essential: The Feedback Loop

Guesses & The Guess Points

essentialist.dev

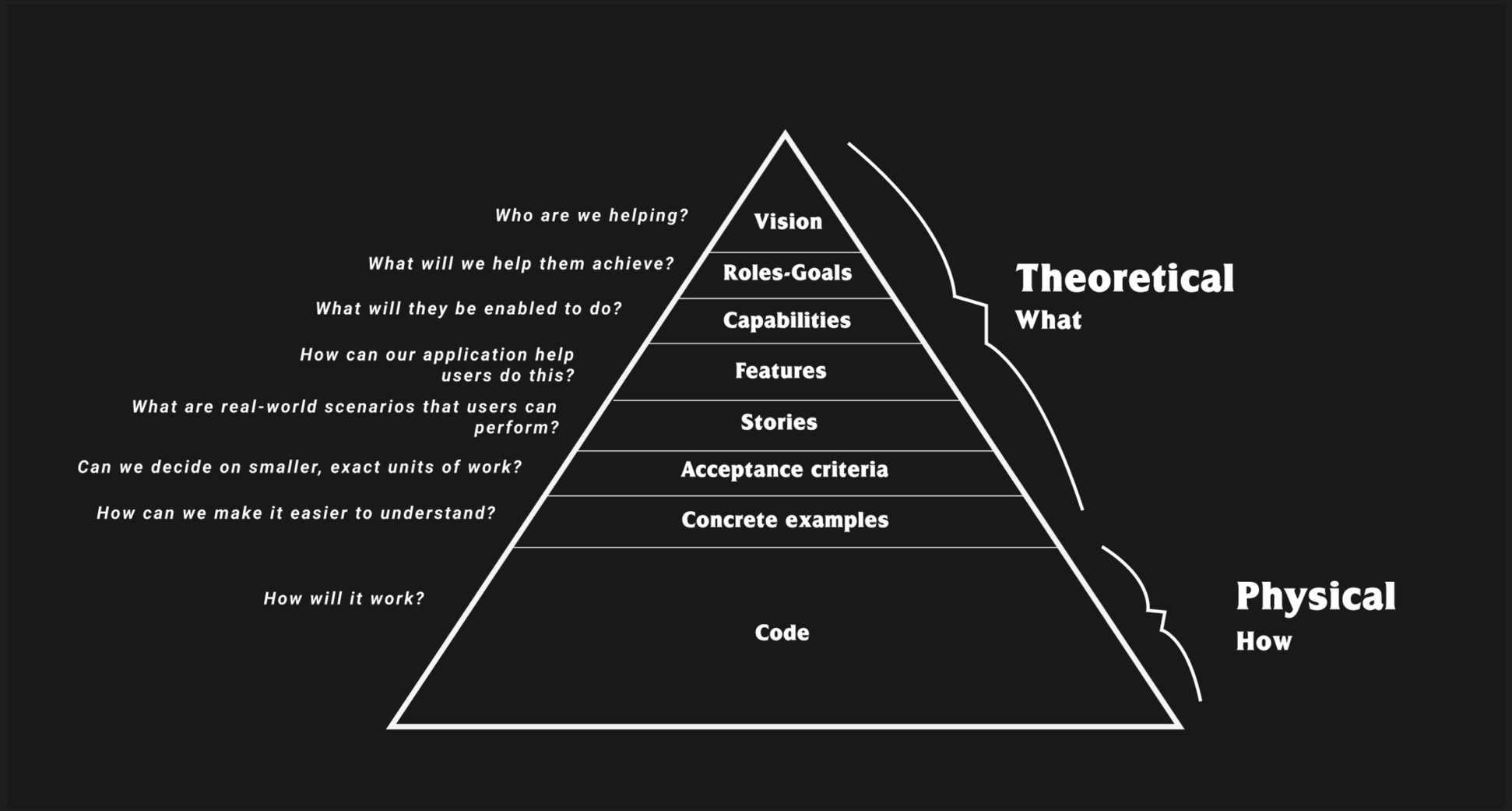
Registration



The reality?

