



# Agile Product Roadmaps for Software Architecture

James Siddle, Skyhook Consulting Ltd  
O'Reilly Software Architecture Conference, New York  
February 2018

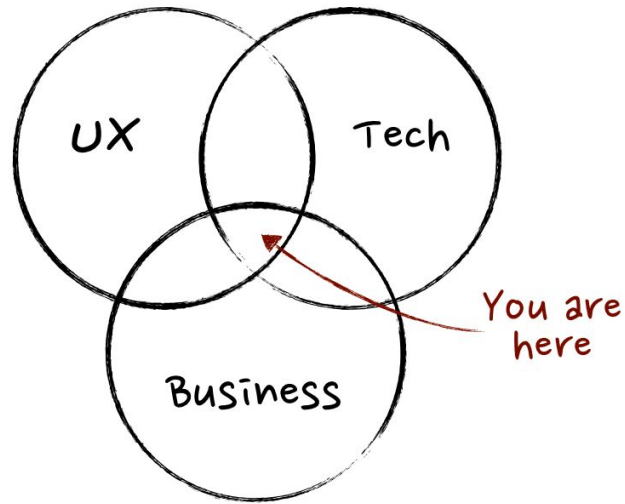
[@jamessiddle](#) / [jim@jamessiddle.net](mailto:jim@jamessiddle.net)

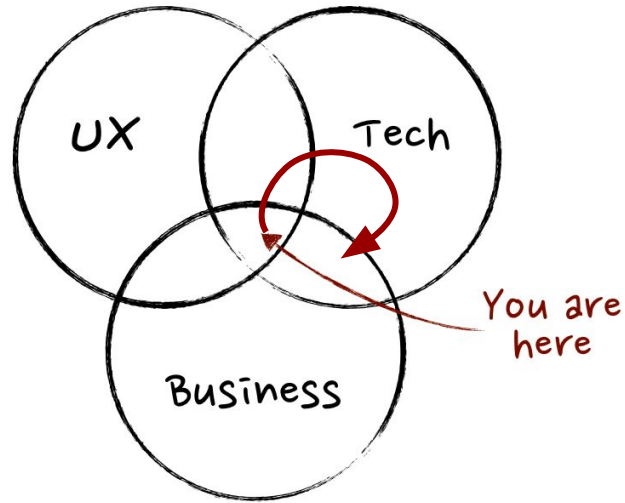


---

# Before we begin...

What's this all about, and who are you?







# Topics for this session

Product Management Intro

Software Architecture in a nutshell

**Strategic Product Tools - for architects**

**Roadmapping worked example**

Delivery Tools / Stakeholder engagement

---

# Architecture as a Product

Agile Roadmaps for Software Architecture? I still don't know what you're planning to talk about.

Please start the presentation already.

## A Product Manager

Is the **voice** of the customer  
... who cares about **creating** products  
... and wants to **delight** users



# (Agile) Product Management Tools

## Strategic

## Tactical

---

Product Vision

Product Backlog

Goals

Sprint Timeline

Hypotheses and Metrics

Roadmaps





# (Agile) Product Management Tools

Strategic

Tactical

---

**Product Vision**

Product Backlog

Goals

Sprint Timeline

Hypotheses and Metrics

Roadmaps



# (Agile) Product Management Tools

## Strategic

## Tactical

---

Product Vision

Product Backlog

**Goals**

Sprint Timeline

Hypotheses and Metrics

Roadmaps



# (Agile) Product Management Tools

Strategic

Tactical

---

Product Vision

Product Backlog

Goals

Sprint Timeline

**Hypotheses and Metrics**

Roadmaps



# (Agile) Product Management Tools

Strategic

Tactical

---

Product Vision

Product Backlog

Goals

Sprint Timeline

Hypotheses and Metrics

**Roadmaps**



# (Agile) Product Management Tools

Strategic

Tactical

---

Product Vision

**Product Backlog**

Goals

Sprint Timeline

Hypotheses and Metrics

Roadmaps



# (Agile) Product Management Tools

## Strategic

## Tactical

---

Product Vision

Product Backlog

Goals

**Sprint Timeline**

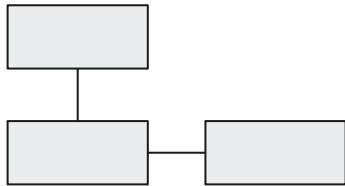
Hypotheses and Metrics

Roadmaps

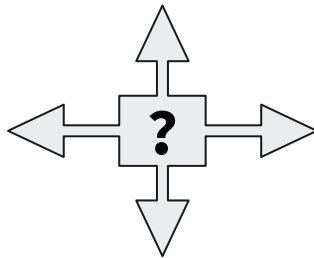
# Software Architecture

In a nutshell

High-level  
technical structure



Major, hard to  
change decisions



Qualities and  
trade-offs



Architectural  
styles, patterns





## Examples

A new app hosting platform

Reworking an existing system

Introduction of common patterns



---

# Strategic Product Tools

Tell me more about the product management tools, they sound really cool and useful.

I want to know what they have to do with Software Architecture.



# Product Tools applied to Architecture

## Product Vision

Goals  
Testable Hypo's and Metrics  
Roadmaps  
Product Backlog  
Sprint Timeline

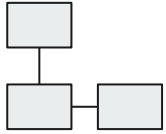


Key concepts, qualities  
Skeletal stepping stones  
Proven in reality  
Show evolution  
Delivery and dependencies  
Balance features with tech





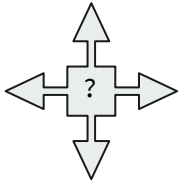
# Product Vision



Key concepts  
Core structure



Motivating qualities  
Architecture purpose



Major decisions  
...in principle



Proposed patterns

## WALL OF TEXT

*"We will build a platform to ingest and expose previously siloed, high-value data from across the business, for all new application development. This will transform the company's ability to adapt to changes in the marketplace, and will give us the opportunity to consolidate duplicated data, improve efficiency, and reduce costs. We'll implement a data transformation broker, highly scalable document-centric data store, and cloud-based microservices to achieve this. "*

## Key concepts / Mental model

*"We will build a platform to ingest and expose previously siloed, high-value data from across the business, for all new application development. This will transform the company's ability to adapt to changes in the marketplace, and will give us the opportunity to consolidate duplicated data, improve efficiency, and reduce costs. We'll implement a data transformation broker, highly scalable document-centric data store, and cloud based microservices to achieve this. "*

## Motivating qualities

*"We will build a platform to ingest and expose previously siloed, high-value data from across the business, for all new application development. This will transform the company's ability to **adapt to changes in the marketplace**, and will give us the opportunity to consolidate duplicated data, **improve efficiency**, and **reduce costs**. We'll implement a data transformation broker, **highly scalable** document-centric data store, and cloud based microservices to achieve this."*

## Major decisions, tech choices

*"We will **build a platform** to ingest and expose previously siloed, high-value data from across the business, for all new application development. This will transform the company's ability to adapt to changes in the marketplace, and will give us the opportunity to consolidate duplicated data, improve efficiency, and reduce costs. We'll implement a **data transformation broker**, highly scalable **document-centric data store**, and **cloud based microservices** to achieve this."*



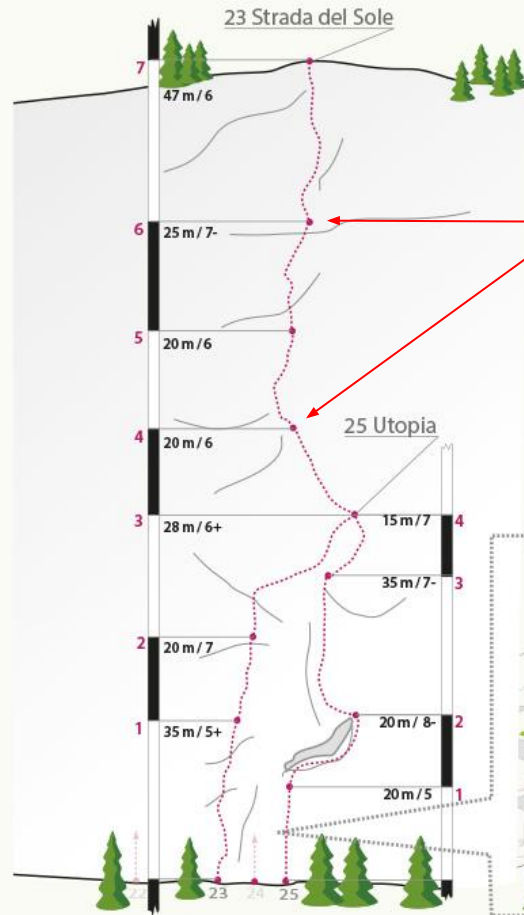
## Proposed Patterns

*"We will build a platform to ingest and expose previously siloed, high-value data from across the business, for all new application development. This will transform the company's ability to adapt to changes in the marketplace, and will give us the opportunity to consolidate duplicated data, improve efficiency, and reduce costs. We'll implement a data transformation **broker**, highly scalable document-centric data store, and cloud based **microservices** to achieve this."*



