

SHIVOM CONSULTANCY LIMITED

CLOUD SUPPORT SERVICE DESCRIPTION

INTRODUCTION

- ▶ Shivom Consultancy Limited is a UK registered company specialising in
 - ▶ IT Strategy
 - ▶ Architecture and Design
 - ▶ Digital Transformation
 - ▶ Public Cloud Adoption / Migration
 - ▶ Delivering Software Engineering Services
 - ▶ Building cost effective secure solutions
 - ▶ Cloud Support and Training
- ▶ Operating in UK market since 2003
- ▶ Specialise in Agile Delivery and Public Cloud Migration/Adoption
- ▶ AWS and Microsoft Cloud Partner



SERVICE DESCRIPTION

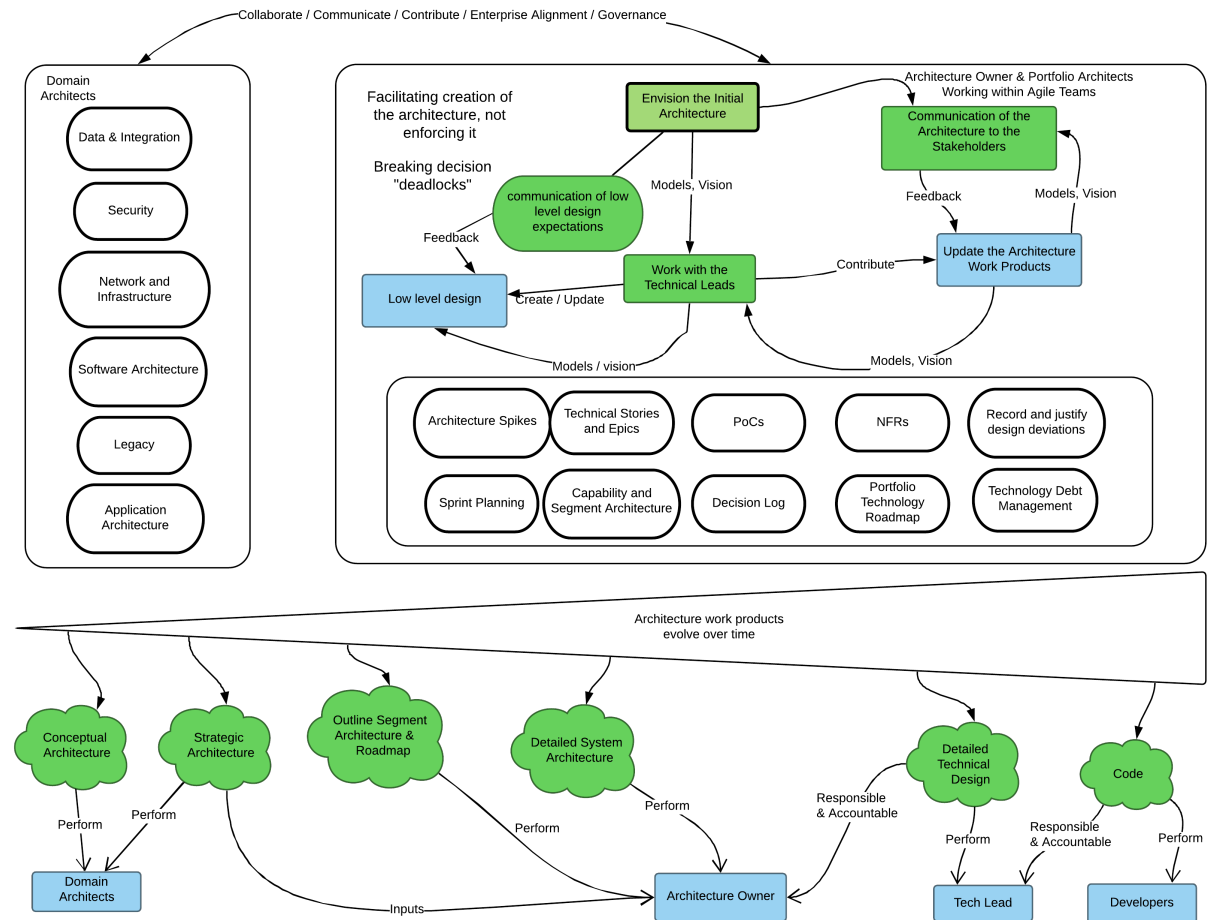
CLOUD SUPPORT PRACTICE

CLOUD SUPPORT SERVICE

- ▶ Strategic Architecture - growth strategy and planning
- ▶ Architecture envisioning - conducting strategic discoveries, user insight and co-creation
- ▶ Business case and investment proposals
- ▶ Alpha: Rapid prototyping to enable ideation and new opportunity development
- ▶ Alpha: Market scanning and landscaping
- ▶ Beta: Route to scale strategy and implementation
- ▶ Live: ongoing architecture support

DESIGN EVOLVES FROM STRATEGY TO IMPLEMENTATION

- ▶ Enterprise/Domain Architects
- ▶ Architect Owners
- ▶ Technical Leads (SMEs)
- ▶ Engineers/Developers