Think about all the work that goes into a single set of interactions from frontend to backend to "register", for example

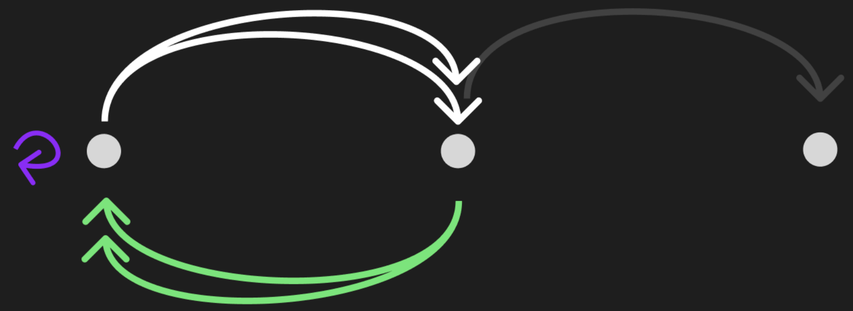
- *Presenting the form*
- *Maintaining the state of the form*
- *Validating the form locally*
- *Marshalling the data into an api call*
- *Making the request*
- *Displaying a loading spinner*
- *Presenting an error if failed*
- *Presenting the failed form fields*
- *Resetting the form fields for a new submission*
- *Redirecting to the onboarding page when new user*
- *Knowing if they've completed onboarding or not (redirect to / dashboard)*


Who are we helping?
Vision
What will we help them achieve?
Roles-Goals
What will they be enabled to do?
Capabilities
How can our application help users do this?
Features
What are real-world scenarios that users can perform?
Stories
Can we decide on smaller, exact units of work?
Acceptance criteria
How can we make it easier to understand?
Concrete examples
How will it work?
Code
Theoretical
What
Physical
How

The reality?

We're going to make more mistakes along the way than successes.

Consider this: everything that we do in Software Development is one big Guess.



Metaphysical Essential

The Feedback Loop

What we'll cover

- *What is the feedback loop & why is it so important?*
- *How we can use The Guess Points to practice The Feedback Loop, making it much easier to solve problems and implement your guesses on any side of the stack*

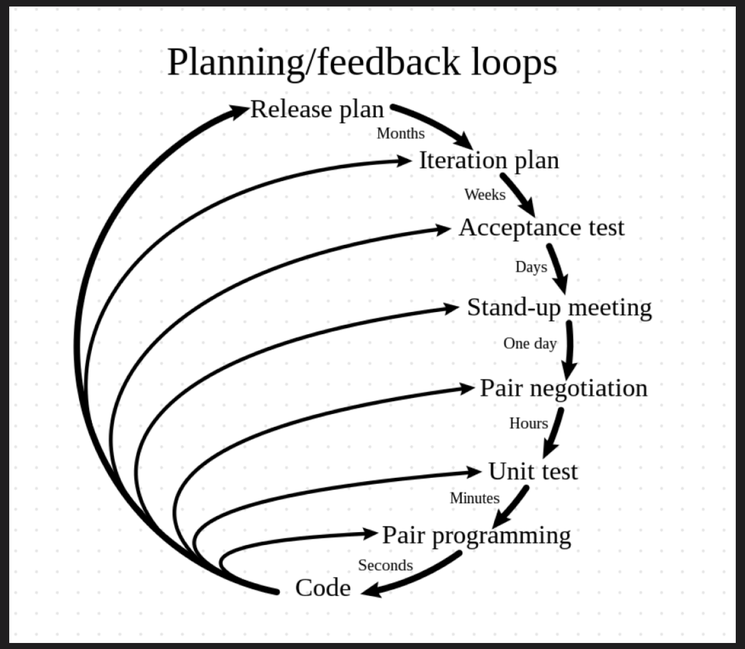
So what are feedback loops?

