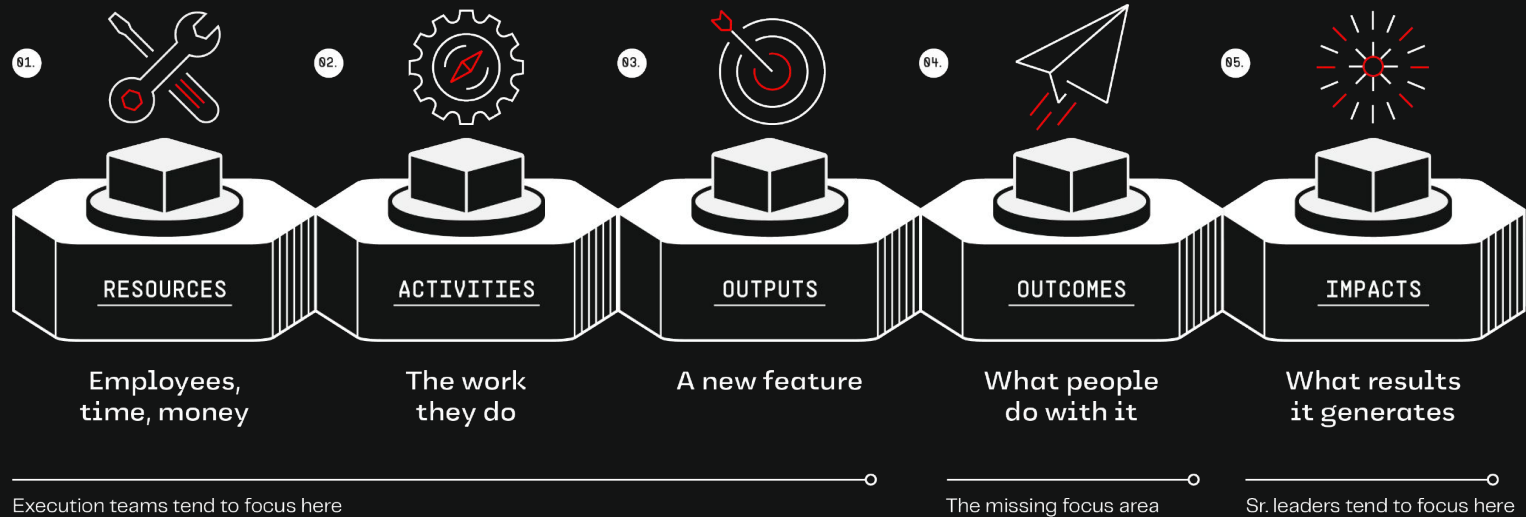


Outcome Factory

Shipping Software in the Age of AI

< /prodacity >

The Mission Logic Model



Adapted from Josh Seiden and the Kellogg Foundation

The Model Is Broken

Effort gets rewarded, not the mission.

The programs that fail aren't failing because nothing was built. They're failing because the model rewards effort, not outcomes. You're buying hours and hoping they become mission value.

The vendor gets paid either way.

87%

OF LARGE GOVERNMENT
SOFTWARE INITIATIVES **FAIL.**

Source: GSA / 18F, De-risking Custom Technology Projects
(Standish Group data)



Billed hours, not outcomes.

Vendors are incentivized to extend engagements. The contract rewards effort, not delivery.



ATO risk sits with you.

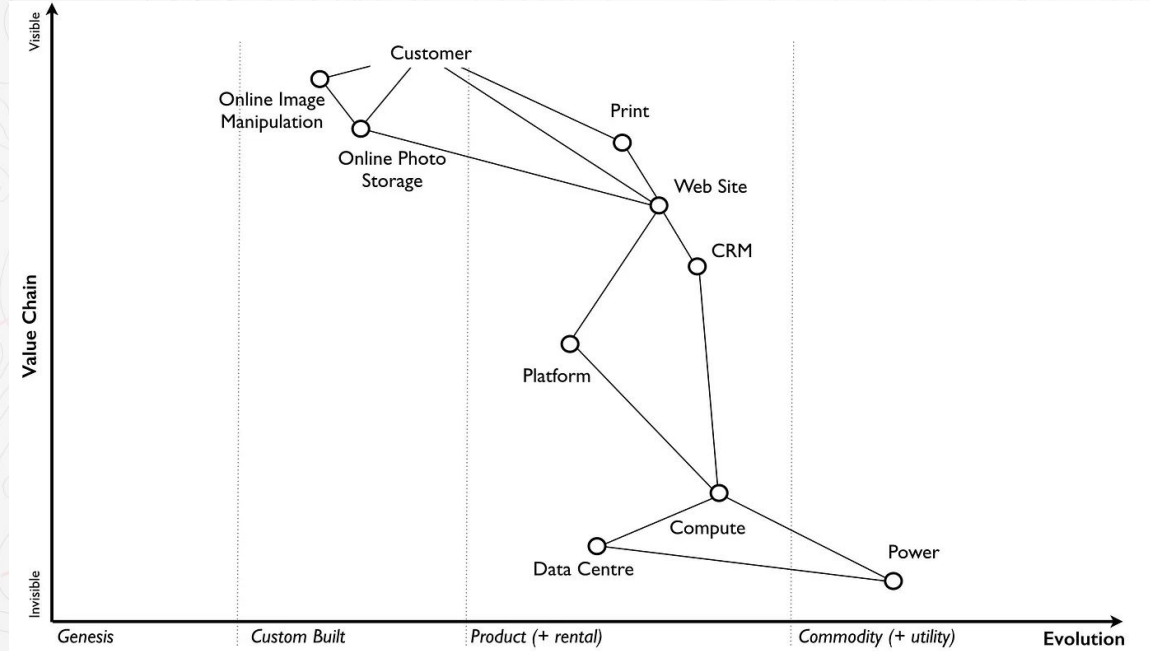
Compliance and safety requirements erode scope after contract award. You absorb the gap.