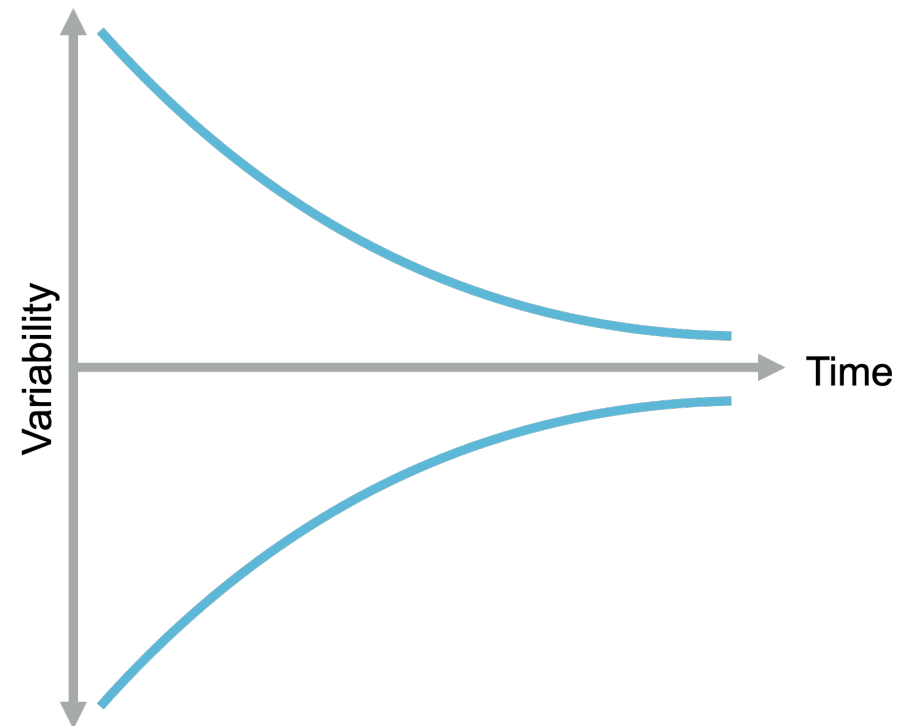


OUR ETHOS OF SOFTWARE ENGINEERING BLENDED
WITH AGILE METHODOLOGY

DELIVERY APPROACH

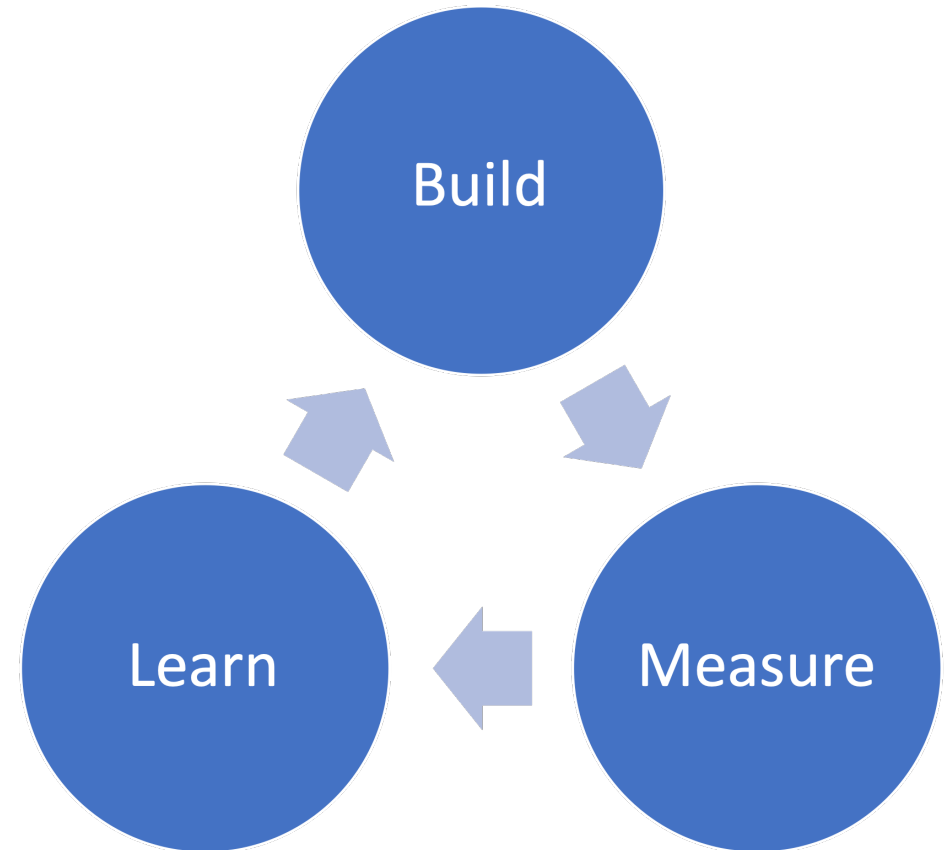
BEFORE WE START A PROJECT WE KNOW LEAST ABOUT MOST THINGS

- ▶ how people will use software before we put it in front of them
- ▶ the delivery rate of the team before they start working together
- ▶ the constraints of the organisation on the project



