MadeTech

Architecture Playbook

Achieving architectural excellence in the public sector

About the playbook

The main goal of this playbook is to provide architecture guidance. It is split out into several phases which relate to the major phases of a software delivery, they are contained within the following themes; **inception**, **inspection** and **execution**.

Each phase contains guidance on the key architectural ingredients that contribute to achieving architecture excellence in public sector deliveries.

The playbook is a living document and relies heavily on feedback from its readers to drive improvements, please do not hesitate to ask questions or put forward suggestions.

Who is the playbook for?

The target audience is **Lead Software Developers** and **Architects** working on a service or set of services within a business area.

However, software developers at all levels will benefit from exposure to the concepts in this playbook.

When to use the Playbook

The narrative of the playbook assumes you have arrived on a new project and an unfamiliar account and therefore it is structured in a way that guides you through the initial weeks to gain clarity and confidence in your technical decision making.

However, the playbook activities work just as well for an established project to help find your bearings and quickly get up to speed.

You might also be familiar with the clients ways of working and business objectives so cherry picking steps that feel valuable to you also works well.

Lastly, sometimes you might feel that the technical team is struggling, decision making is hard or delivery of value is questionable. You might find revisiting this playbook identifies the root cause of these issues and helps you take effective action.

How to use the Playbook

We have found the simplest approach is to take the stages and checkpoints and encode them into a clients delivery tracker. This might simply be a page on confluence.

We also found working through each stage and recording the output (this could be links to evidence or output of workshops) is extremely valuable.

When taking this approach it becomes a snapshot of the assumptions and decision making during this period which will be valuable to future team members and the client.

You might choose to maintain this knowledge base throughout the project, adding updates as business goals change or the landscape evolves.

Architecture Playbook Checklist

SS Created by Stephen Strudwick
Last updated: Dec 16, 2022 • 1 min read • 6 people viewed

This page documents the Made Tech Architecture Playbook Checklist which works through the questions that need to be answered before we can confidently define a:

- Target Architecture
- A Minimum Viable Architecture for the MVP
- Possible routes from the MVA to the Target Architecture

The checklist links off to documents or internal pages answering each question.

Finding You Bearings

- [x] Organisational Objectives and Strategy
- [x] How is IT perceived ?
- [x] What is the Digital and Technology strategy ?
- [x] What is driving the work?

Verify The Destination

- [x] Review the Service Vision
- [ ] Review alignment of Business strategy, IT strategy and Service Vision
- [ ] Ensure the User Research aligns with service vision (solving the right problem)
- [x] Workshop to Impact Map the MVP goals

Part 1 – Holistic Thinking

The ‘Right Brain’ of an architect

Understanding the domain and context so you can deliberately design solutions that enable the service to meet its goals and avoid unexpected technical impediments

The Main Stages

Inception

1. **Finding your bearings** - guidance around ensuring understanding of the current situation both of an organisation, it's IT strategy and the drivers behind the delivery
2. **Verifying the destination** - guidance around ensuring a shared understanding around the vision, alignment, impact, goals, trade-offs and strategy before embarking on the journey of delivery

The Main Stages

Inspection

3. **Terrain analysis and optimisation** - guidance around understanding the digital maturity of the organisation sponsoring the delivery, studying and adapting their approach to work towards modern, lean approaches to help reduce friction during the delivery
4. **Squad checks** - guidance around ensuring teams are set-up for success through psychological safety, team skills, shared beliefs of effective software delivery teams and bringing everyone along on the journey

The Main Stages

Execution

5. **Options** - guidance on visualising the delivery options and evaluating them based on risk, setting technical milestones to ensure there are opportunities to course correct and ensuring a flexible minimum viable architecture
6. **Navigating the journey** - guidance around ensuring your team reaches its destination in the safest, most economical and timely fashion that conditions will permit

Phase 1: Find your bearings

Make sure you know your current position

Understand the Organisations Objectives & Strategy

Understand how IT is perceived by the organisation

- **A cost center** - the only thing that matters is reducing cost, typical strategy is to outsource IT
- **An asset** - IT is viewed as a getting high return on investment, typical strategy is around rationalisation and efficiency
- **A partner** - IT has strongly related to generating value for the organisation, typical strategy is to insource IT
- **An enabler** - IT **is** the business and being able to rapidly expand and react to the market is key to success

Understand the IT strategy of the organisation

Understand the main motivation for the work

What is the root cause and does it influence our decision making e.g:

- Regulation demands certain activities and has significant consequences if not met
- Costs must be reduced
- A significant business or technical failure has prompted action
- An important person wants a particular technology

Phase 1: Checklist

- ❑ You can describe the organisations objectives and strategy clearly and succinctly
- ❑ You have identified how IT is perceived by the organisation
- ❑ You can describe the organisations IT strategy clearly and succinctly
- ❑ You have identified the main motivation behind the work

Phase 2: Verify the destination

Make sure you know where you're going

Review the Service Vision

The service vision allows your team to take a top-down approach to your service's development.

