HackIt - service design tactics manual

This tactics manual helps you take a problem and start to solve it, using the principles of [service design](https://en.wikipedia.org/wiki/Service_design).

WE’ve used all of the tactics before, most of which have been borrowed from organisations like

## Why

When faced with a problem, we have traditionally asked individuals to develop a proposal for a project to fix the problem. It can take time before work begins to really understand the problem, and we define the solution without certainty about whether it will work.

This manual is rooted in an alternative approach: working in small teams to examine a problem from multiple perspectives, developing multiple hypotheses which are then tested by developing rapid ‘prototypes’. These prototypes help us find the right solution, reducing the risk that our initial ideas won’t succeed.

The benefits of this approach are:

* **Working faster** - developing key elements of a solution in days
* **Reducing risk**, checking an approach at low cost and learning about what works before making a larger investment
* **Improve outcomes,** particularly where user behaviour is critical, by defining how a successful approach needs to work

## How

You can also use any one of the tactics to help a team work together to understand a problem, develop hypotheses to solve it and then test solutions rapidly with users. We have used each of these tactics on previous service design projects, and collated them in a single manual[[1]](#footnote-0). You’ll want to use different tactics at different times on a project in order to help build a team and ensure it works efficiently to tackle complex problems.

You can combine most of these tactics to run a 5-day mini project, known as a ‘Google Sprint’. This is a device used by Google Ventures to deliver rapid improvements to a product or service.

Whilst these tactics are focused on delivering solutions that have an element of technology, they have been used successfully for many different forms of product and service development in government and the private sector.

[**Why**](#_db9986fqypt8) **1**

[**How**](#_4jx5wnfcyqw1) **1**

[**Setting up a Google Sprint**](#_mlpniiwke828) **4**

[The bigger the challenge, the better the sprint](#_x7ijkgndkrs6) 4

[Setting up the team](#_imryolybdhky) 4

[Time and space](#_qa2t5tarzxkg) 4

[For inspiration](#_5xhirj19r0ka) 5

[The timetable](#_9ol1x36ebobp) 5

[**Tactic #1 - Setting up a team**](#_ra0hrzrim8vd) **5**

[**Tactic #2 - Set an ambitious goal and make it achievable**](#_37wr7fb2rdye) **5**

[Make it achievable](#_r2gg6763cym6) 6

[**Tactic #3 - Draw a map of the service**](#_noao07o47m7o) **6**

[**Tactic #4 - Interview the experts**](#_a3zoyaku6pq2) **7**

[**Tactic #5 - 5 Whys**](#_aqspix3amc6d) **7**

[**Tactic #6 - Develop personas to visualise your users**](#_u9b45zar0r1) **8**

[For inspiration](#_dukn5melmryk) 8

[**Tactic #7 - How Might We**](#_k5msl5qxc9y2) **8**

[**Tactic #8 - Dot voting**](#_vbbdis1p2mll) **9**

[**Tactic #9 - Develop a vision**](#_xk84ndmzzg82) **9**

[**Tactic #10 - Lightning demos**](#_716wo8edzucg) **9**

[Make a list](#_hcpitgk3iqh7) 9

[Give a demo](#_fvfm2ylnvvpj) 10

[Capture big ideas](#_oha2yrj4dclo) 10

[**Tactic #11 - Sketch your ideas**](#_99cmyt5aj0xy) **10**

[Making a decision](#_rrvl71ipkwwa) 11

[A rumble](#_akknt3nrezuo) 11

[**Tactic #12 - Storyboard the journey**](#_bkw1qlwglp8i) **11**

[For inspiration](#_v25ke4smks4y) 11

[**Tactic #13 - User stories**](#_f0ru157gdd2w) **11**

[For inspiration](#_f2j4rku7g75c) 12

[**Tactic #14 - Prototyping**](#_j5rkpfhqfago) **12**

[For inspiration](#_1mxyddvgk20d) 12

[**Tactic #15 - Interviewing users**](#_gsi3ysuyw2xi) **12**

[**Tactic #16 - Writing business rules**](#_aeqax44j0h7v) **13**

[**Tactic #17 - Show and Tell**](#_fmarsux7lssh) **13**

## 

## Tactic #1 - How to align the team

If the team hasn’t worked together before it can be useful to break down barriers and develop a shared understanding of what people want. Give everyone a pad of Post-Its and ask them to write down:

1. what they would like to get out of the project and,
2. on a separate Post-It, what they would like to give to the project.