Adoption is also on you.

User adoption, mission impact, and ROI are never in scope. They become your problem after sign-off.

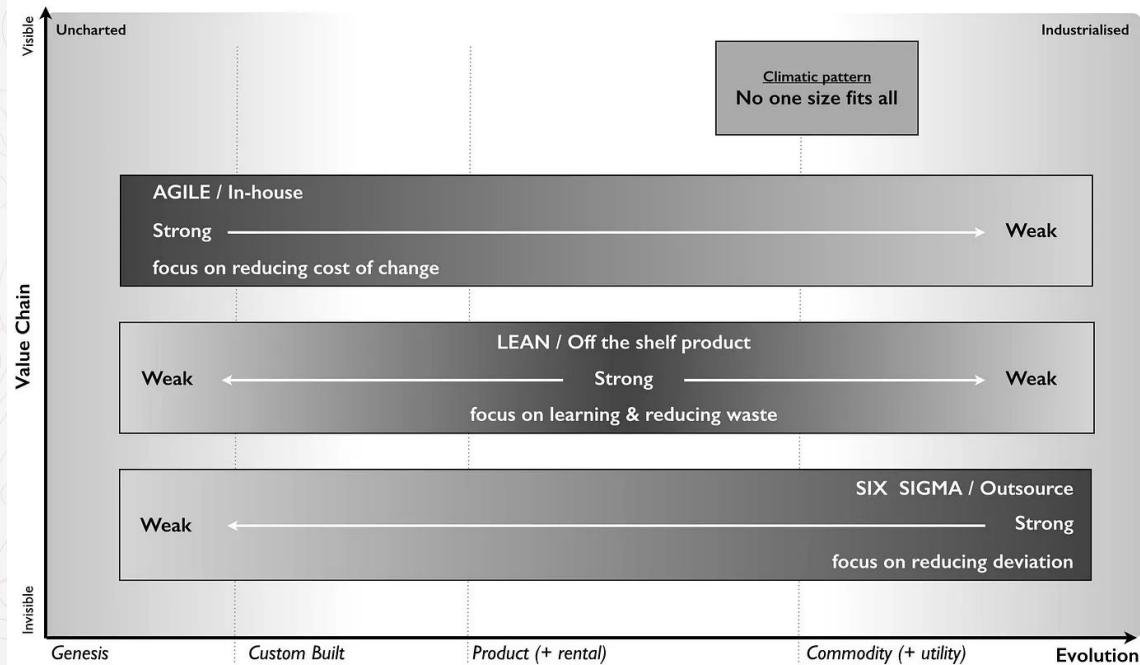
How To Decide Between Project / Experiment



Simon Wardley @ <https://medium.com/wardleymaps>



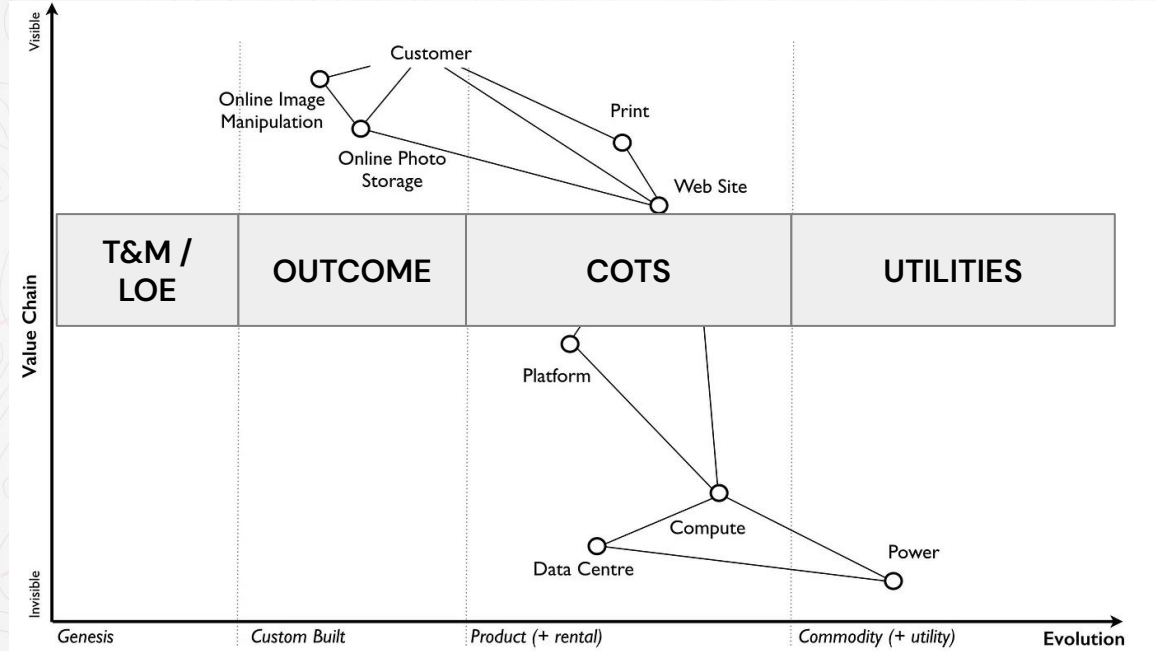
How To Decide Between Project / Experiment