# Product Tools applied to Architecture



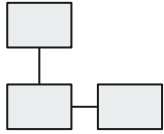


Belay points





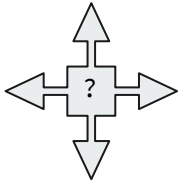
# Goals



Structural elaboration  
Architecture evolution



NFR delivery  
Quality realisation  
Expected tradeoffs



Technical decisions  
De-risking decisions



Pattern sequence

# Platform Goals

WALL OF TEXT

1. Stabilize and automate the current production systems
2. Elaborate Data Integration Architecture
  - a. Make changes as needed, but incrementally to address key risks first
  - b. Develop in a staging environment
  - c. Primary goal is to prove that key data integration use cases are supported
3. Promote new architecture into production
  - a. Requires parity with existing data available to users
4. Integrate key *skincare* datasets to enable *Athlete's Foot* and *Psoriasis* research
  - a. Ensure minimum data is available to support initial research programmes
5. Ensure user tools meet minimum user needs for research programmes
  - a. Add capabilities to expose underlying data as needed (TBD)
6. Enhance user tools based on rapid user feedback cycle

# Platform Goals

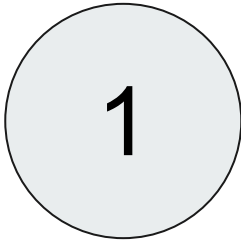
1. Stabilize and automate the current production systems
2. Elaborate Data Integration Architecture
  - a. Make changes as needed, but incrementally to address key risks first
  - b. Develop in a staging environment
  - c. Primary goal is to prove that key data integration use cases are supported
3. Promote new architecture into production
  - a. Requires parity with existing data available to users
4. Integrate key *skincare* datasets to enable *Athlete's Foot* and *Psoriasis* research
  - a. Ensure minimum data is available to support initial research programmes
5. Ensure user tools meet minimum user needs for research programmes
  - a. Add capabilities to expose underlying data as needed (TBD)
6. Enhance user tools based on rapid user feedback cycle

1

2

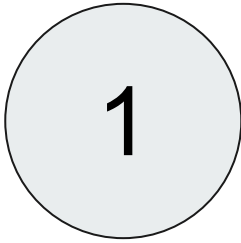
3

# Goals Example - key steps



Stabilize  
production  
infrastructure

# Goals Example - key steps

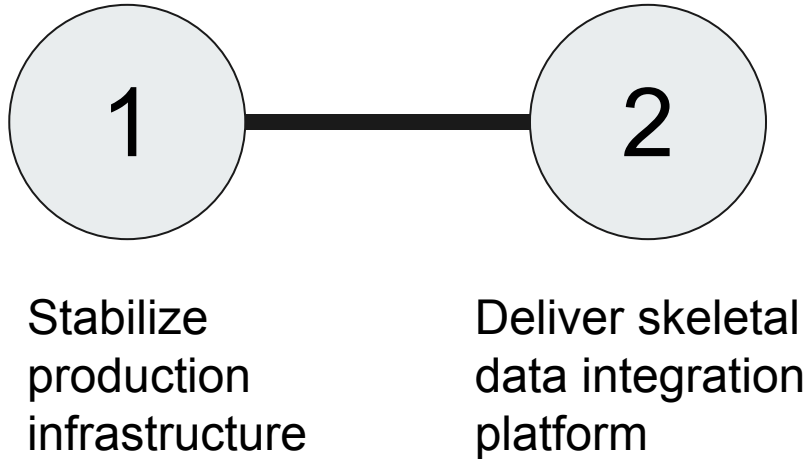


Stabilize  
production  
infrastructure

**Major risk**



# Goals Example - key steps



**Major risk**

# Goals Example - key steps



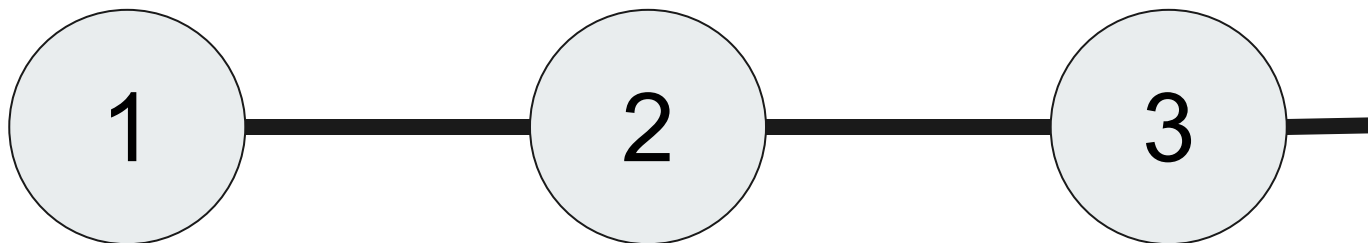
Stabilize  
production  
infrastructure

**Major risk**

Deliver skeletal  
data integration  
platform

**Platform viability**

# Goals Example - key steps



Stabilize  
production  
infrastructure

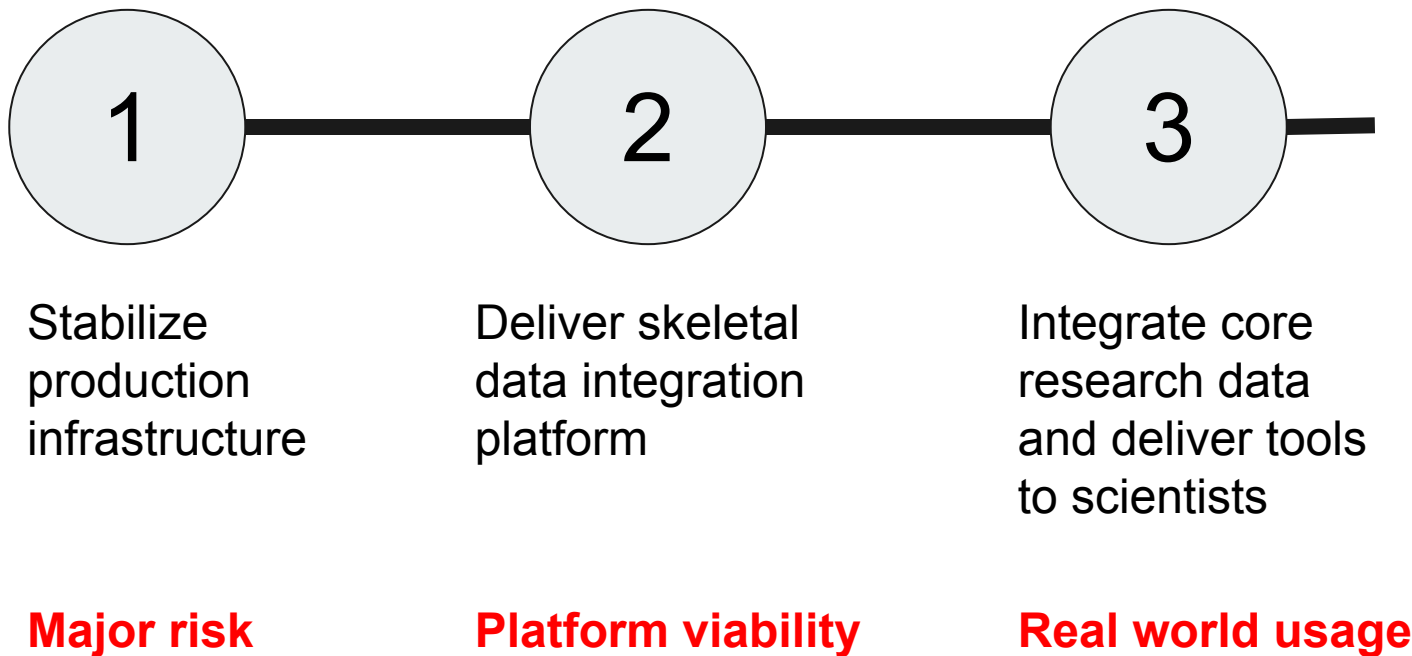
**Major risk**

Deliver skeletal  
data integration  
platform

**Platform viability**

Integrate core  
research data  
and deliver tools  
to scientists

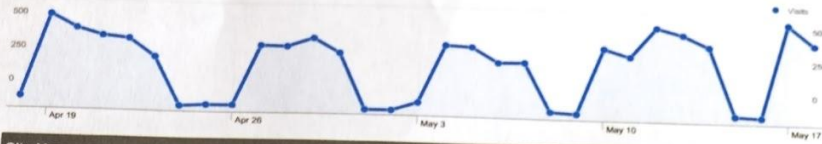
# Goals Example - key steps





# Product Tools applied to Architecture





#### Site Usage

7,649 Visits

43.64% Bounce Rate

25,423 Pageviews

00:04:08 Avg. Time on Site

3.32 Pages/Visit

28.30% % New Visits

#### Traffic Sources Overview



**Direct Traffic**  
 3,097.00 (40.49%)  
**Search Engines**  
 2,910.00 (38.94%)  
**Referring Sites**  
 1,642.00 (21.47%)