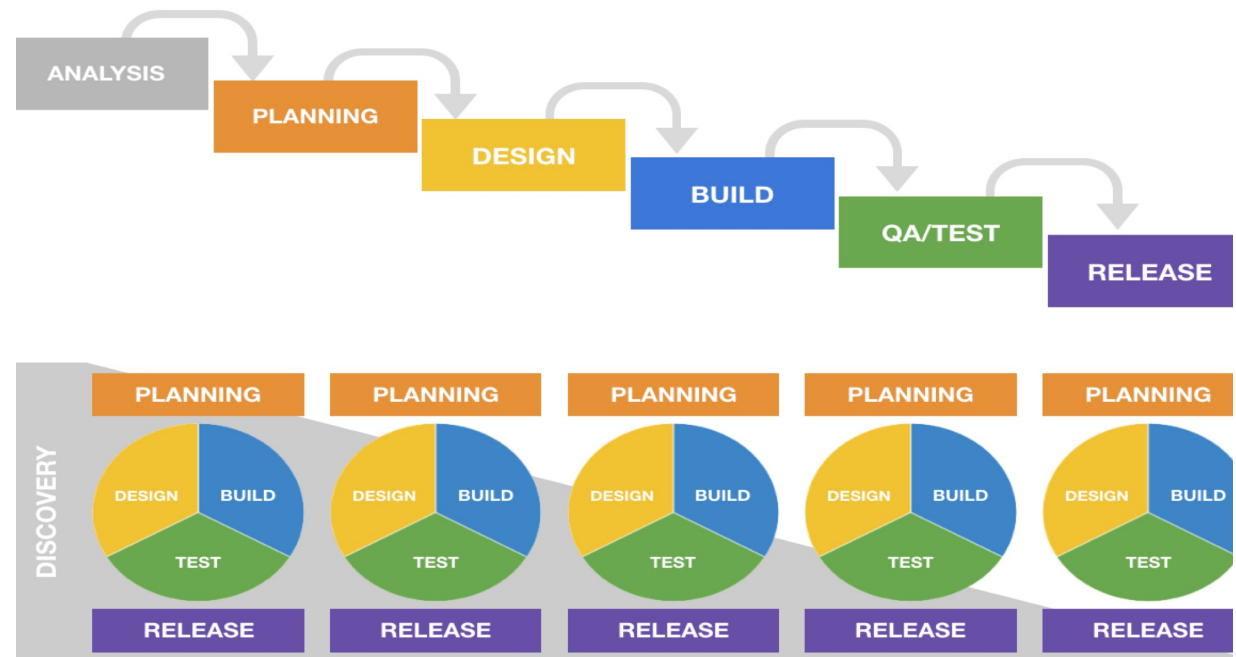
LEAN STARTUP CYCLE

- ▶ We start to learn more about unknowns when we explore and start delivering



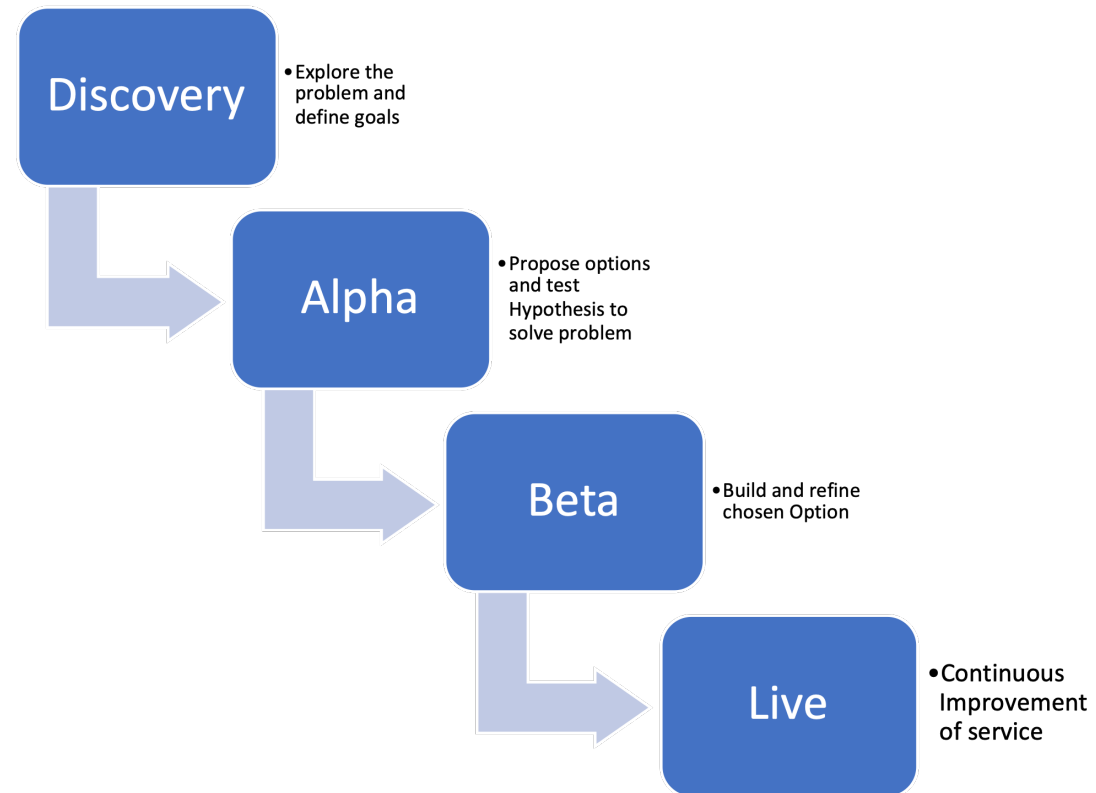
CONTINUOUS DELIVERY TO REDUCE RISKS OF

- ▶ Delivering wrong thing
- ▶ Wasting time and money
- ▶ Inability to change focus and priorities
- ▶ Misalignment of expectations



SERVICE DESIGN PHASES

- ▶ Discovery: start to understand the problem
- ▶ Alpha: explore many options that may address the problem, test them with users to find the route to take
- ▶ Beta: start building the service, then start scaling the service
- ▶ Live: switch other channels off, maintaining and optimising
- ▶ Retiring: safely switching a service off



A short phase, in which you start researching the needs of your service's users, find out what you should be measuring, and explore technological or policy-related constraints.

DISCOVERY PHASE

SHORT (4-6 WEEKS) OR MEDIUM (8 - 12 WEEKS)

DISCOVERY

- ▶ Set clear goals for discovery
 - ▶ It will help with scope and working out when discovery is finished
- ▶ Define the problem
 - ▶ spot problems and estimate how much they're costing
 - ▶ break down and validate assumptions
- ▶ Explore the problem
 - ▶ understand users and their context
 - ▶ the constraints that affect your problem
 - ▶ and any opportunities to improve things
- ▶ Share what you learn
 - ▶ Regular show and tells
 - ▶ Discovery Report