Simon Wardley @ <https://medium.com/wardleymaps>



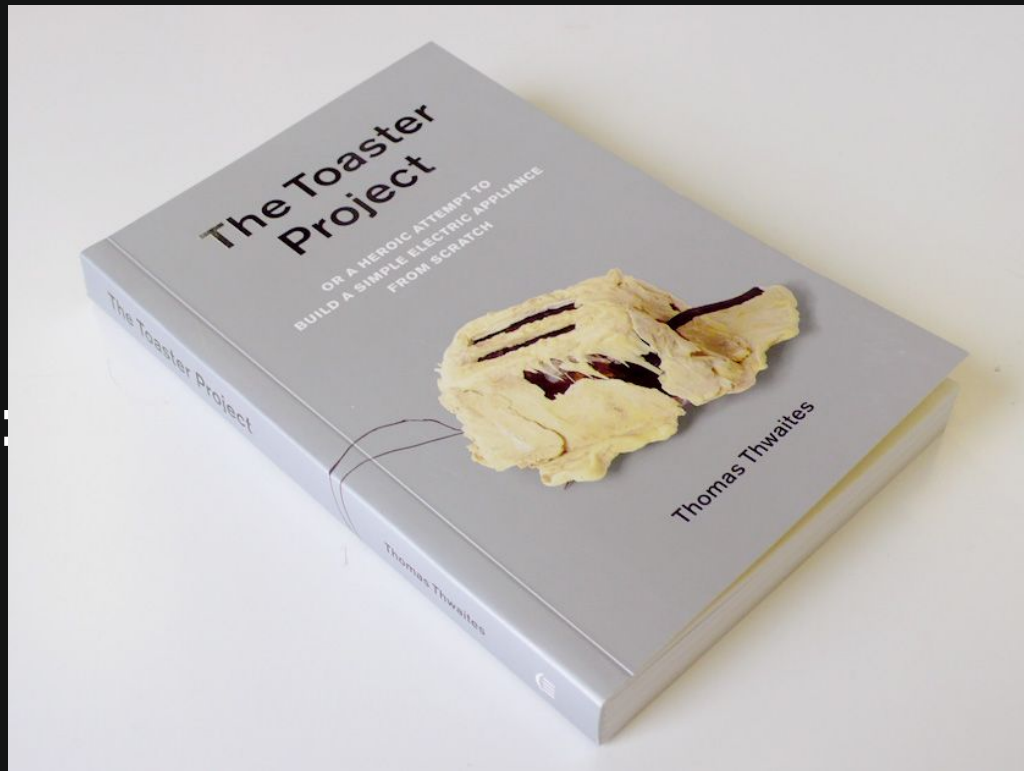
How To Decide Between Project / Experiment



Simon Wardley @ <https://medium.com/wardleymaps>



Value



Success

Price

Options On the Table



TRADITIONAL SYS INTEGRATOR

End-to-end “delivery”

One vendor owns the entire problem

✗ 4-5 year delivery timelines. Waterfall. When it ships, if ever, it doesn't meet the mission need



INTERNAL BUILD

Do it yourself

No vendor dependency

✗ Difficult to hire, train, or retain A-player talent. Clearances and comp make it nearly impossible.



STAFF AUGMENTATION

Bodies on contract

Flexibility, low barrier to entry

✗ You own all the delivery risk, but not the talent. Knowledge walks when the contract ends.



COTS, OFF THE SHELF

Pre-built, fast start

Speed to deploy, lower initial cost

✗ It never quite fits. Config costs exceed custom build. Locked into vendor's roadmap.



MAINTAIN THE STATUS QUO

Do nothing

No disruption to procurement

✗ Programs stall and eventually become unusable putting the mission, warfighters, and civilians at risk.

What Good Looks Like



Outcome-based accountability

Mission success isn't measured in features shipped or milestones hit. It's measured in mission outcomes delivered to real users in production.



Speed to mission value

First outcomes delivered in weeks not years. Real users. Real impact. Real result.



Elite technical & domain talent

Engineers, PMs, and Designers who know bleeding-edge tech and government insiders who know how gov and missions actually work. On every team.



Security built in from day one

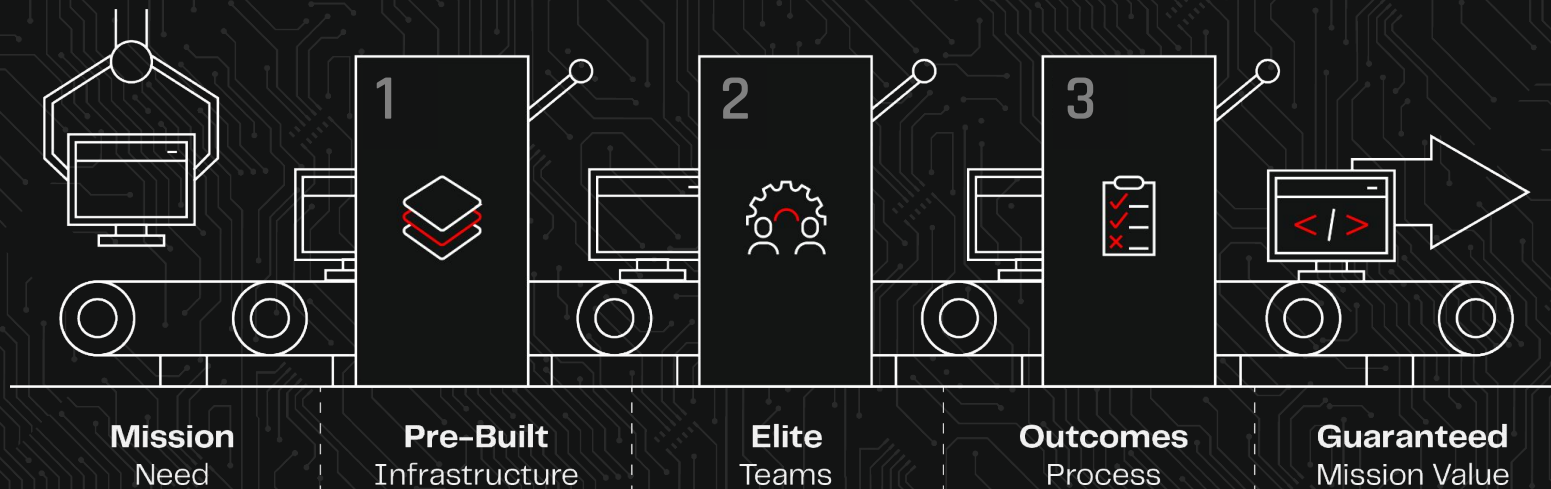
Compliance and authorization are engineered into delivery, not bolted on as a final gate.