#### Map Overlay



#### Visitors Overview



Visitors  
2,958

#### Content Overview

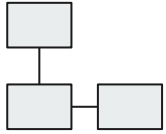
Pages	Pageviews	% Pageviews
/	5,932	23.33%
/information-resources	1,306	5.14%
/decisions	867	3.41%
/information-privacy	697	2.74%
/information-privacy-guidelines	692	2.72%



Visits	
Bounce Rate	
43.64%	
Site Avg: 43.64% (0.00%)	
% New Visits	Bounce Rate
27.27%	43.55%
85.19%	74.07%
56.52%	39.13%
95.45%	40.91%
92.31%	38.46%
85.71%	28.57%
100.00%	
40.00%	



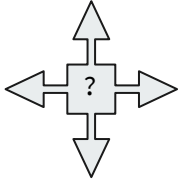
# Testable Hypotheses and Metrics



Test your architectural assumptions



Evidence of sound decisions, expected consequences

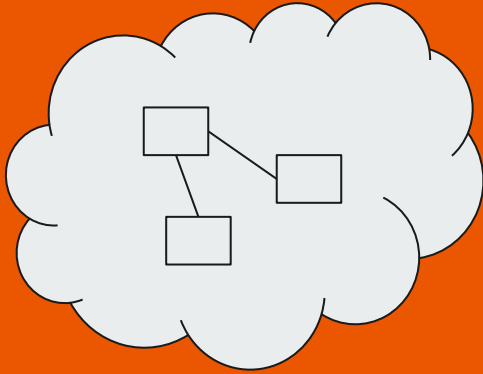


Reduce technical debt

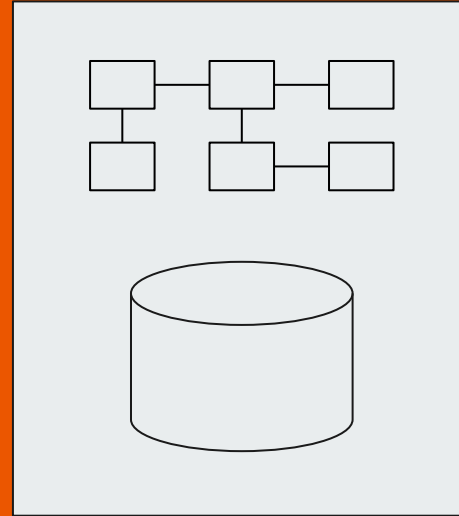


Patterns in reality  
Confirm context

# Passport Application Processing



**Replacement**



**Legacy Application**

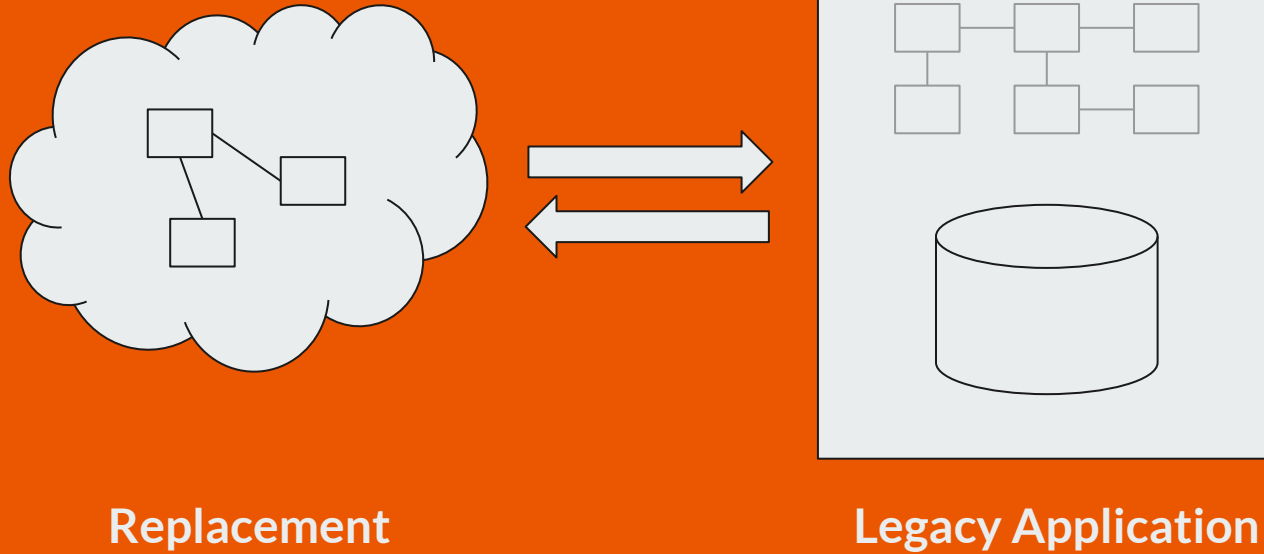


## Hypothesis

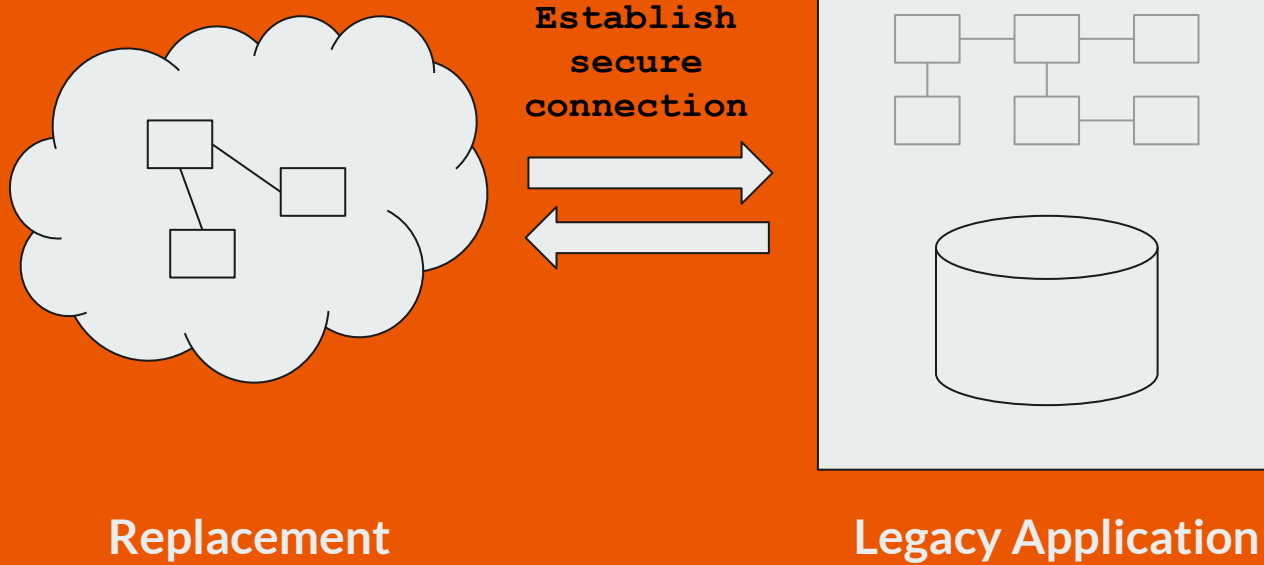
**We assert that the new passport processing system:**

1. Can establish a secure link to legacy data used for validating user details in passport applications
2. Will get the same results (in real-time) as the legacy application, indicating if the application is valid
3. Can utilize legacy data stores without adversely impacting existing operational workflows
4. Is viable as replacement for user detail validation currently performed by the legacy system

# Hypothesis

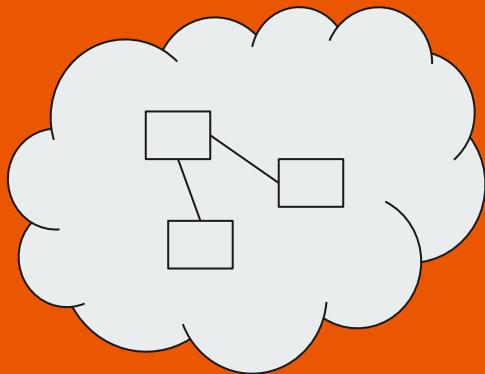
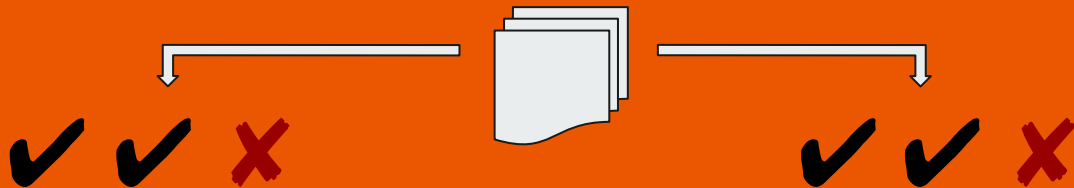