Ensure the development team understand and are aligned.

Work closely with the service or product owner but be prepared to take a lead on the service vision if they are unable to articulate it.

Start with a high-level vision statement, then translate that vision into a strategic guide and action plan.

Workshop: Service Vision

Goals

- Determine the first pieces of the service puzzle and how they are going to come together
- Outline the initial path from idea to launching the service and what the strategy for getting there will be
- Define the essence of the organisations value that conveys a clear and convincing message to users
- Collaboratively define the vision of the service

Workshop: Service Vision

Activity overview

1. Show the service vision template (see next slide)
2. Create breakout groups and ask each group to fill in the blanks in the template
3. Gather the results and collaborate to form a sentence that describes the service vision

Workshop: Service Vision

Service Vision Template

For: [final client],
Whose: [problem that needs to be solved]
The: [name of the service]
Is a: [service category]
That: [key-benefits, reasons to user it],
Different from: [current system]
Our service is: [key-difference]

Architecture Assessment: Alignment checks

Goals

- Ensure the organization strategy aligns with the IT strategy

- Ensure any existing research:
  - Is focused around the service vision
  - Contains no gaps or leaps
  - Contains enough data

- Ensure the service vision fits within the IT strategy

- Ensure the entire team understands and is aligned

MadeTech

Architecture Assessment: Alignment checks

Activities

1. Review the four operating models (as shown on the next slide)
2. Based on what you have learned about the organization choose which operating model is the best fit for the service you are delivering
3. Check whether this aligns with what you've learned about the organisation's overall strategy and IT strategy
4. Check whether the service vision fits within the operating model

The Four Operating Models

Organisation Process Integration	Low Organisation Process Standardisation	High Organisation Process Standardisation
High	**Coordination** Unique departments with a need to know each other's transactions Examples: DVLA Key IT capability: access to shared data, through standard technology interfaces	**Diversification** Independent departments with different customers and expertise units Examples: HMRC Platform Key IT capability: provide economies of scale without limiting independence
Low	**Unification** Single organisation with country wide process standards and data access Examples: MoJ Key IT capability: enterprise systems reinforcing standard processes and providing global data access	**Replication** Independent but similar organisations Examples: DLUHC Key IT capability: provide standard infrastructure and application components for global efficiencies

MadeTech

Workshop: Impact Mapping

Goals

- Help delivery teams and stakeholders to:
  - Visualise roadmaps
  - Explain how deliverables connect to user needs
  - Communicate how user outcomes relate to higher level organisational goals
- Can also help verify the MVP and gain alignment

Workshop: Impact Mapping

Activities

1. Show the goal you want to achieve
2. Understand who is impacted by discussing the following:
   - Who can produce the desired effect and who can obstruct it?
   - Who are the consumers or users of our product and how will they be impacted by it?
3. Understand how they are impacted by discussing the following:
   - How should our actors' behaviour change?
   - How can they help us to achieve the goal and how can they obstruct or prevent us from succeeding?
4. Understand the scope by discussing what the organisation, delivery team can do to support the required impacts

Learn more here

MadeTech

Workshop: The service Is – Is Not – Does – Does Not do

Goals

- Help explain the role of service
- Aligns collaborators views of what the service does, as well as what it doesn't do
- Help clarify strategic decisions