Performance guarantee

Your first mission outcome should be in production within 180 days, guaranteed. If it takes longer, something is broken.

The mission outcome factory.



Ship the first mission outcome to the warfighter in **less than 180 days**,
with continuous delivery thereafter, **guaranteed**.

The mission outcome factory.

— INPUT

Mission
Need

Program objective

User problem

Operational gap

Compliance reqs



1. OUTCOME-BASED DELIVERY SYSTEM

Measured in mission impact. Not sprints. Not story points. Not status decks.



2. ELITE TEAMS

Engineers, PMs, and designers who know modern software and get your mission context.



3. PRE-BUILT INFRASTRUCTURE

Platform, pipelines, and path to production already built. Not starting from scratch.

— OUTPUT

Mission
Outcome

In production

Users adopt & love

Mission impact

180

GUARANTEED

Redefining Our Goals

Lead Time
Deployment Frequency

Speed

Mean Time To Restore
Change Fail Rate

Stability

Site Reliability

Reliability

Increase Mission Capability

Mission Metric that Matters

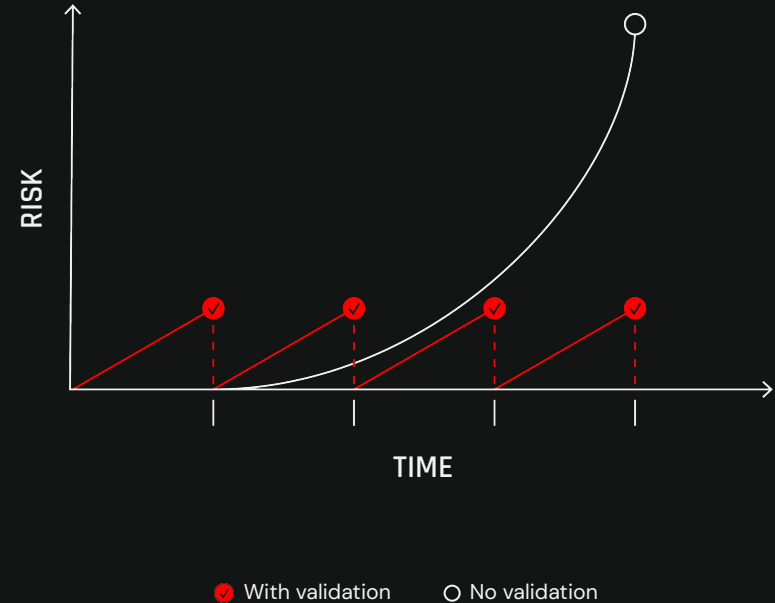
Elite Software Delivery Performance

PERFORMANCE LEVEL	DEPLOYMENT FREQUENCY	CHANGE LEAD TIME	CHANGE FAILURE RATE	FAILED DEPLOYMENT RECOVERY TIME
Elite	On demand	Less than one day	5%	Less than one hour
High	Between once per day and once per week	Between one day and one week	10%	Less than one day
Medium	Between once per week and once per month	Between one week and one month	15%	Between one day and one week
Low	Between once per week and once per month	Between one week and one month	64%	Between one month and six months