# Hypothesis



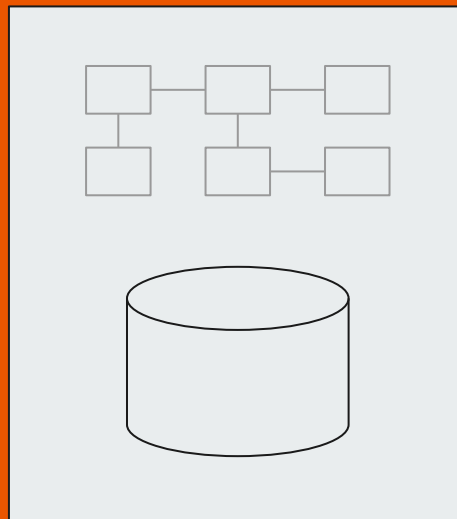
# Hypothesis

Same results  
as legacy  
application



**Replacement**

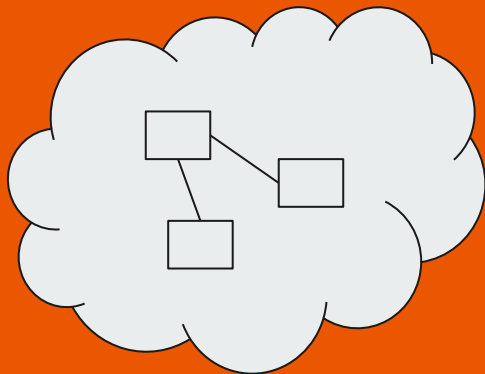
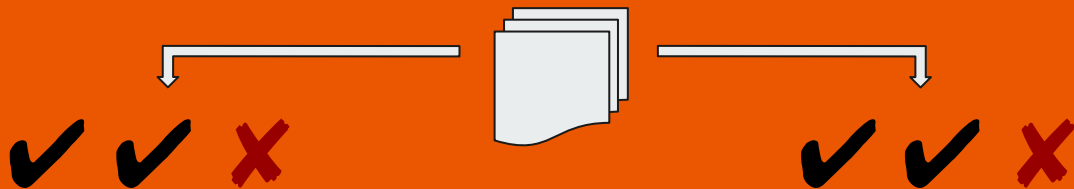
Establish  
secure  
connection



**Legacy Application**

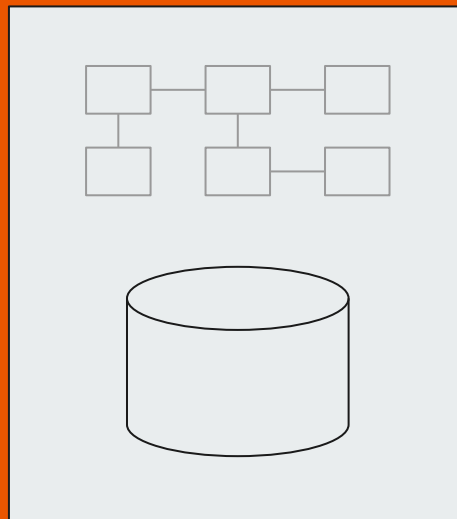
# Hypothesis

Same results  
as legacy  
application



Replacement

Establish  
secure  
connection

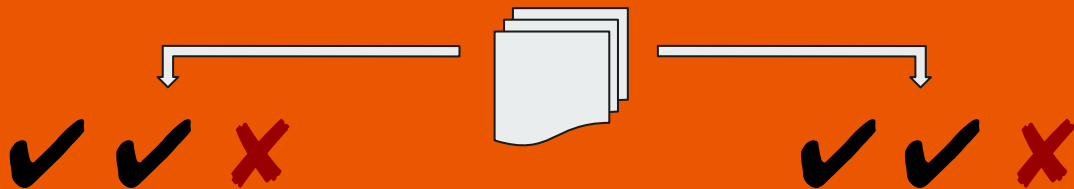


No adverse  
operational  
impact

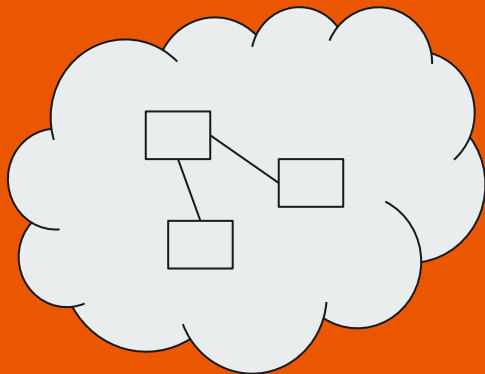
Legacy Application

# Hypothesis

Same results  
as legacy  
application

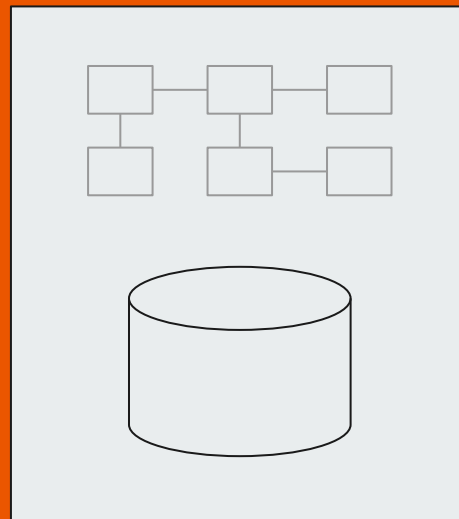


Viable  
replacement  
for legacy



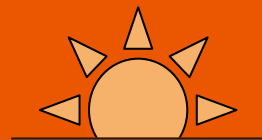
Replacement

Establish  
secure  
connection



Legacy Application

No adverse  
operational  
impact





## Passport Processing - initial metrics

# of applications validated Daily, eg. 472 today

# operational issues in legacy 1

# appls. rejected vs accepted 470 ✓ 2 ✗



# Product Tools applied to Architecture

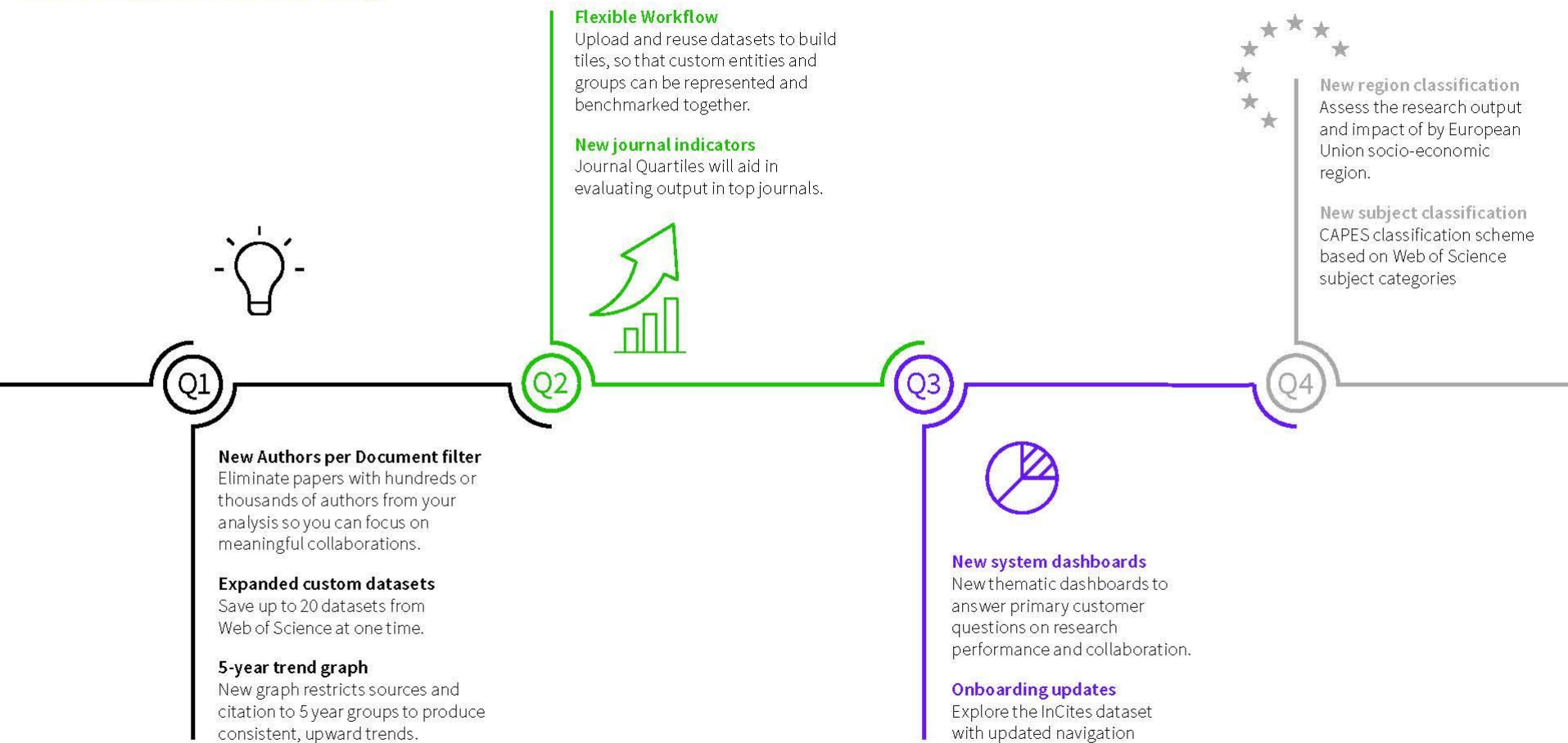
Product Vision  
Goals  
Testable Hypo's and Metrics  
**Roadmaps**  
Product Backlog  
Sprint Timeline