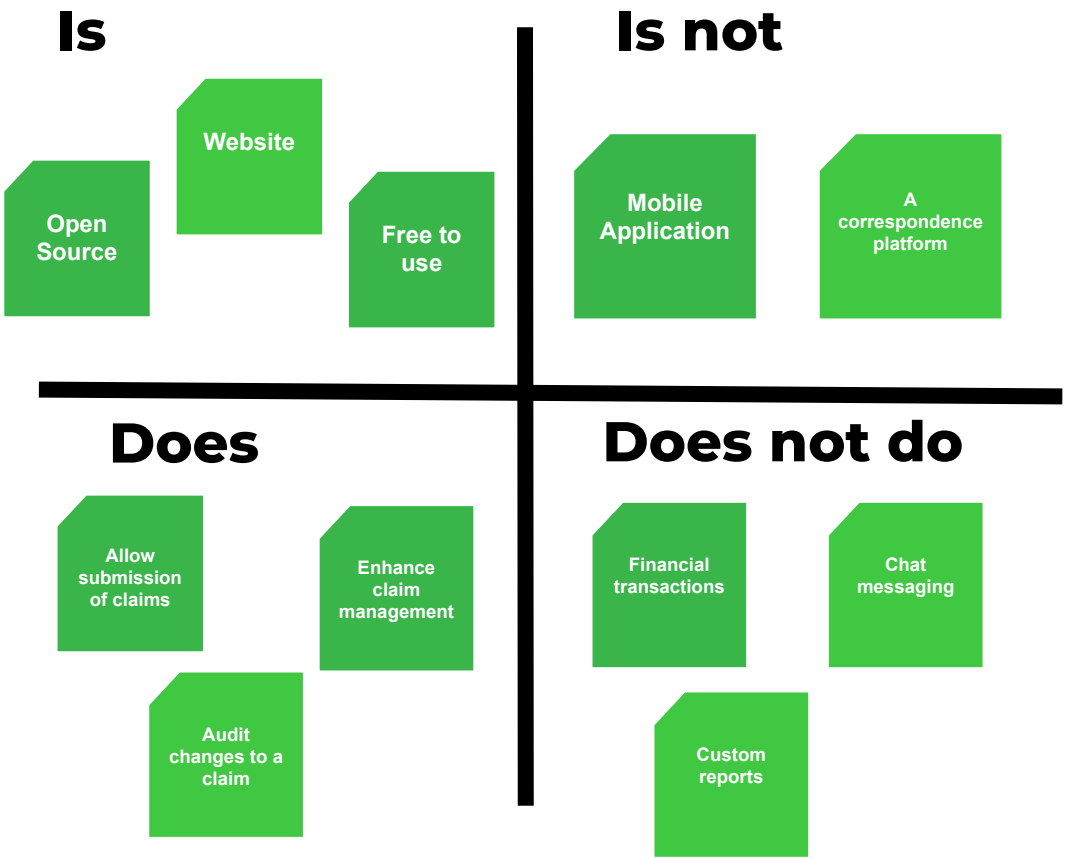
Workshop: The service Is – Is Not – Does – Does Not do

Activity overview

1. Divide a canvas into four areas (Is / Is not / Does / Does not do)
2. Ask each participant to describe the service on Post-It notes and put them in corresponding areas
3. Read and gather the notes

(Tip: If the description is a noun or adjective it goes in the ‘Is’ row, verbs go in the ‘Does’ row)

MadeTech


Is
Website
Open Source
Free to use
Is not
Mobile Application
A correspondence platform
Does
Allow submission of claims
Enhance claim management
Audit changes to a claim
Does not do
Financial transactions
Chat messaging
Custom reports

MadeTech

Workshop: Non Functional Characteristics

Goals

- Help discover silent requirements
- Sets up trade off workshop
- Help clarify strategic decisions

Workshop: Understanding Trade-Offs

Goals

- Build a common understanding of the trade-offs
- Decide what is most important based on the context
- Clarify trade-offs helping avoid misunderstandings and inform rapid decision making

Workshop: Understanding Trade-Offs

Activities

1. Describe all categories that are relevant to the service on post-it notes (e.g. security, performance, usability, etc.)
2. Place the categories in a column at the left hand side of the board and draw horizontal lines under each category - the Y axis
3. At the top of the board create a the X axis which will be numbered post-it notes, the maximum number will be the total number of categories - this represents importance
4. Ask participants to write their initials on several post-it notes and put at most one post-it note on each line - signifying where they feel the categories ranks in terms of importance (they cannot rank two categories at the same levels of importance)
5. Calculate the average level of importance for each category

Validate the service transformation strategy

- Understand alignment with ‘Accelerate’ measurements
  - The better aligned, the higher chance of success
- Consider iterative modernisation first
  - Is the application bespoke?
  - Can it be modernised?
- Next consider iterative replacement
  - Is there a quality off the shelf product the organisation could leverage?
  - Do you have the right team available to build the replacement parts?
- Last resort - Big Bang
  - What will you do if everything goes wrong?
  - Do you have teams in place that are ready to deal with the fallout?