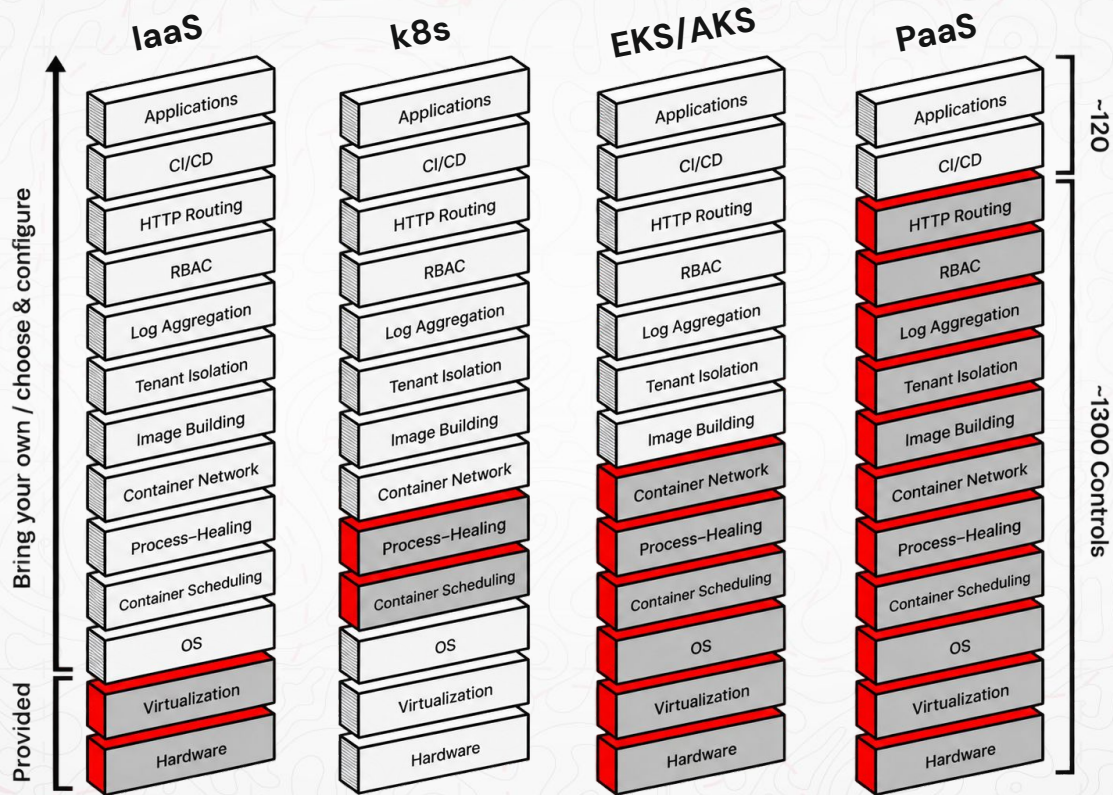
Continuous Delivery

Continuous Delivery (CD) is the ability to get features, configuration changes, bug fixes, and experiments into production **safely** and **quickly** in a **sustainable** way.

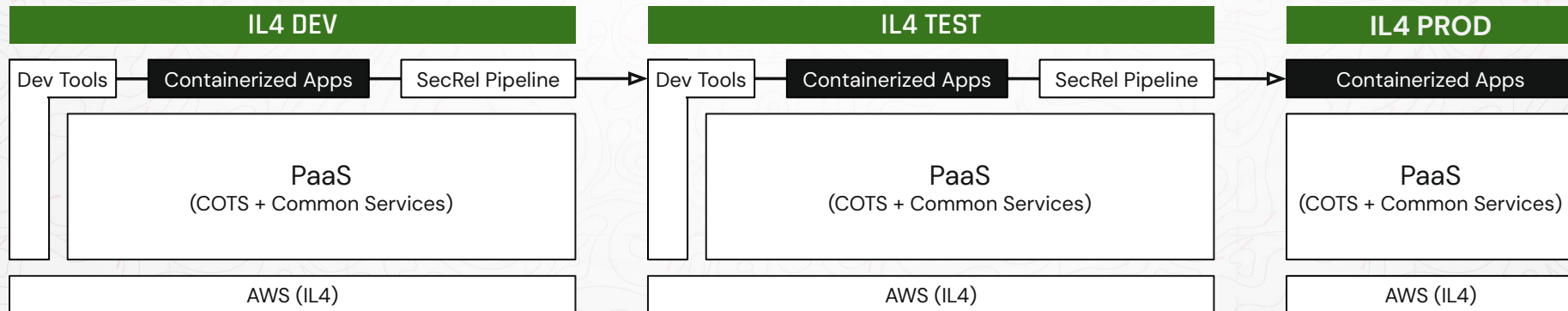
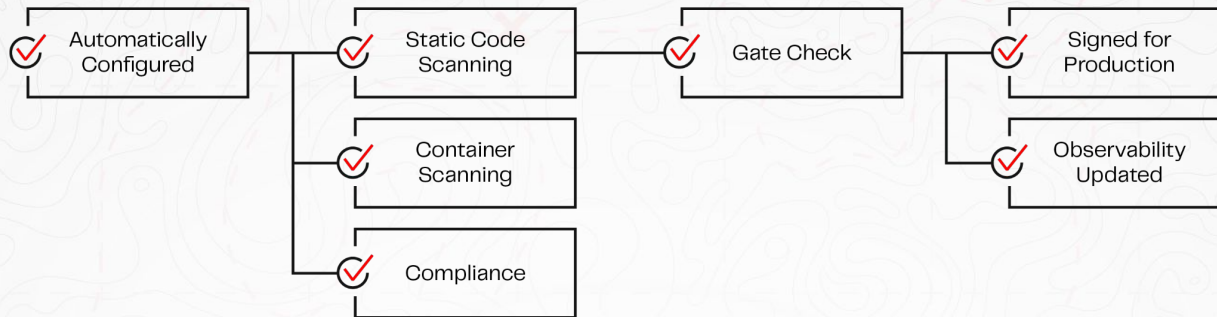
Production is not an arbitrary designation, it is the setting where software is put into **operation** for its **intended uses** by **end users**.



Structured & Opinionated PaaS



Continuous GRC & Paved Path-to-Production



The mission outcome factory.

— INPUT

Mission
Need



1. OUTCOME-BASED DELIVERY SYSTEM

Measured in mission impact. Not sprints. Not story points. Not status decks.



2. ELITE TEAMS

Engineers, PMs, and designers who know modern software and get your mission context.



3. PRE-BUILT INFRASTRUCTURE

Platform, pipelines, and path to production already built. Not starting from scratch.

— OUTPUT

Mission
Outcome

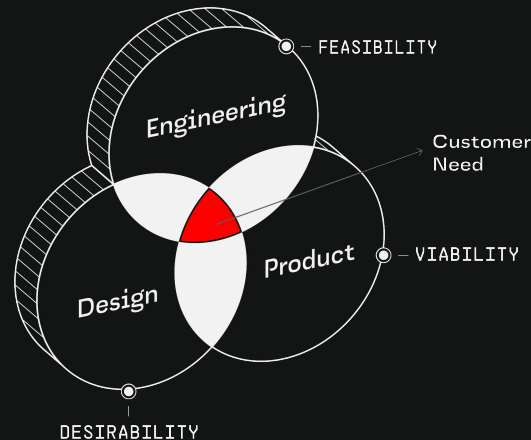


Balanced Teams

Start small

Small, balanced teams of product, design, and engineering—accounting for mission, users, and architecture—create the best outcomes with the least risk.

Start with 1–3 teams deploying real working software to solve a real problem for a real user in a real operations environment.