Key concepts, qualities  
Skeletal stepping stones  
Proven in reality  
Show evolution  
Delivery and dependencies  
Balance features with tech

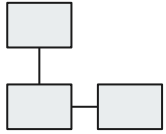


# InCites 2017 Product Roadmap





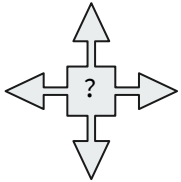
# Roadmaps



Illustrate elaboration  
Component deliveries  
Key dependencies



Show qualities and  
tradeoffs over time






Flag key decisions

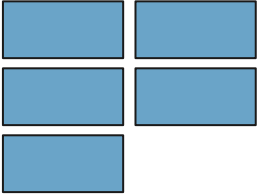

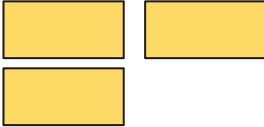


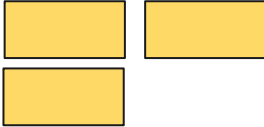

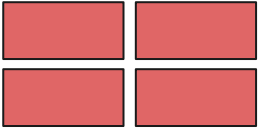
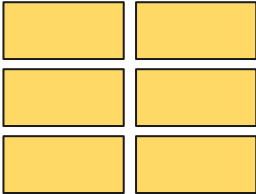



Show sequencing

# THE GO PRODUCT ROADMAP

 <b>DATE</b> The release date or timeframe	Date or timeframe	Date or timeframe	Date or timeframe	Date or timeframe
	When will the release be available?			
 <b>NAME</b> The name of the new release	Name/version	Name/version	Name/version	Name/version
	What is it called?			
 <b>GOAL</b> The reason for creating the new release	Goal	Goal	Goal	Goal
	Why is it developed? Which benefit does it offer?			
 <b>FEATURES</b> The high-level features necessary to meet the goal	Features	Features	Features	Features
	What are the 3-5 key features?			
 <b>METRICS</b> The metrics to determine if the goal has been met	Metrics	Metrics	Metrics	Metrics
	How do we know that the goal is met?			



APP WORKFLOW →	Step 1	Step 2	Step 3	Step 4
GOAL 1				
GOAL 2				
GOAL 3				

[Feature Map](#) Roadmap format



HMPO roadmap example  
(poor quality image intentional)

---

# Roadmapping: Worked Example

Can you please bring those things together so that  
I can make sense of them?



# Vision: HR Company

## Key concepts

Recruitment intelligence  
Data harvesting, integration  
Tooling innovation, recommendations

## Decisions in principle

Use Spark / HDFS for data processing  
Neo4J - host knowledge graph  
Amazon Web Services / EC2

## Motivating qualities

ML-ready data quality / integrity  
Ability to innovate quickly  
Security and regulation compliance

## Proposed Patterns

Microservices  
Pipes and Filters

**Date / Name**

**Goal**

**Hypothesis**

*We believe we can...*

**Success  
metrics**

**Features**

**NFR impact**



**Date / Name**

**1 month**

**3 months**

**6 months**

**Goal**

**Hypothesis**

*We believe we can...*

**Success  
metrics**

**Features**

**NFR impact**

(1) Create a timeline

*Time*



**Date / Name**

**1 month: Data Foundations**

**3 months: Core Platform**

**6 months: Scaled Platform**

**Goal**

**Hypothesis**

*We believe we can...*

(2) Convey intent and theme

**Success  
metrics**

**Features**

**NFR impact**

Date / Name	1 month: Data Foundations	3 months: Core Platform	6 months: Scaled Platform
Goal	Implement basic data retrieval, integration, distribution patterns	Build & deploy an alpha app, and quickly iterate on recommenders	Easily add new/ interesting datasets, beta application
Hypothesis <i>We believe we can...</i>	(3) Summarise your stepping stones		
Success metrics			
Features			
NFR impact			

Date / Name	1 month: Data Foundations	3 months: Core Platform	6 months: Scaled Platform
Goal	Implement basic data retrieval, integration, distribution patterns	Build & deploy an alpha app, and quickly iterate on recommenders	Integrate further interesting datasets, beta application
<b>Hypothesis</b> <i>We believe we can...</i>	<p>(4) Select a hypothesis to test your assumptions</p>		
Success metrics			
Features			
NFR impact			

Date / Name	1 month: Data Foundations	3 months: Core Platform	6 months: Scaled Platform
Goal	Implement basic data retrieval, integration, distribution patterns	Build & deploy an alpha app, and quickly iterate on recommenders	Integrate further interesting datasets, beta application
<b>Hypothesis</b> <i>We believe we can...</i>	<ul style="list-style-type: none"> <li>- harvest core datasets</li> <li>- disambiguate and integrate data</li> <li>- make employers/ees searchable</li> <li>- build demo HR intelligence tool</li> <li>- experiment with recommenders</li> </ul>		
<b>Success metrics</b>	0.8 f-score vs test set Zero security incidents Zero high sev PEN test results		
Features			
NFR impact			

(5) Quantify how you'll prove your hypothesis

Date / Name	1 month: Data Foundations	3 months: Core Platform	6 months: Scaled Platform
Goal	Implement basic data retrieval, integration, distribution patterns	Build & deploy an alpha app, and quickly iterate on recommenders	Integrate further interesting datasets, beta application
<b>Hypothesis</b> <i>We believe we can...</i>	<ul style="list-style-type: none"> <li>- harvest core datasets</li> <li>- disambiguate and integrate data</li> <li>- make employers/ees searchable</li> <li>- build demo HR intelligence tool</li> <li>- experiment with recommenders</li> </ul>	<ul style="list-style-type: none"> <li>- implement prod. infrastructure</li> <li>- roll an application out into alpha</li> <li>- gather feedback</li> <li>- integrate a recommender</li> <li>- iterate on recommenders</li> </ul>	Iterate...
Success metrics	0.8 f-score vs test set Zero security incidents Zero high sev PEN test results	New app/model deploy < 12 hours Issue response < 48 hours App usage metrics available	
Features			
NFR impact			

Date / Name	1 month: Data Foundations	3 months: Core Platform	6 months: Scaled Platform
Goal	Implement basic data retrieval, integration, distribution patterns	Build & deploy an alpha app, and quickly iterate on recommenders	Integrate further interesting datasets, beta application
<b>Hypothesis</b> <i>We believe we can...</i>	<ul style="list-style-type: none"> <li>- harvest core datasets</li> <li>- disambiguate and integrate data</li> <li>- make employers/ees searchable</li> <li>- build demo HR intelligence tool</li> <li>- experiment with recommenders</li> </ul>	<ul style="list-style-type: none"> <li>- implement prod. infrastructure</li> <li>- roll an application out into alpha</li> <li>- gather feedback</li> <li>- integrate a recommender</li> <li>- iterate on recommenders</li> </ul>	<ul style="list-style-type: none"> <li>- add three more datasets</li> <li>- roll an application into beta</li> <li>- demo additional tools</li> </ul>
Success metrics	0.8 f-score vs test set Zero security incidents Zero high sev PEN test results	New app/model deploy < 12 hours Issue response < 48 hours App usage metrics available	Dataset integration < 1wk 0.9 f-score vs test set
Features			
NFR impact			