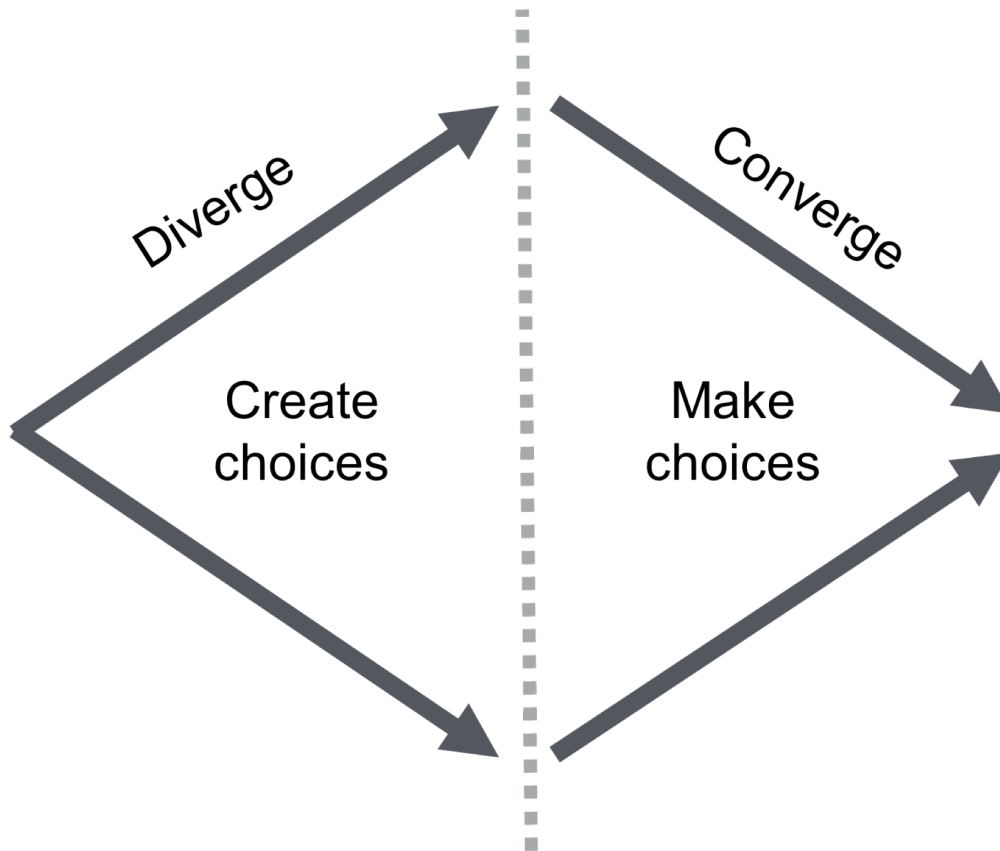
FINISHING DISCOVERY

- ▶ are clear on how what you're working on fits into that wider problem space
- ▶ have a list of ideas you'd like to test at alpha and an idea of which one you'd like to test first
- ▶ know roughly who you need in your team for alpha
- ▶ know how you'll measure whether you've been successful

A short phase in which you will learn through prototyping solutions for your users' needs. You'll be testing a number of options with a small group of users or stakeholders, and getting early feedback about the design of the service and iterating towards beta.

ALPHA PHASE

ALPHA – DIVERGE TO CONVERGE



- ▶ Start with multiple hypothesis and options to solve problem
- ▶ Test hypothesis with real user feedback by prototyping the options
- ▶ Make informed decision to select option(s)

FINISHING ALPHA

- ▶ explain how you came to the decision using the success metrics you identified at the end of discovery
- ▶ you can create something that meets user's needs and is cost-effective
- ▶ you'll have the budget and people necessary to deliver what you need to in beta

YOU'RE DEVELOPING AGAINST THE DEMANDS OF A LIVE ENVIRONMENT, UNDERSTANDING HOW TO BUILD AND SCALE WHILE MEETING USER NEEDS. YOU'LL ALSO EVENTUALLY BE RELEASING A VERSION TO TEST IN PUBLIC AND MAY ALSO TEST IN PRIVATE FIRST.

BETA

BETA DEVELOPMENT



- ▶ take your best idea from alpha and start building it for real
- ▶ roll out the service to real users - while minimising risk and maximising the potential to learn and iterate the service
- ▶ involves thinking about how your service will integrate with existing services
- ▶ and preparing for the transition to live.

FINISHING BETA