PRODUCT

PMs lead a team to deliver a product that creates meaningful mission value.

Masters of lean enterprise and own the value stream. They ask, "Will it help the mission?"



DESIGN

Designers focus on the collective experience of the product.

Practice human centered design and design thinking. They ask, "Does this solve a real user problem?"



ENGINEERING

Engineers champion all technical decisions for the product.

Practice Extreme Programming (XP) and own the path-to-prod. They ask, "Is this feasible?"

The mission outcome factory.

— INPUT

Mission
Need



1. OUTCOME-BASED DELIVERY SYSTEM

Measured in mission impact. Not sprints. Not story points. Not status decks.



2. ELITE TEAMS

Engineers, PMs, and designers who know modern software and get your mission context.



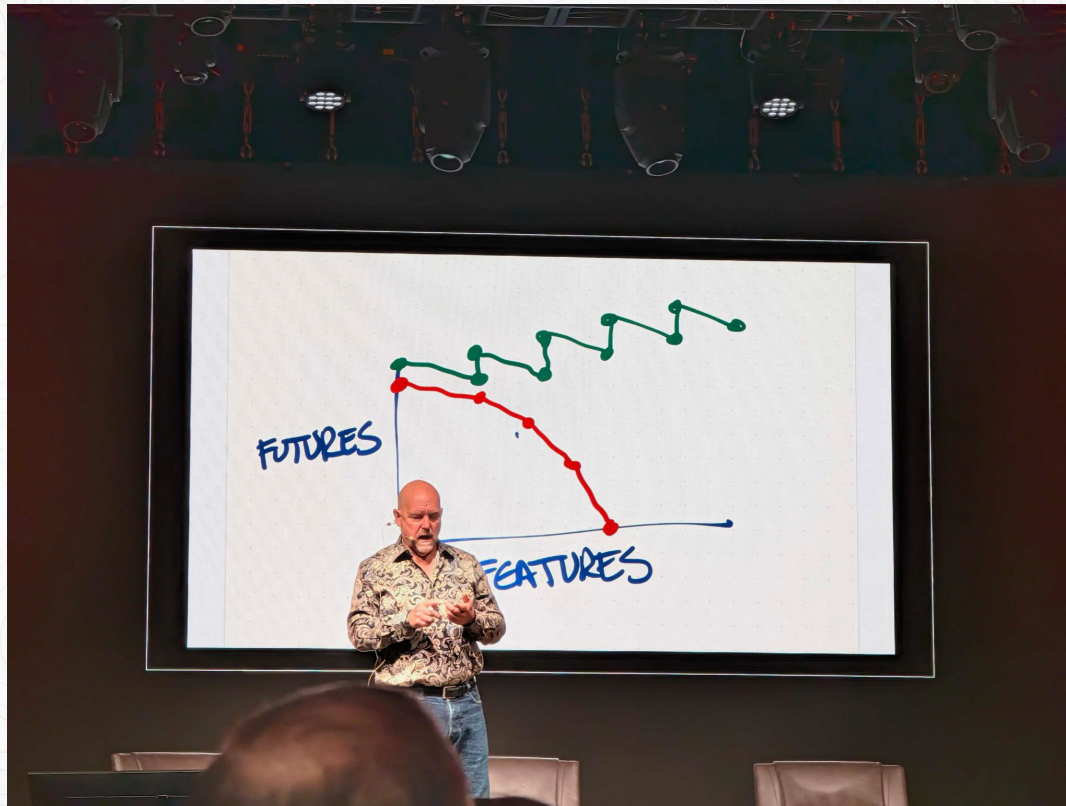
3. PRE-BUILT INFRASTRUCTURE

Platform, pipelines, and path to production already built. Not starting from scratch.