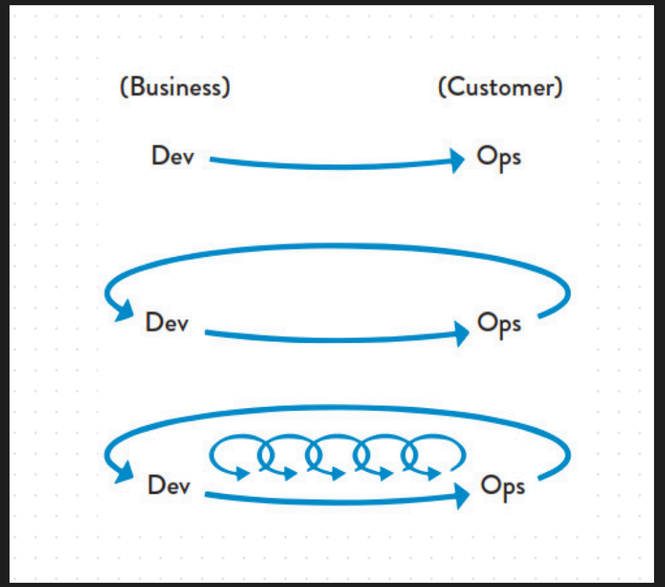
Three Ways

Value-driven software development is a Guess. Every step of the way to working software is uncertain. We need to use Feedback Loops to make the journey.

- ***Feedforward*** *Implement the guess*
- ***Feedback*** *Learn from the guess*
- ***Refinement*** *Make better guesses, faster*

It’s always been a guessing game

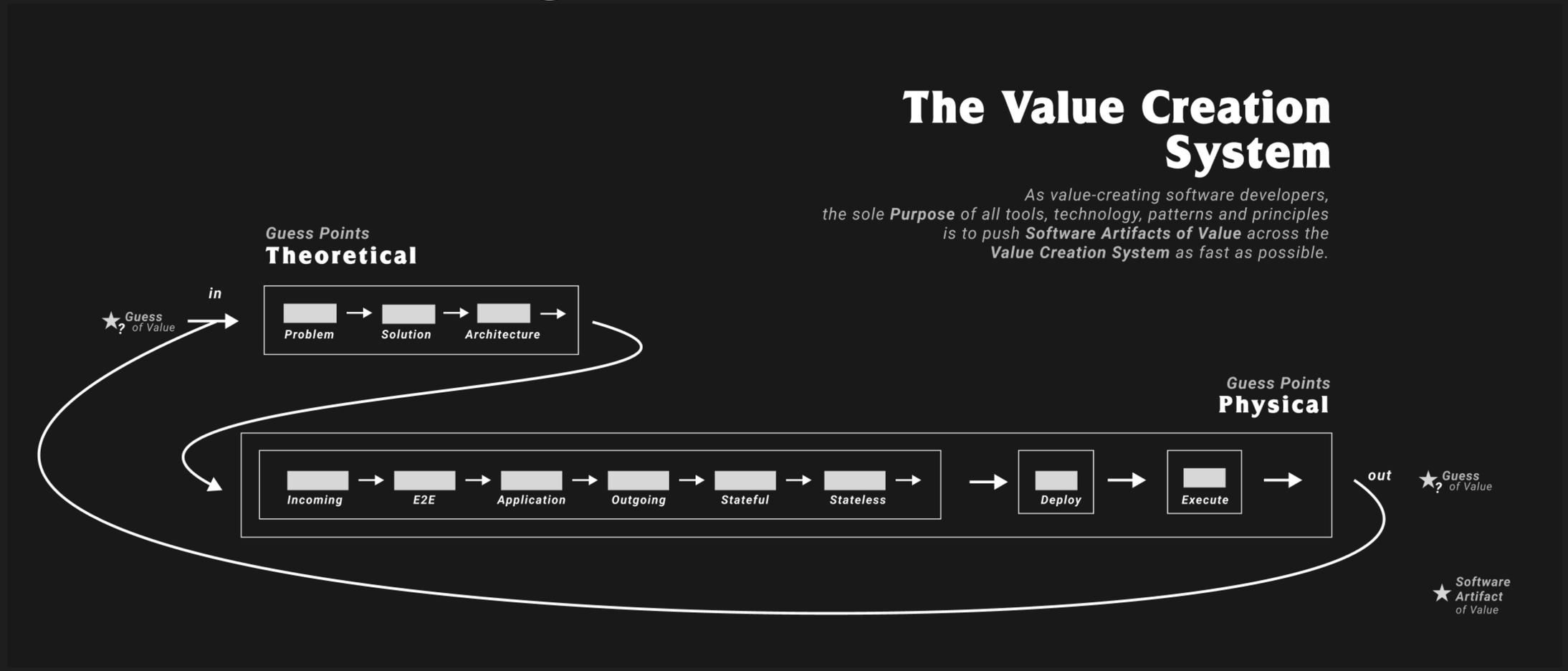




Challenges implementing Guesses

- *1) When we don’t know what Guesses we’re making*
- *2) When we don’t know how to validate a Guess*
- *3) When we’re not using tight enough feedback loops (getting the answers to the Guesses very late)*