Ideally, the facilitator would collect these up, put them on a wall and lead a discussion around each of the Post-Its. This technique helps the group share the comment, without necessarily associating it with the person who made it. This can help break down barriers if the team hasn’t worked together before.

This exercise should help identify:

* Shared and / or conflicting priorities
* Any gaps in skills or perspectives
* Any commitments that people have outside the project team

For inspiration

Example:

[The secret to good team working](https://qz.com/625870/after-years-of-intensive-analysis-google-discovers-the-key-to-good-teamwork-is-being-nice/)

## Tactic #2 - How to set an ambitious goal and make it achievable

If you’re working in a short timeframe, it’s important to remain focused on a longer-term goal. This tactic is useful if you aren’t sure that you are going to setup a single project or don’t have a clear brief to work to.

Effective goals will normally define an outcome rather than an output. For example, the following is a better goal:

“residents having **enough confidence and trust** to **self-serve online** rather than **picking up the phone**”.

Than

“A website that tells residents everything they need to know so that they don’t pick up the phone or write to us”.

Because:

* It refers to an outcome (self-serve online) rather than an output (a website)
* It reflects *why* people will change behaviour (confidence and trust) rather than just *what* they will do differently (picking up the phone)
* It avoids guessing what the solution might look like

If you’re working on a project with a clear brief, it may be better to agree [a vision statement](#_xk84ndmzzg82).

#### Make it achievable

Once you’ve set the goal

1. What questions do we want to answer in this sprint?
2. To meet our long term goal, what has to be true?
3. Imagine at the end of the project, we’ve failed. What caused that?

Rephrase assumptions and obstacles as questions.

## Tactic #3 - How to understand the whole journey

Local government services can be complex. Often we only provide part of the service that a resident needs. For example, we provide a service tot enable residents to apply for a school place. But before they can do that, they want to understand which schools are right for their child. Mapping the customer journey helps you understand the whole process, including elements that aren't covered by our service today.

On the left hand-side of the board, write the key stakeholders involved in delivering the goal. For example, in housing repairs it might be tenants and leaseholders, contact centre, repair staff.

Write out the [critical path](https://en.wikipedia.org/wiki/Critical_path_method) that need to be followed to produce the end result. For example, eating a cake involves a baker, a shop and a customer. The baker bakes a cake and sends it to a shop. The shop advertises it, a customer buys it and then eats it. This is a more [sophisticated map](https://zapier.cachefly.net/storage/photos/7eab738ac3810ab574c1404854d63e70.png) showing how a patient goes from making an appointment to beginning therapy.

Keep this simple for now, avoiding the complexity of rules. For example - ‘assess eligibility’ is better than setting out all of the qualification rules.

Remember that the end result might not necessarily be where the council service stops.

### For inspiration

[An example map](https://zapier.cachefly.net/storage/photos/7eab738ac3810ab574c1404854d63e70.png)

## Tactic #4 - Interview the experts

You can accelerate your understanding by interviewing just four different experts. Try to interview them as a group to develop a shared understanding. If that’s not possible, then make sure the interviewer presents the findings to the group.

Interview the project sponsor. Understand:

* What will make this project a success?
* What’s our unique advantage or opportunity?
* What’s the biggest risk?

Understand the customer:

* What are they doing currently to solve this problem?
* Where are the key pain points?

Understand the subject matter expert:

* What are the constraints of legislation or policy?
* What are the opportunities and challenges of existing technology solutions, service design and regulatory arrangements?

Understand previous efforts:

* What have we already tried?
* To what extent has it worked and what can we understand from previous attempts?
* Why does the problem remain?

There are key rules to sharing the outputs:

* Iterate your existing work, making sure the interviews add colour and detail to your other work - eg amend the map
* Make the findings visible - eg key statistics on the wall, key quotes
* Share it across the team - present your findings

For inspiration

## Tactic #5 - Defining the real prpblem

One of the key challenges in any piece of work is understanding the real problem that you’re trying to solve. For example, are customers not using computers to access council services because they don’t want to, don’t know they can, don’t know how to or don’t have access to the technology?

The 5 Whys is a technique used in the Analyze phase of the Six Sigma DMAIC (Define, Measure, Analyze, Improve, Control) methodology.  
  
By repeatedly asking the question “Why” (five is a good rule of thumb), you can peel away the layers of symptoms which can lead to the root cause of a problem. Very often the ostensible reason for a problem will lead you to another question. Although this technique is called “5 Whys,” you may find that you will need to ask the question fewer or more times than five before you find the issue related to a problem.