Be mindful of the organisation’s goals and objectives

MadeTech

Visualise the current system

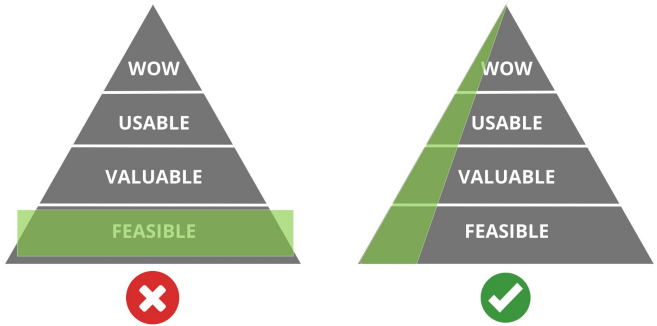
- Helps with understanding how software systems fit together within the bounds of an organisation
- Helps with understanding how the people and software systems are related within the organisation
- Useful for both technical and non-technical people, inside and outside of the software delivery team
- Helps with playing back to the organisation the current situation, the issues and the benefits of changing
- Technical risks have been identified

Verify the MVP

A minimum viable product (MVP) is **the simplest version** of a product that can be made available to the users **to validate a business idea.**

MVP Checklist:

- ❑ **At the highest level, you've addressed desirability, feasibility, and viability**
- ❑ **At a strategic level, you know what to build**
- ❑ **At an executional level, you know how to build**
- ❑ **Before going live, you know what to test**



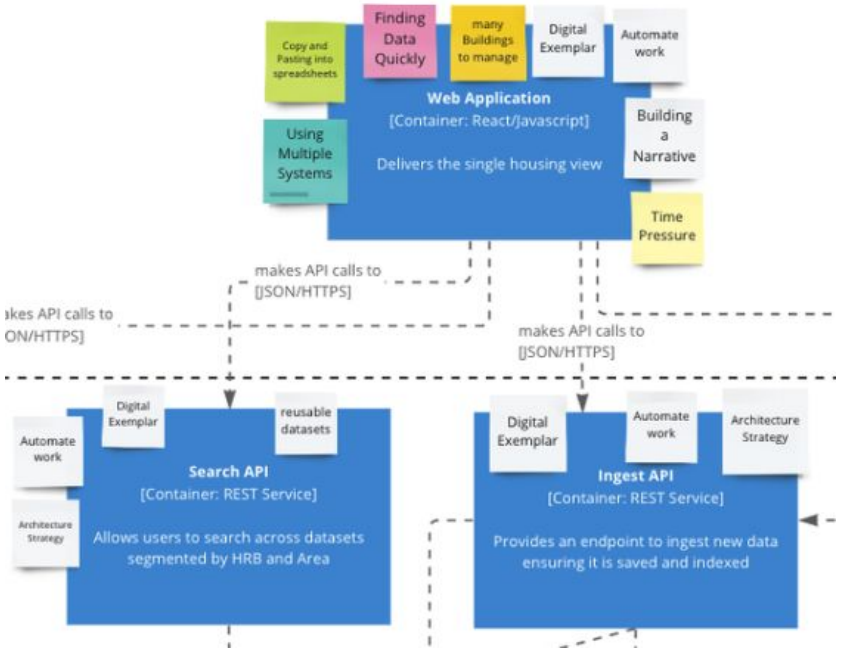
MadeTech

Visualise the target system

- A good target picture is simple and convincing to a diverse audience
- Show how new software systems fit together within the bounds of an organisation
- Show how the people and software systems are related within the organisation will be impacted
- Helps when playing back the target system to the organisation and its benefits
- Useful for both technical and non-technical people, inside and outside of the software delivery team

Demonstrate it is a focused Architecture

- Show how each component of the architecture meets a goal or solves a problem for a user



MadeTech

Play it back to relevant stakeholders

- Present the current and target systems in a simple and concise manner while staying true to technical implementation
- Helps ensure your knowledge of the current system is well understood and covers all bases
- Any Elephants in the room identified
- Helps demonstrate the value of the proposed changes
- Final stage of alignment