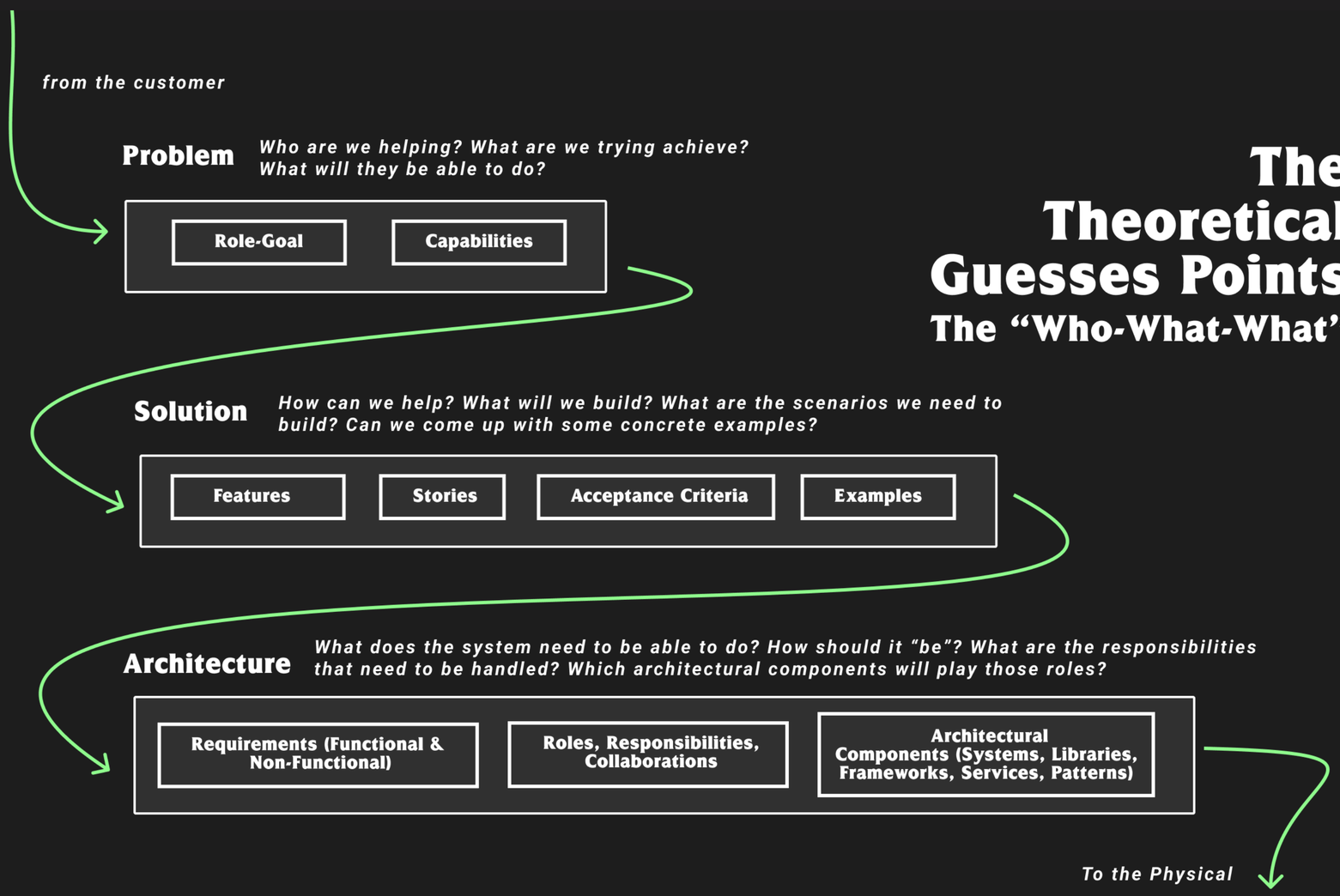
Introducing The Guess Points



essentialist.dev

The Theoretical Guesses Points

The "Who-What-What"



from the theoretical

The Physical Guess Points

The “How”