Date / Name	1 month: Data Foundations	3 months: Core Platform	6 months: Scaled Platform
Goal	Implement basic data retrieval, integration, distribution patterns	Build & deploy an alpha app, and quickly iterate on recommenders	Integrate further interesting datasets, beta application
Hypothesis <i>We believe we can...</i>	<ul style="list-style-type: none"> <li>- harvest core datasets</li> <li>- disambiguate and integrate data</li> <li>- make employers/ees searchable</li> <li>- build demo HR intelligence tool</li> <li>- experiment with recommenders</li> </ul>	<ul style="list-style-type: none"> <li>- implement prod. infrastructure</li> <li>- roll an application out into alpha</li> <li>- gather feedback</li> <li>- integrate a recommender</li> <li>- iterate on recommenders</li> </ul>	<ul style="list-style-type: none"> <li>- add three more datasets</li> <li>- roll an application into beta</li> <li>- demo additional tools</li> </ul>
Success metrics	0.8 f-score vs test set Zero security incidents Zero high sev PEN test results	New app/model deploy < 12 hours Issue response < 48 hours App usage metrics available	Dataset integration < 1wk 0.9 f-score vs test set
Features	Data harvesters / Spark / HDFS Entity disambiguator Core Knowledge Graph in Neo4J Recommender PoC	Core Production infrastructure Microservice platform Recommender framework Instrumentation Alpha support model	New harvesters / dictionaries HA Infrastructure Beta support model Tooling experiments
NFR impact	(6) Describe what will be built		



Date / Name	1 month: Data Foundations	3 months: Core Platform	6 months: Scaled Platform
Goal	Implement basic data retrieval, integration, distribution patterns	Build & deploy an alpha app, and quickly iterate on recommenders	Integrate further interesting datasets, beta application
Hypothesis <i>We believe we can...</i>	<ul style="list-style-type: none"> <li>- harvest core datasets</li> <li>- disambiguate and integrate data</li> <li>- make employers/ees searchable</li> <li>- build demo HR intelligence tool</li> <li>- experiment with recommenders</li> </ul>	<ul style="list-style-type: none"> <li>- implement prod. infrastructure</li> <li>- roll an application out into alpha</li> <li>- gather feedback</li> <li>- integrate a recommender</li> <li>- iterate on recommenders</li> </ul>	<ul style="list-style-type: none"> <li>- add three more datasets</li> <li>- roll an application into beta</li> <li>- demo additional tools</li> </ul>
Success metrics	0.8 f-score vs test set Zero security incidents Zero high sev PEN test results	New app/model deploy < 12 hours Issue response < 48 hours App usage metrics available	Dataset integration < 1wk 0.9 f-score vs test set
Features	Data harvesters / Spark / HDFS Entity disambiguator Core Knowledge Graph in Neo4J Recommender PoC	Production infrastructure Microservice platform Recommender framework Instrumentation Alpha support model	New harvesters / dictionaries HA Infrastructure Beta support model Tooling experiments
NFR impact	<b>Data Integrity and Security</b> <b>Scalability (data processing)</b> <b>Operability (data management)</b>	(7) Show impact on qualities (+/-)	

Date / Name	1 month: Data Foundations	3 months: Core Platform	6 months: Scaled Platform
Goal	Implement basic data retrieval, integration, distribution patterns	Build & deploy an alpha app, and quickly iterate on recommenders	Integrate further interesting datasets, beta application
Hypothesis <i>We believe we can...</i>	<ul style="list-style-type: none"> <li>- harvest core datasets</li> <li>- disambiguate and integrate data</li> <li>- make employers/ees searchable</li> <li>- build demo HR intelligence tool</li> <li>- experiment with recommenders</li> </ul>	<ul style="list-style-type: none"> <li>- implement prod. infrastructure</li> <li>- roll an application out into alpha</li> <li>- gather feedback</li> <li>- integrate a recommender</li> <li>- iterate on recommenders</li> </ul>	<ul style="list-style-type: none"> <li>- add three more datasets</li> <li>- roll an application into beta</li> <li>- demo additional tools</li> </ul>
Success metrics	0.8 f-score vs test set Zero security incidents Zero high sev PEN test results	New app/model deploy < 12 hours Issue response < 48 hours App usage metrics available	Dataset integration < 1wk 0.9 f-score vs test set
Features	Data harvesters / Spark / HDFS Entity disambiguator Core Knowledge Graph in Neo4J Recommender PoC	Production infrastructure Microservice platform Recommender framework Instrumentation Alpha support model	New harvesters / dictionaries HA Infrastructure Beta support model Tooling experiments
NFR impact	Data Integrity and Security Scalability (data processing) Operability (data management)	Adaptability (model) Fault tolerance, Dev Scalability Development complexity Performance (memory)	

Date / Name	1 month: Data Foundations	3 months: Core Platform	6 months: Scaled Platform
Goal	Implement basic data retrieval, integration, distribution patterns	Build & deploy an alpha app, and quickly iterate on recommenders	Integrate further interesting datasets, beta application
<b>Hypothesis</b> <i>We believe we can...</i>	<ul style="list-style-type: none"> <li>- harvest core datasets</li> <li>- disambiguate and integrate data</li> <li>- make employers/ees searchable</li> <li>- build demo HR intelligence tool</li> <li>- experiment with recommenders</li> </ul>	<ul style="list-style-type: none"> <li>- implement prod. infrastructure</li> <li>- roll an application out into alpha</li> <li>- gather feedback</li> <li>- integrate a recommender</li> <li>- iterate on recommenders</li> </ul>	<ul style="list-style-type: none"> <li>- add three more datasets</li> <li>- roll an application into beta</li> <li>- demo additional tools</li> </ul>
Success metrics	0.8 f-score vs test set Zero security incidents Zero high sev PEN test results	New app/model deploy < 12 hours Issue response < 48 hours App usage metrics available	Dataset integration < 1wk 0.9 f-score vs test set
Features	Data harvesters / Spark / HDFS Entity disambiguator Core Knowledge Graph in Neo4J Recommender PoC	Production infrastructure Microservice platform Recommender framework Instrumentation Alpha support model	New harvesters / dictionaries HA Infrastructure Beta support model Tooling experiments
NFR impact	Data Integrity and Security Scalability (data processing) Operability (data management)	Adaptability (model) Fault tolerance, Dev Scalability Development complexity Performance (memory)	Data Coverage Availability Operations complexity

# Timeline (now / next / someday)

Date / Name	1 month: Data Foundations	3 months: Core Platform	6 months: Scaled Platform
Goal	Implement basic data retrieval, integration, distribution patterns	Build & deploy an alpha app, and quickly iterate on recommenders	Integrate further interesting datasets, beta application
Hypothesis <i>We believe we can...</i>	<ul style="list-style-type: none"> <li>- harvest core datasets</li> <li>- disambiguate and integrate data</li> <li>- make employers/ees searchable</li> <li>- build demo HR intelligence tool</li> <li>- experiment with recommenders</li> </ul>	<ul style="list-style-type: none"> <li>- implement prod. infrastructure</li> <li>- roll an application out into alpha</li> <li>- gather feedback</li> <li>- integrate a recommender</li> <li>- iterate on recommenders</li> </ul>	<ul style="list-style-type: none"> <li>- add three more datasets</li> <li>- roll an application into beta</li> <li>- demo additional tools</li> </ul>
Success metrics	0.8 f-score vs test set Zero security incidents Zero high sev PEN test results	New app/model deploy < 12 hours Issue response < 48 hours App usage metrics available	New app/model deploy < 12 hours Issue response < 48 hours 0.9 f-score vs test set
Features	Data harvesters / Spark / HDFS Entity disambiguator Core Knowledge Graph in Neo4J Recommender PoC	Production infrastructure Microservice platform Recommender framework Instrumentation Alpha support model	New harvesters / dictionaries HA Infrastructure Beta support model Tooling experiments
NFR impact	Data Integrity and Security Scalability (data processing) Operability (data management)	Fault tolerance Scalability (development) Development complexity Performance (memory)	Quality Realisation Availability Operations complexity

Skeletal Stepping stones

Proven in reality

NFR-based metrics

Major tech decisions

Components

Patterns

Quality Realisation

Tradeoffs

---

# Tactical Product Tools

So I have a roadmap, but what happens next?