Phase 2: Checklist

- ❑ You have a single sentence to describe the service vision
- ❑ You have checked the alignment of the organisational strategy, IT strategy and Service Vision
- ❑ You understand the role of the service as well as who and how it will impact people
- ❑ You have clarified what the service will do and what it will not do
- ❑ You understand your non functional requirements and how they influence architecture
- ❑ You have identified the trade offs and ranked them in order of importance
- ❑ You have validated that the transformation strategy is the best fit for the service
- ❑ You have produced architecture diagrams to visualise the current system
- ❑ You know your MVP
- ❑ You have produced architecture diagrams to visualise the target system
- ❑ You have played back the current and target systems back to key stakeholders and reached an agreement of the MVP

Phase 3: Study and help shape the terrain

Understanding how an organisation approaches software delivery and enabling improvements

MadeTech

Assess digital maturity

7 Lenses Maturity Matrix

	Vision	Design	Plan	Transformation leadership	Collaboration	Accountability	People
	The vision gives clarity around the outcomes of the transformation and sets out the key themes of how the organisation will operate	The design sets out how the different organisations and their component parts will be configured and integrated to deliver the vision	The plan needs to retain sufficient flexibility to be adapted as the transformation progresses while providing confidence of delivery	Delivering a transformation often means motivating into action a large network of people who are not under the direct management of the transformation leader	Collaboration is key to transformation in a multi-dimensional environment that increasingly cuts across organisational boundaries	Having clear accountability for transformation within an organisation enables productivity and improved decision making, and leads to better outcomes	Transformation will require people in your organisation to be engaged and to change their ways of working - you need to communicate effectively with them at every stage of the transformation
5	The vision is embedded in everything people do. It flows from top to bottom and is aligned with public outcomes	The public are at the heart of design work. Outcomes for different changes across the organisation are aligned. It's clear how to bridge the gap between the current and future states	Planning is joined up and fully resourced. Plans adapt as transformation progresses	Leaders embody transformation and create an environment of trust where it's safe to speak freely	The organisation compromises for the greater good and leads the way in transformation communities	Clear governance results in decisions being made at the right level and at the right time to drive progress	Ways of working needed for the future are adopted. Mature workforce planning exists
4	The vision sets a clear direction that people buy into. It is articulated in different ways	It's clear how different parts of the organisation will fit together. It's possible to assess progress as the design evolves	Planning is informed, coherent and mature, supporting both transformation and business as usual	Leaders tell a consistent story. They 'push' and 'pull' as needed to create the right environment for change	Roles, responsibilities and incentives reflect the need to collaborate, leading to new ways of working	People are becoming empowered and accountable for making decisions	Plans to deliver new skills or ways of working are being realised and people are engaged
3	There is a vision that is stretching but achievable. People see how they can fit into it	The design considers users and contains enough examples to bring it to life	Plans have the right level of detail and balance of tight and loose planning	There is sufficient ownership of transformation. Leaders talk about it. There are visible role models	Many decisions are made across boundaries. Shared outcomes are starting to be developed	There is broadly the right structure around transformation, with a focus on making decisions at the right time	Plans are in place to address the impact on people, ways of working and culture
2	A vision exists, but it means different things to different people	The design attempts to define the future in too much detail or doesn't cover everything it should	Plans are beginning to be joined up. Ambition and achievability need more focus	There is support for transformation at the top, and some change agents. There are meetings and ways to submit ideas	There is some understanding of stakeholders. Collaborative behaviour isn't yet commonplace	There is a growing level of accountability for transformation	The impact of transformation on people, ways of working and culture is understood
1	There is no clear vision for the future, or there are competing visions	There is no single design, or various designs are not joined up	Planning is not joined up. Plans are not flexible or achievable	Leaders talk about transformation on occasion. They make some effort to canvass views but avoid difficult messages	Collaboration across boundaries is limited	Responsibilities and accountabilities for transformation are unclear	The impact of transformation on people, ways of working and culture is not understood

MadeTech

Digital maturity: Understanding the work

Study the organisations approach to work, look closely at teams you will be collaborating with:

- Is work visible?
- Is work tracked?
- Is work standardised and documented?
- Has the organisation identified the four types of work?
    - Planned Work
    - Internal Projects
    - Changes
    - Unplanned Work
    - Is there a healthy balance or not ? e.g. does unplanned work dominate ?

Digital maturity: Accomplishing the work

Study how the organisation accomplishes its workload:

- From inception to release, what are the stages work goes through?
- What gatekeeping checks are in place?
- Does the organisation continuously review and improve how work is accomplished?
- Is the work managed through project or product management techniques?
- Is there a difference between the organisations approach and your current delivery?