Incoming Adapter *Can the system be reached? Do the correct operations get called when a request is made?*

- Acceptance Test
- Executable Specification
- Protocol Driver
- System API Contract

System (E2E) *Does the system do what the customer asked for? Does the entirety of the system work together? Do all the architectural components work together?*

- Features
- Stories
- Acceptance Criteria
- Examples

Application *Do the internals of the system do the right things when we perform scenarios and edge cases? Are internals being called properly? Does the app call the right external services and attempt to save at the right times?*

- Features
- Stories
- Acceptance Criteria
- Examples

Stateful (Domain Modelling)

- ...

Are we accurately modelling the business logic and the heart of the domain? Does the application enforce business rules?

Outgoing Adapter

- ...

Can I reach the external services? Do they work the way I intend? Am I properly integrated with them? Do they persist data properly?

Stateless *Do my functions work correctly?*

- ...
- ...

Deployment & Delivery *Can I deploy to production? Does my deployment pipeline mitigate negative value? Does it enforce a code standard?*

- ...
- ...

Execution *Are users using the feature? Are they using it the way we intended?*

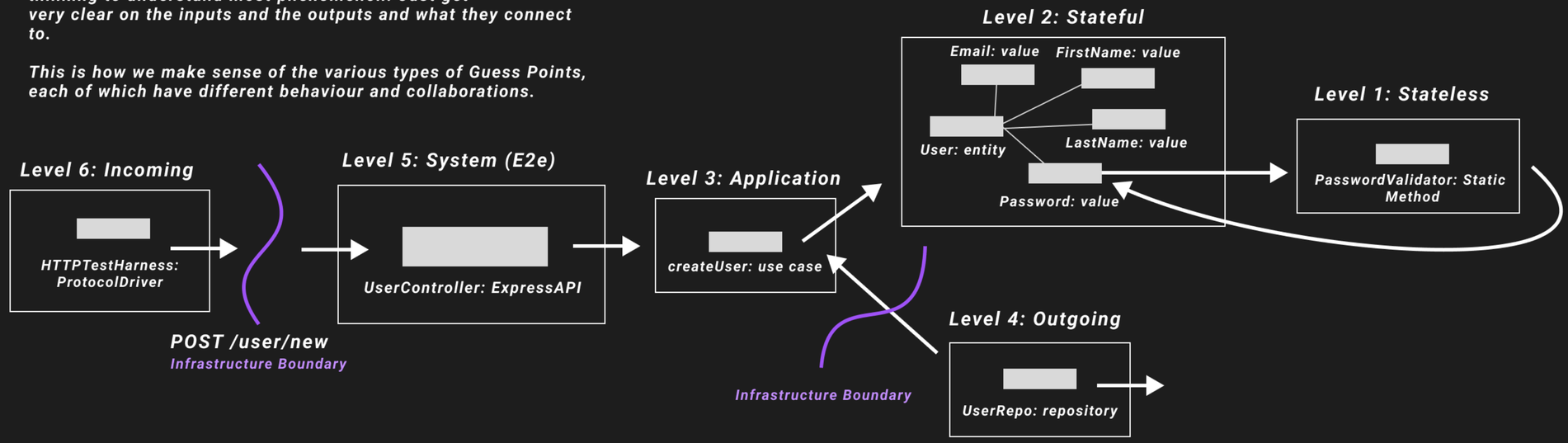
- ...
- ...