— OUTPUT

Mission
Outcome





Feature: Login

As a user

I want to be able to log in to the application

So that I can access my account information

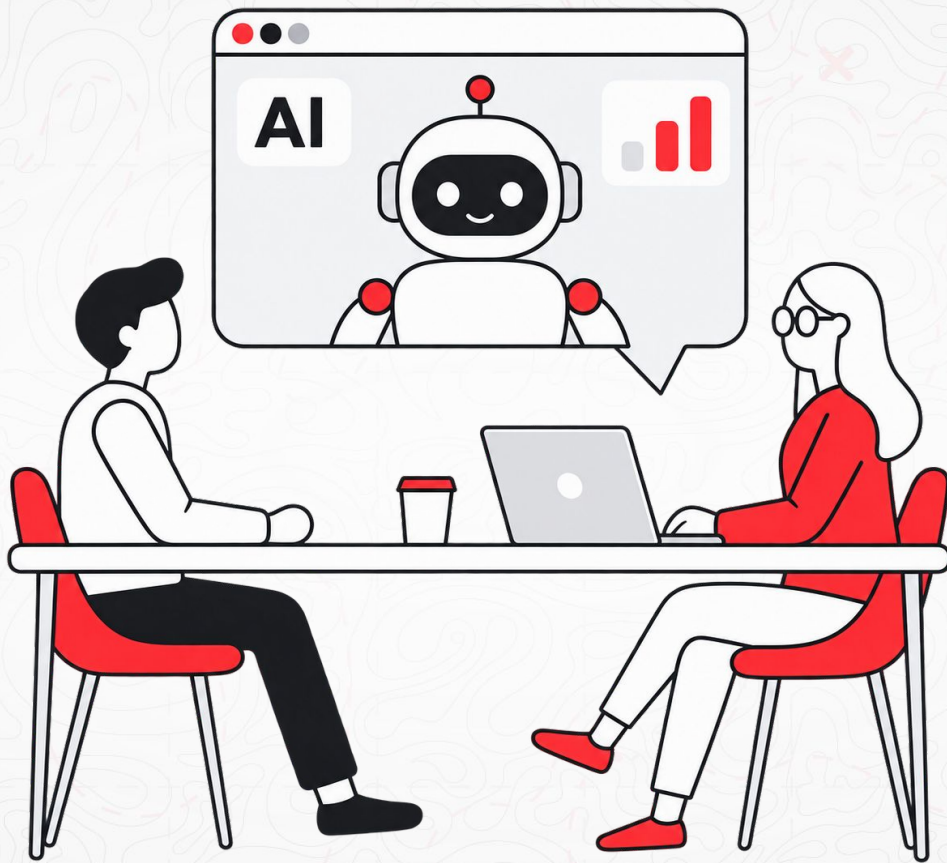
Scenario: Successful login

Given I am on the login page

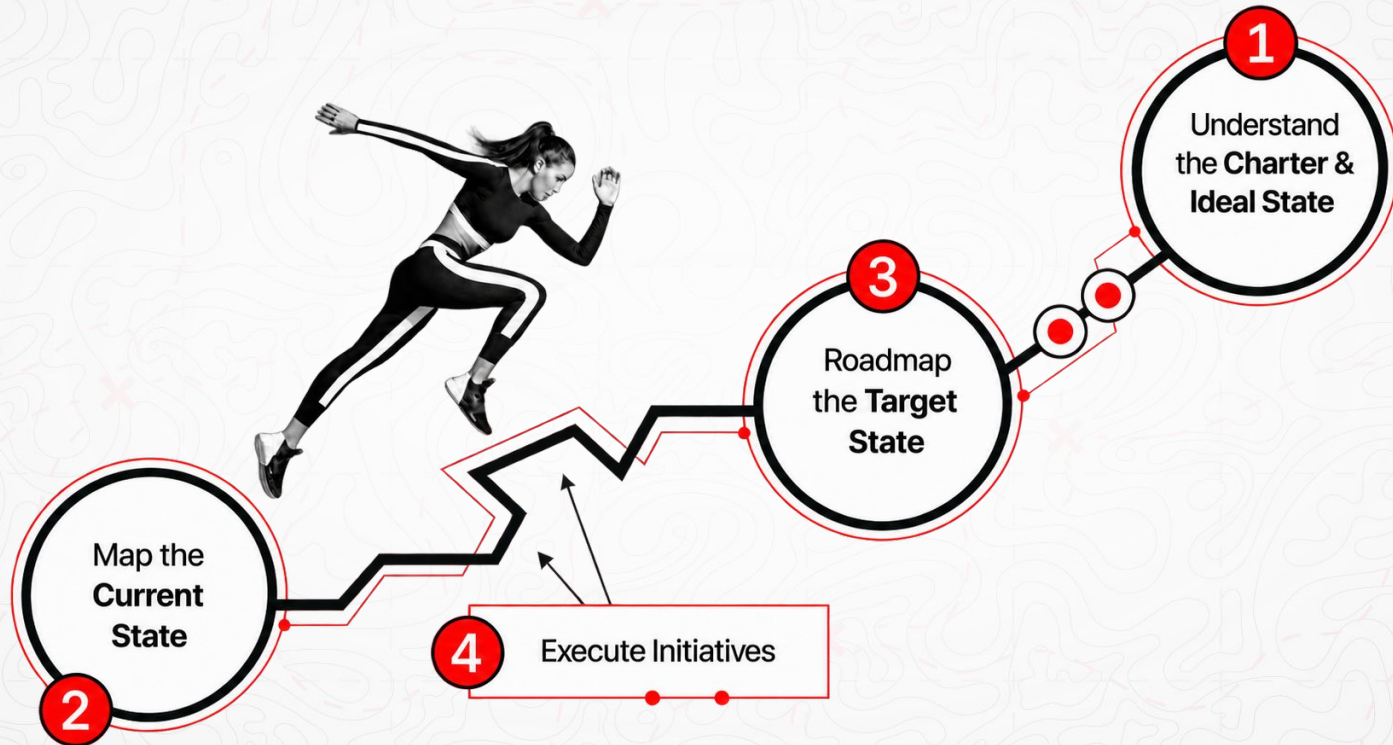
When I enter my username and password

And I click the login button

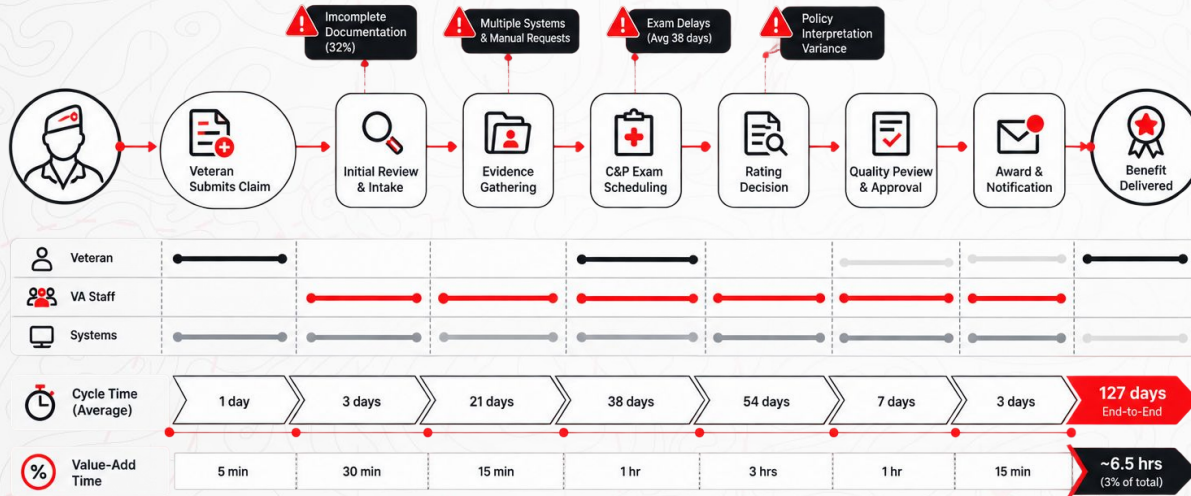
Then I should be taken to the dashboard page