DevOps Maturity

Development teams should be able to make changes to their projects without impacting other teams, it should be simple to make those changes.

How does the organisation rate against the DevOps Research and Assessments Metrics?

Do they have the CI/CD infrastructure and tooling in place which you can adopt or will your team be ‘breaking trail’

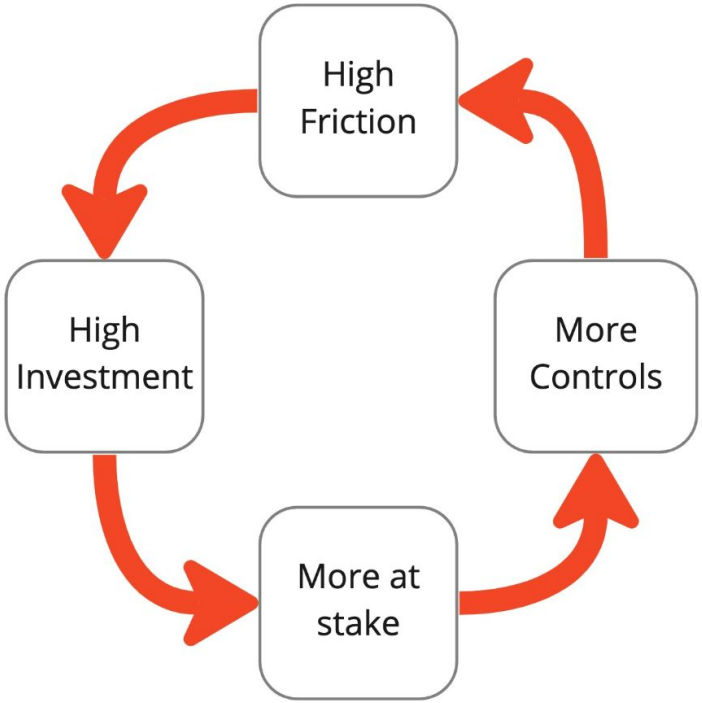
Continuous Delivery

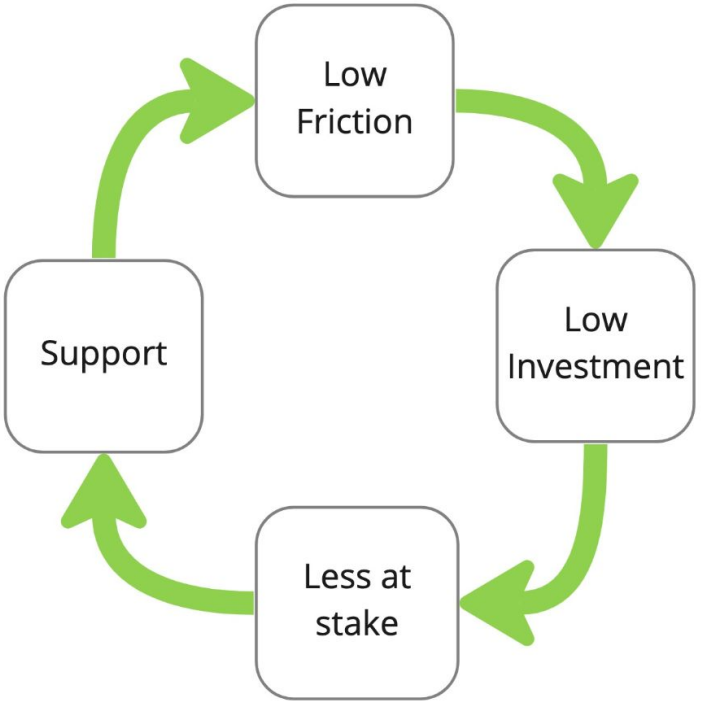
Does the organisation and the delivery teams practice Continuously Delivery, if they don't you must understand what's getting in the way and eliminate waste.

Ask the following questions:

- Is the team working so the software is in a releasable state all of the time? (once every hour?)
- Does the team adopt a "you build it, you run it" approach or is there a strong feedback loop to the development team on the performance of the application through monitoring etc.