- ▶ you've got appropriate metrics in place to measure the success of the service, based on what you've learned during beta
- ▶ you are able to monitor the service and maintain uptime and availability
- ▶ you are able to maintain quality assurance criteria
- ▶ you are able to maintain security posture, vulnerability management
- ▶ you understand what level of continuous improvement it makes sense to support, and who you'll need on the team

THE WORK DOESN'T STOP ONCE YOUR SERVICE IS AVAILABLE TO THE PUBLIC. YOU'LL BE ITERATIVELY IMPROVING YOUR SERVICE, REACTING TO NEW NEEDS AND DEMANDS, AND MEETING TARGETS SET DURING ITS DEVELOPMENT. YOU'LL EVENTUALLY BE SWITCHING OFF ANY EXISTING SERVICES AND REMOVING THE BETA BADGE.

LIVE

LIVE DEVELOPMENT

- ▶ continue to address any constraints you identified at beta
- ▶ continue to develop the service and work with others in the organisation providing services that are part of the same journey, so that you're iterating towards solving a whole problem for users
- ▶ transition or integrate any existing transactions that meet a similar need to yours - making sure that what you end up with has a scope that makes sense to users
- ▶ service should be backed by a plan and budget that allows for continuous large and small improvements throughout its lifetime

BUILDING HIGH FUNCTIONING TEAMS

ONE TEAM

- ▶ Social Integration - Team member feels accepted by their peers and superiors
- ▶ Self-Efficacy - Team Member feels they are doing their job well
- ▶ Team member knows how to navigate client/team's unique culture and is comfortable with ways of working within the team
- ▶ Role clarity - Team member understand their role, roles of others in the team, and expectations