Mission Improvement Framework



Mission Value Stream Mapping

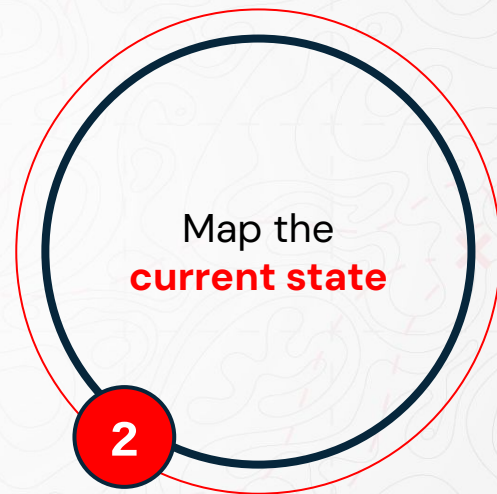


Map the
current state

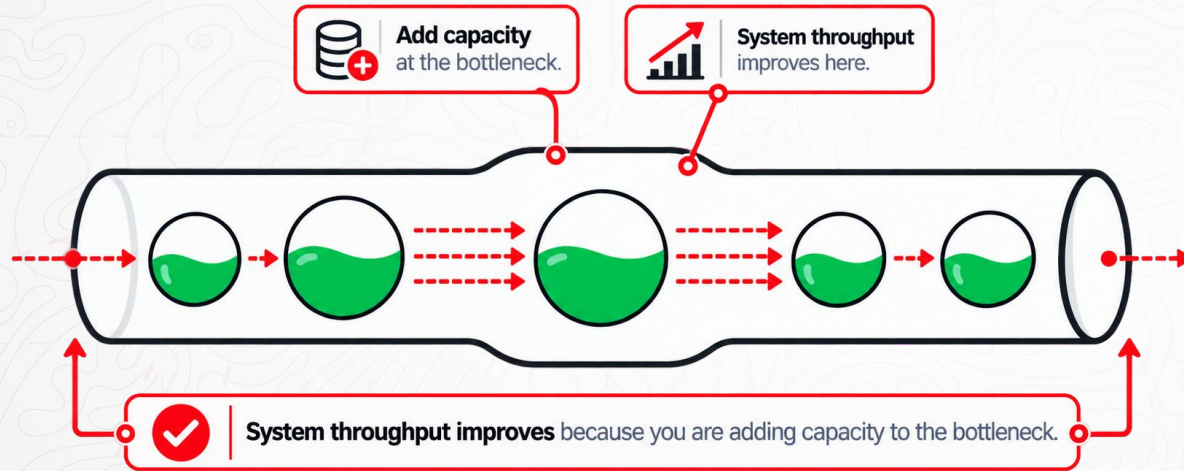
2

Map the Current State

- Mission Value Stream Mapping
- Mission Process Mapping
- Event Storming
- User Journey Mapping
- Current Constraint



Set the Target State



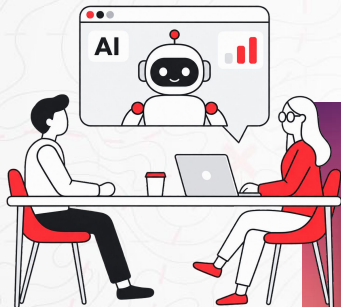
Roadmap the Current State

- Set Target State
- Impact Mapping
- Hypothesis Statement
- Outcome Oriented Roadmap
- Backlog (Themes, Epics, Stories)



Execute

- Backlog (Themes, Epics, Stories)



Feature: Login

As a user

I want to be able to log in

So that I can access my account

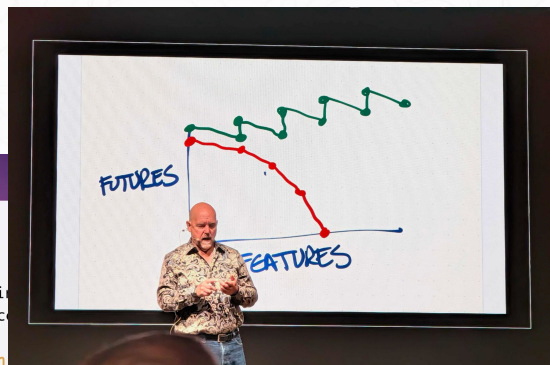
Scenario: Successful login

Given I am on the login page

When I enter my username and password

And I click the login button

Then I should be taken to the dashboard page



Execute
Initiatives

4

The mission outcome factory.

— INPUT

Mission
Need



1. OUTCOME-BASED DELIVERY SYSTEM

Measured in mission impact. Not sprints. Not story points. Not status decks.



2. ELITE TEAMS

Engineers, PMs, and designers who know modern software and get your mission context.



3. PRE-BUILT INFRASTRUCTURE

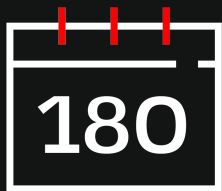
Platform, pipelines, and path to production already built. Not starting from scratch.

— OUTPUT

Mission
Outcome

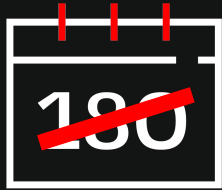


The mission outcome factory.



GUARANTEED

The mission outcome factory.



GUARANTEED

120... 90... 60... 30... real time?