Disablement culture vs Enablement culture





MadeTech

Identify and communicate constraints

- Work directly with teams to discover constraints - leverage observation and documentation skills to identify and document the constraints
- Look for constraints across the full development process
- Look for constraints in analysis and requirements gathering
- **Only improvements to the single biggest constraint can improve throughput**

The 5 Ideals

1. Locality and Simplicity
   - Development teams should be able to make changes to their projects without impacting other teams, it should be simple to make those changes
2. Focus, Flow and Joy
   - Working in small batches getting fast and continual feedback
3. Improvement of Daily Work
   - Pay technical debt down and continuously improve the architecture to support CI/CD
4. Psychological Safety
   - Solving problems requires honesty, and honesty requires an absence of fear
5. Customer Focus
   - Ability to gain feedback from a variety of end users

Phase 3: Checklist

- ❑ You have good measure of the organisations digital maturity
- ❑ You have good measure of the organisations DevOps maturity
- ❑ You have understood what the working culture looks like
- ❑ You have identified and clarified the constraints
- ❑ You have checked the alignment of the organisation vs the 5 ideals
- ❑ You have shared your observations with key stakeholders in the organisation that are empowered to make changes
- ❑ You have clarified what will and what will not change
- ❑ You have highlighted the risks with the organisations current approach to software delivery

Phase 4: Squad checks

Ensure the team is set up for success

Psychological Safety

Solving problems requires honesty, and honesty requires an absence of fear.

Mistakes are something to be celebrated, as they present opportunities to learn and grow.

If someone feels unsafe they are unlikely to raise mistakes to the wider team, resulting in missed opportunities to learn.

Exercise: Safety Checks

Primary Goal

To gauge how comfortable team members feel before tackling larger issues

Exercise: Safety Checks

Activities

1. Ask each team member to write a number between 0 and 5 on a post-it, to mean one of the following:
    - 5 – I'll talk about anything
    - 4 – I'll talk about almost anything, but one or two few things might be hard
    - 3 – I'll talk about some things, but others will be hard to say
    - 2 – I'm not going to say much, I'll let others bring up issues
    - 1 – I'll smile, claim everything is great and agree with authority figures
    - 0 – I'm not comfortable talking / I don't want to do this / I want to leave
2. Tally the count - if the numbers are low you can run this exercise to help determine why people are feeling unsafe:
    - Ask everyone to put themselves in the shoes of someone who might not feel safe, then note down what could make them feel that way
    - Based on the submitted notes, have the team work together to list reasons that might cause these issues
    - Based on the potential causes, ask the team to present and discuss potential solutions

Validate the team structure

- Compare the skills of the team with the skills required to deliver
  - Consider creating a skills radar for the team
- Are there any major weaknesses or skill gaps in your team?
- Are there any single points of failure within the team?
- Do the team members trust each other?

Align the teams approach to work

- Discover the software delivery beliefs of individual team members
- Determine the balance between team members that hold siloed development beliefs vs DevOps beliefs
- If the team is heavily weighted towards siloed development consider developing a small application such as a kata guided by DevOps principles
- Leverage pair programming

Clarify and record the decisions that lead to the architectural approach

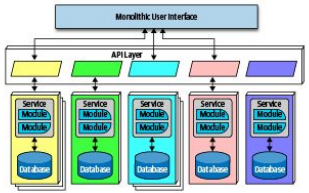
Without clear and concise clarification of the architecture approach developers can get caught up in the *why* and *what* taking focus away the *how*. It's important to:

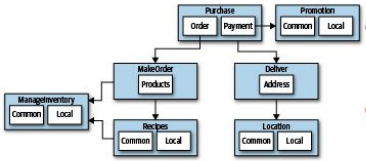
- Make it clear how the architecture approach fits the problem domain, meets service goals and delivers value to users
- Show alignment between the architecture approach and the architecture characteristics (non-functional requirements)
- Clarify the impact that data design might have on the architecture design
- Highlight where organisational factors been taken into account, such as cost
- Capture the trade-offs based on the current context

MadeTech

Record decisions as ADR's within source control

Foster technical collaboration





Leadership

Mentoring

MadeTech

Phase 4: Checklist

- ❑ You have established psychological safety amongst the team
- ❑ You have ensured the team have the skills and experience to deliver the service
- ❑ The team is aligned on how work will be delivered - definitions of ready and definitions of done are clear and all team members agree with them
- ❑ Any up front architecture decisions have been recorded in source control
- ❑ The team understands your role in delivery and trusts your ability to lead and mentor them