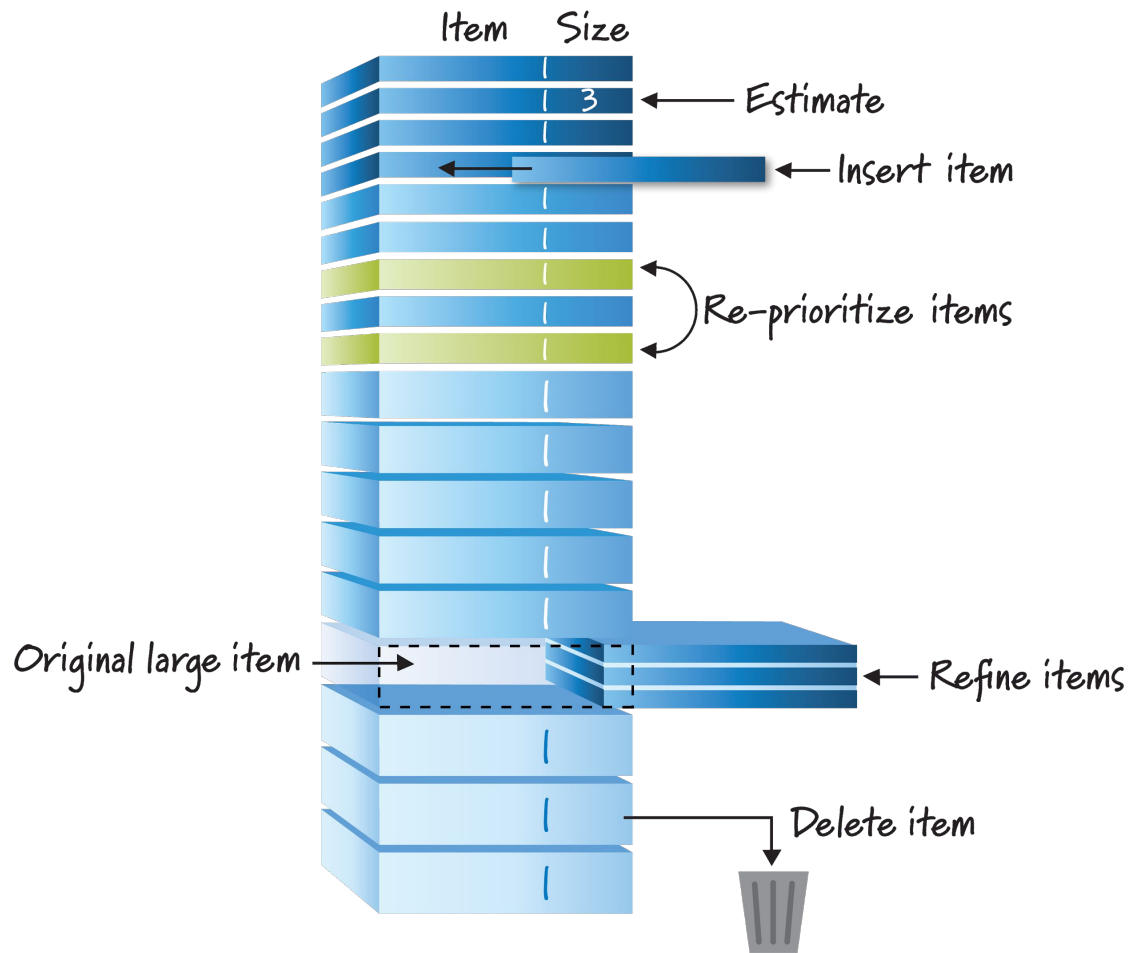


# Product Tools applied to Architecture

Product Vision  
Goals  
Testable Hypo's and Metrics  
Roadmaps  
**Product Backlog**  
Sprint Timeline

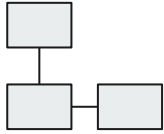


Key concepts, qualities  
Skeletal stepping stones  
Proven in reality  
Show evolution  
Delivery and dependencies  
Balance features with tech





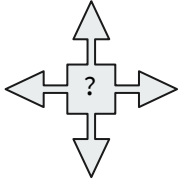
# Product Backlog



Technical tasks  
Arch. dependencies



Measurement tasks  
Acceptance criteria

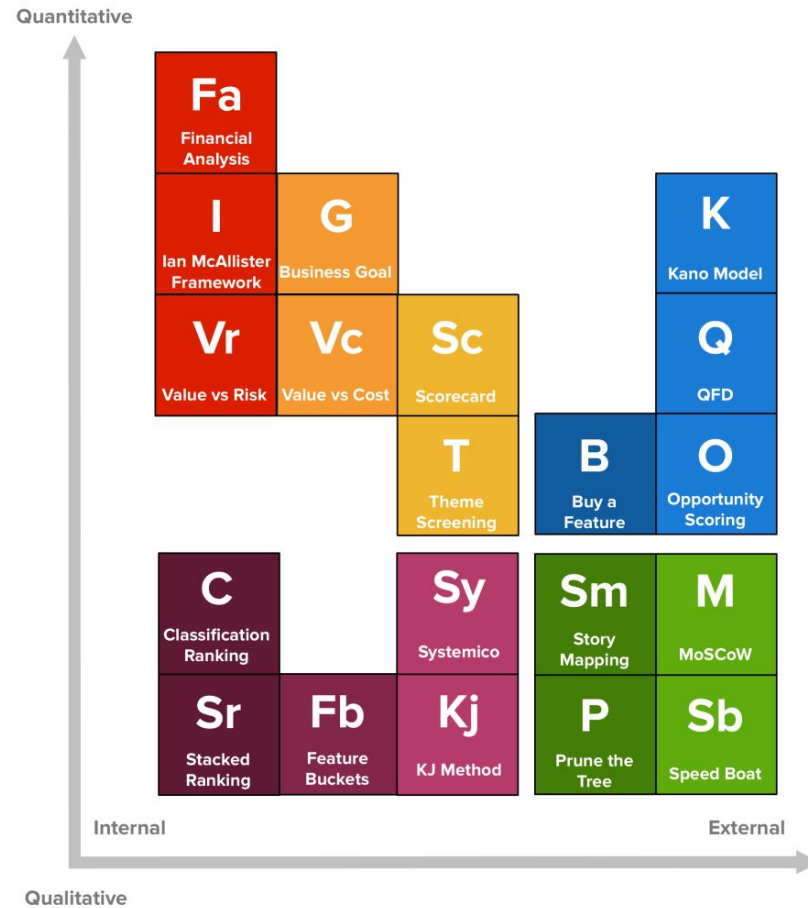


Spikes  
Technical workshops  
Actual decision making



Pattern experiments  
Implementations





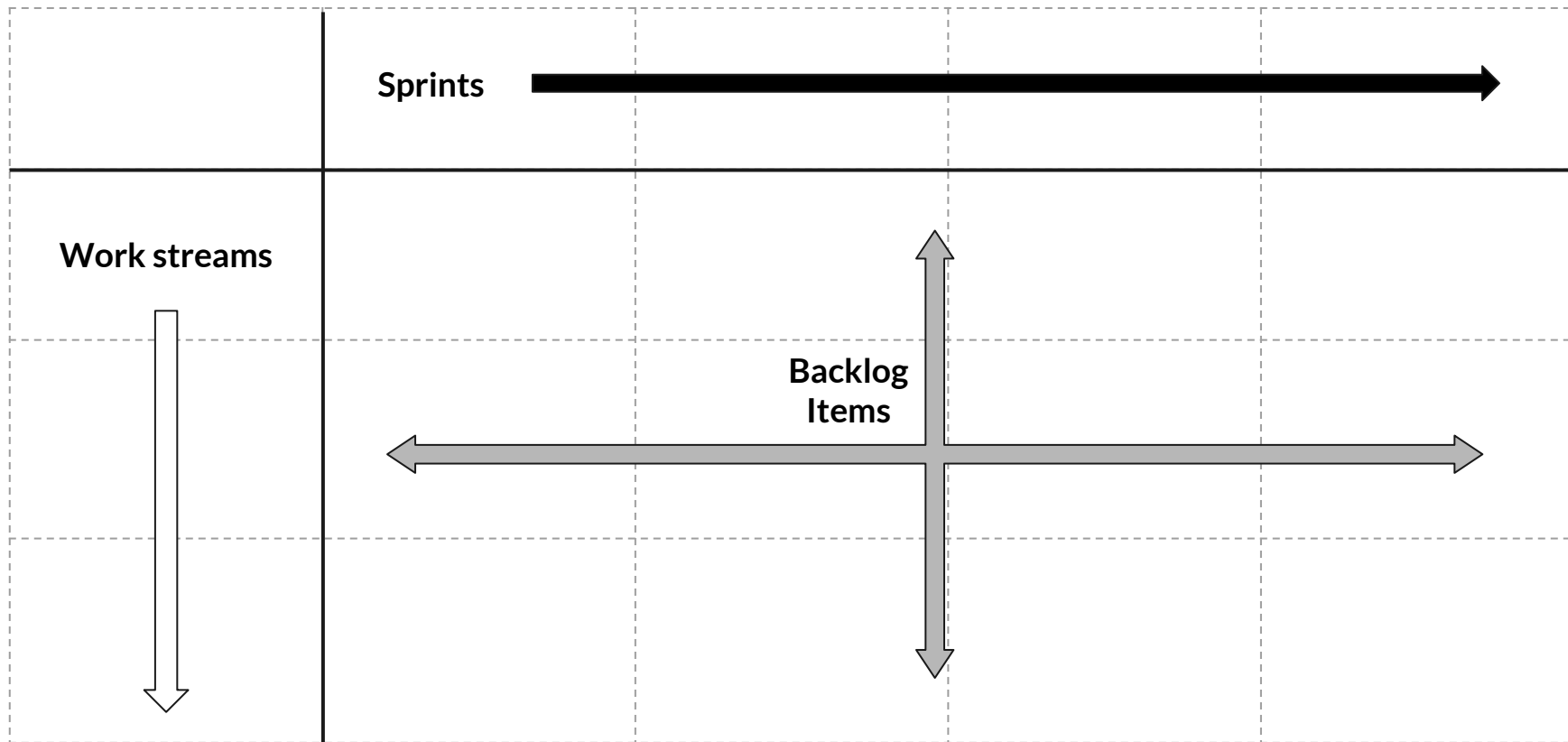
Credit: *Folding Burritos*  
 20 Product Prioritization  
 Techniques