Mental Model: Systems Thinking & The Guess Points

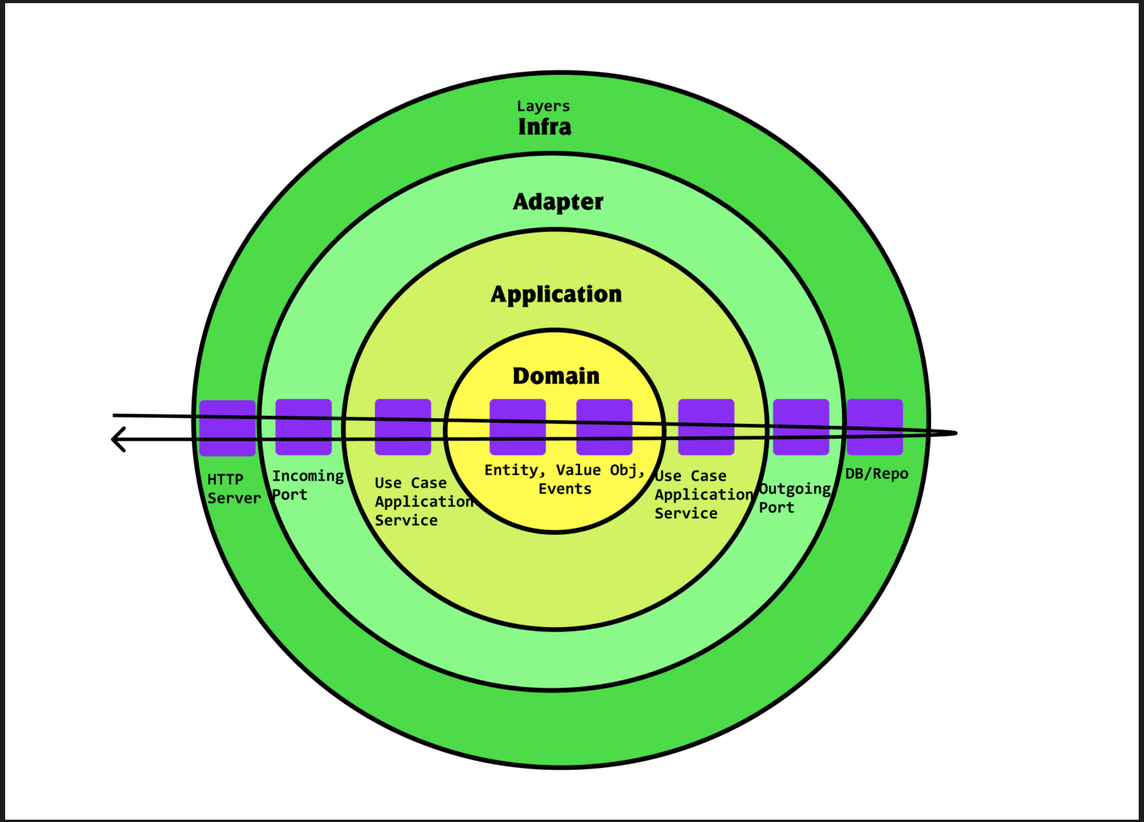
Understanding How & Where To Write Tests

You can use functional decomposition and systems thinking to understand most phenomenon. Just get very clear on the inputs and the outputs and what they connect to.

This is how we make sense of the various types of Guess Points, each of which have different behaviour and collaborations.

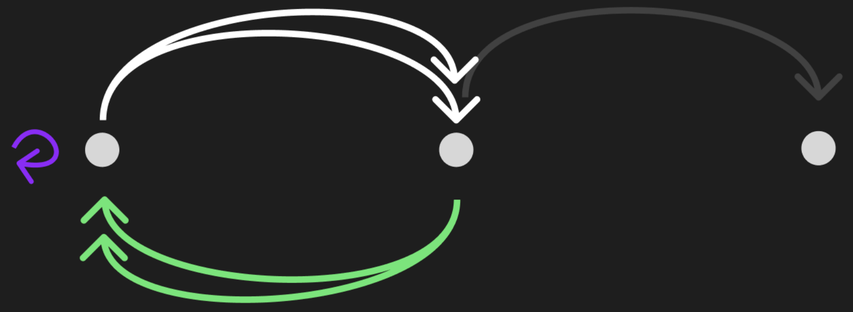


Q: How do you know the physical guess points? Where does that come from?



Q: Are there more guess points? Or is that it?

What we covered



Metaphysical Essential

The Feedback Loop

Every single aspect of software development is a Guess.

Do our API calls work?

Are we missing any edge cases?

Are we saving to the database properly?

Will this architecture serve us in the long-term?

Does this feature do what the Customer wants?

Will users actually use this?

It's all a Guess.

The only way to progress is 3 Ways: Feedforward, Feedback, and Refinement.

Declare the end result, take a leap forward, use the feedback to refine.