Phase 5: Plot course options

In uncertain times it’s good to have options

Visualise the options

- Diagram and present the options
  - Consider using C4 Context and Container diagrams to visualise each of the available solution options
  - Make diagram selections based on the target audience
- Diagrams should reflect reality
  - Include technology details on the diagrams, make technology choices explicit
  - The diagram must reflect how the system would be built
- Present it back to the relevant stakeholders
  - Be mindful of time, as a presenter you control how quickly ideas are unfolding

Identify the risks

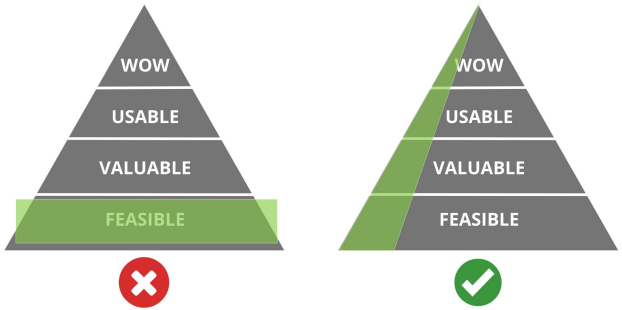
- Use architecture diagrams to help facilitate risk identification
- No one can single-handedly determine the overall risk of a system
  - Risks can be overlooked by individuals
  - Individuals are unlikely to have full knowledge of the entire system
- Pick the right stakeholders when identifying risks
- Ensure risks are identified first by individuals then collaboratively as group to gain a consensus on risk areas, discuss risks and form solutions to mitigate prioritised risks based on impact and likelihood
- Consider using a collaborative approach to identifying risks, such as **Risk Storming** (there is a template for this in Miro)

Compare the options

- There’s always a trade off with architecture decisions
- Base your decisions off of whether they are a good fit for the organisation, the service vision, the trade-offs identified during inception as well as the team
- There is likely no ideal option - opt for the least worst option

Define an Minimum Viable Architecture (MVA)

- Map out things we know must be part of the technical design
- Ensure the MVA is open to extension
- The MVA must be a finished slice of the solution



Set technical milestones

- Provides opportunities for course correction
- Gives the team an opportunity to view the progress through a technical lens
- Helps keep alignment with organisation and service goals as well as the service vision
- A good technical milestone delivers clear value to the business or users
- Milestones can be thought of as a series of MVP's and MVA's that can be composed in any order

Phase 5: Checklist

- ❑ You have visualised each of the options highlighting the pro's and con's of each option
- ❑ You have compared each option in collaboration with the team and chosen the least worst option
- ❑ You have collaborated with the team to identify risks and have mitigations for each risk
- ❑ A MVA has been created that is a finished slice of the full solution
- ❑ Technical milestones have been identified to help you move into the future

Phase 6: Navigate the journey

Ensure your team reaches its destination in the safest, most economical and timely fashion that conditions will permit

Architecture Evolves

- Every service, component or even class has a purpose
- Composable and loosely coupled allows growth
- Anchor each iteration on learnings from phase 1 and 2 , you might need to adjust your direction and goals

Look out for obstacles

- Obstacles can come from any direction
- The sooner an obstacle is identified the better
- Ensure effective monitoring is built in

Periodic Squad Checks

- Identify potential problems before they become real ones
- Review the critical components of your team and the delivery:
  - Does the team have the risk skills and expertise?
  - Does the team have the right tools to do the job?
  - Is the team utilising CI/CD effectively, can it be improved?
  - Are all the necessary feedback loops in place to help maintain the course?

Reflect on progress

- Review the changes that affect the delivery
  - Has there been a major change in the context that could affect the technical approach?
- Perform **technical** retrospectives at key technical milestones of delivery:
  - Are our technology choices helping or hindering?
  - Have we introduced any unnecessary dependencies?
  - Is the architecture still aligned with the goal?
  - Are we hitting the mark on the architecture characteristics? Are they still relevant?
  - Is our technical debt manageable?

Course correct

- It's crucial to recognize when a mistake is a mistake, to learn from any shortcomings, and then to change course and move forward
- Negotiate with the product owner for time to course correct
- Prevent technical debt from becoming insurmountable
- Have a backup plan

Phase 6: Checklist

- ❑ You implemented mechanisms that allow you to measure and monitor the health of service and the delivery
- ❑ You have created a service schedule to monitor the critical components of the team and delivery
- ❑ You have decided on the frequency in which you will reflect on journey progress
- ❑ You have a backup plan and ready to course correct should the need arise