<https://foldingburritos.com/product-prioritization-techniques/>



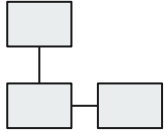
# Product Tools applied to Architecture







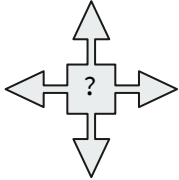
# Sprint Timeline



Feature/tech balance  
Scheduling  
Technical dependencies



Technical feedback  
Architecture reviews



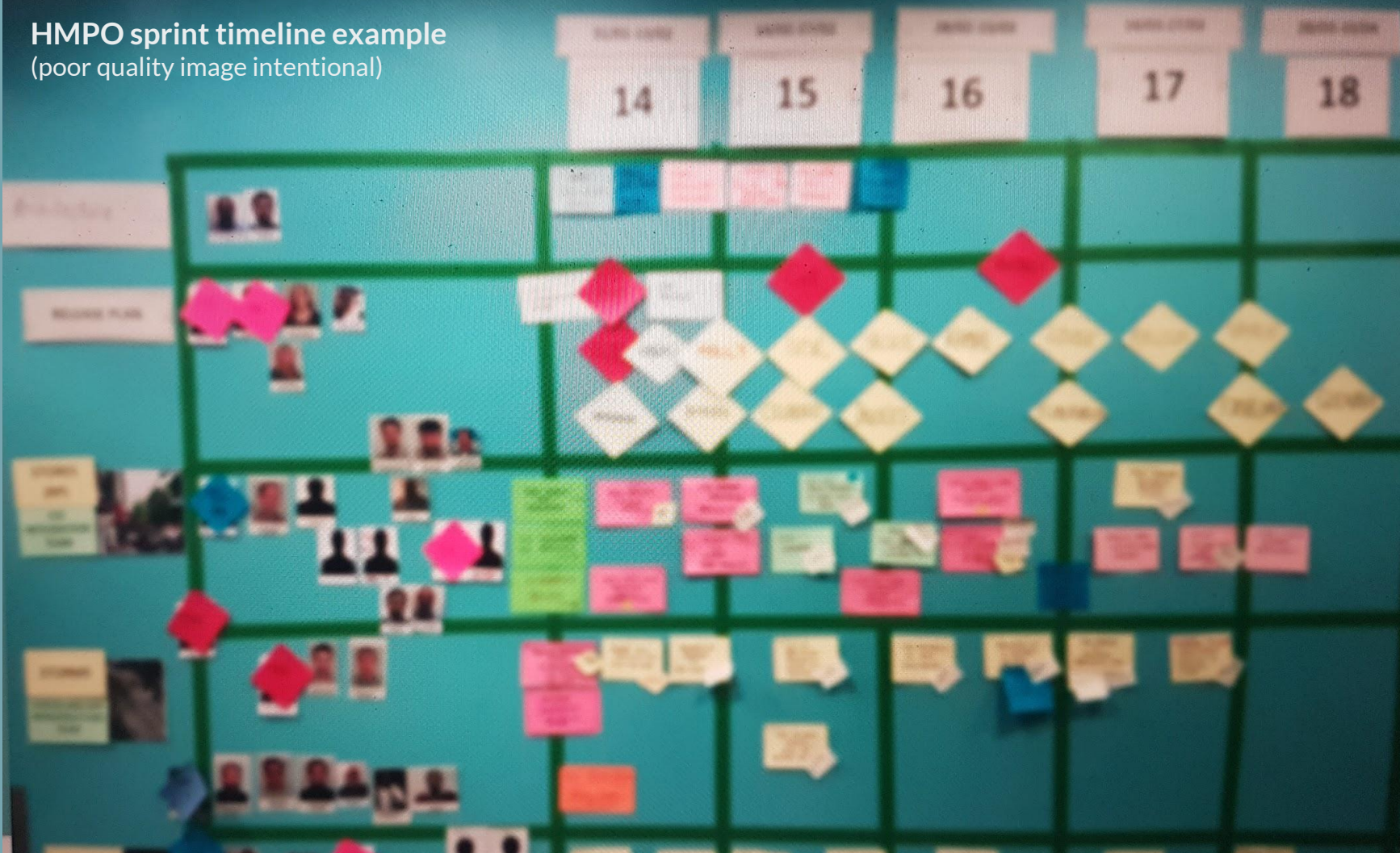
Imminent decisions  
Decision visibility



Plan rollout

# HMPO sprint timeline example

(poor quality image intentional)





---

# Stakeholder Management

That's all well and good, but it's stakeholder management that really makes me want to stick a fork in my eye.

Please make it less painful for me.









# Negotiation Points

Agree overarching purpose, benefits, and scope

Agree acceptable risks, overall logistics

Agree expected evidence vs assumptions

Agree product and technical specifics

Decide near term priorities

Decide near term development logistics



## Some Scenarios

Analysis paralysis

Conflicting priorities

Ambiguous authority

Contradictory directives

Competing initiatives

Trumped by direct business value

Steamrollers / Asshats

Ignorant stakeholders

---

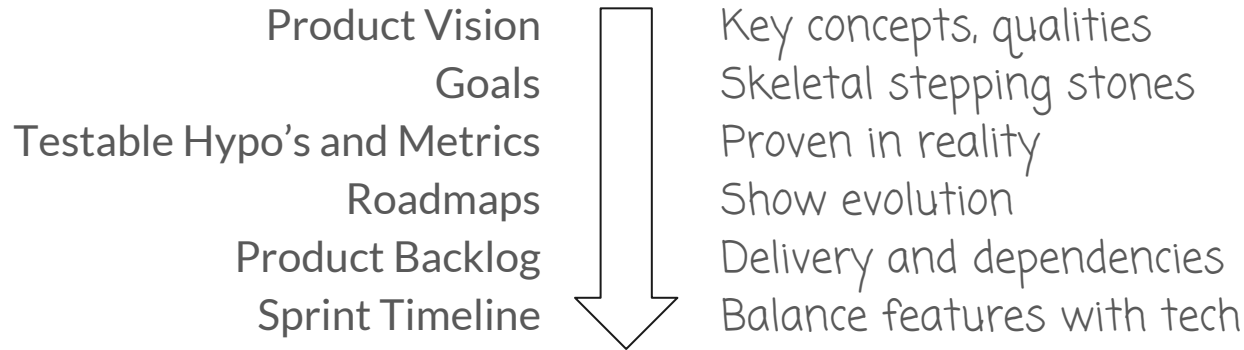
# Key Takeaways

That's nice, but I've already been to four sessions today and I'm on my fifth cup of coffee.

Can you wrap it up so I have some good sound bites to tell my manager?



## (Agile) Product Tools applied to Architecture



---

# Questions?

I wasn't really paying attention, would you mind repeating everything from the first slide onward?

# Thank you

Lyndsay Prewer / Equal Experts

Simon Bostock / Equal Experts

Alison Taylor / M&S

Andrew Neilson / M&S

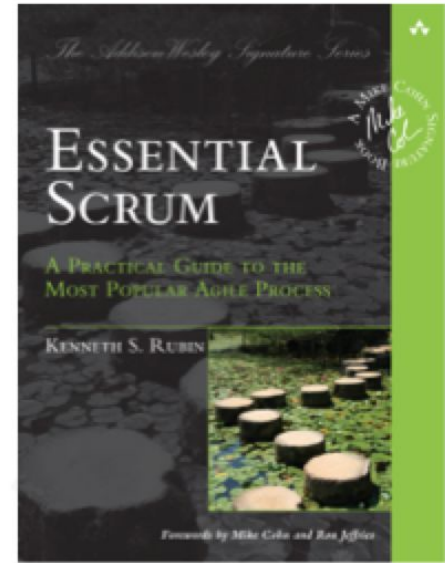
Roman Pichler / Pichler Consulting

---

Her Majesty's Passport Office

M&S

- ✱ Slides in this presentation contain items from the Visual AGILExicon®, which is a trademark of Innolution, LLC and Kenneth S. Rubin.
- ✱ The Visual AGILExicon is used and described in the book: ***Essential Scrum: A Practical Guide to the Most Popular Agile Process.***
- ✱ You can learn more about the Visual AGILExicon and permitted uses at: <http://innolution.com/resources/val-home-page>



Visual AGILExicon®