### For inspiration

A fun [example of 5 Whys](http://www.velactionstore.com/content/assets/36/364415/Product_Images/Module_Product_Images/5_Whys/5-Whys-Exercise-Solution-Sample.png)

## Tactic #6 - Understanding your users

Personas can help give an identity to the types of users we’re working for, and bring to life their needs and challenges. Talking about them collaboratively can help a team with different expertise develop a shared understanding of what they need to do.

Personas can provide a reference point as you work through the service design process, so that you can check back and ask whether the proposed solution really would meet the needs of the persona.

There are lots of ways to develop personas, but we favour a light-touch way which emphasises high-level hypotheses over detail.

1. Divide a sheet of A4 into 4 boxes
2. In the top left box, give the persona a picture, firstname, age and role (eg user, practitioner etc)
3. In the top right box, introduce them with pertinent points that affect the way they engage with the service
4. In the bottom left box, describe their experience of the current situation in 5-7 bullet points
5. In the bottom right box, describe their experience of an ideal ‘future state’

Ask participants to introduce their persona to the group and present their thoughts. They should not be challenged, but after everyone has introduced a persona, you may wish to conduct 2-3 further rounds of sketching.

#### For inspiration

An [example persona](https://drive.google.com/open?id=0B0tr28lhzP30eUVEaldSVUhTUTQ)

[Personas and user journeys in design thinking](https://medium.com/@arnevanoosterom/personas-journey-mapping-some-thoughts-on-implementing-design-thinking-7a546140facc#.gsr85v4jz)

## Tactic #7 - How to empathise with users

There are lots of good ideas that just don’t work in practice. Sometimes that’s because the people providing teh service don’t understand how people use the service. For example, Facebook only made its mobile app really good when Mark Zuckerberg, the founder, banned access to the desktop version in Facebook HQ. Forcing staff to use the mobile app meant they understood what wasn’t working.

Take your team outside the council offices and put them in the shoes of users. For example, if you’re developing a reporting service, go out into the street and get the team to report problems they see.

## Tactic #8 - How Might We

It can be easy to jump towards specific solutions before you have enough evidence to be sure that any one solution is right. Once you have a solution in mind, it can be increasingly difficult to see its flaws.

This tactic enables a group to examine the key question or challenge, rather than jumping to a solution. Key challenges are expressed as open questions - eg ‘How might we give residents reassurance that we are working on their request?’ in order to generate more ideas.

Each person writes their own notes, one at a time on different Post-Its. Each Post-It begins ‘How Might We’ (write HMW in the top corner for speed). Convert your thinking into questions. These will often be responses to the interviews

Organise the notes in themes or groups on the wall. Give each theme or group a label.

For inspiration

[Example](https://goo.gl/photos/VCRPyfkAjMitmj6Q8) of ‘How Might Wes’ in Hackney’s housing service

## Tactic #8 - Making quick decisions

It’s important to involve the team in decision-making, partly to reassure everyone that good ideas which you don’t have time to focus on won’t be lost. Getting people to vote through a show of hands can be difficult in a hierarchical context.

Dot voting enables you to quickly come to a group decision.

1. Give a dot (or a pen to make a dot) to each participant.
2. Ask everyone to review the goal and sprint questions
3. Ask everyone to vote (2-4 times, depending on the number of Post-Its) for the most useful ‘How Might We’ questions.
4. Consider giving the Decider double the number of votes
5. Take the top HMW Post-Its and put them on the relevant part of the map

For inspiration

[An example](https://goo.gl/photos/Mi7vgA6rod7J93Jr9) of dot voting

## Tactic #9 - Developing a vision

Vision statements traditionally take a long time to develop. However, it can be useful to have a shared vision across a team of what you’re trying to achieve - particularly if people haven’t worked together before, or if part of the team has already developed a view of what the solution should look like.

This template helps rapidly develop a vision statement for a project team:

**A** [insert the name of the thing - eg ‘digital service]

**For** [insert the types of people who will use the thing - eg older people living alone in Hackney]

**That** [what does it do? - eg. helps them chat to other older people]

**So that** [what benefit are we trying to deliver - eg. they feel less lonely]

**Unlike** [what’s your key point of difference - eg. charity services that have negative connotations]

Ideally your vision statement will take less than 20 seconds to say. Print it out on a big sheet and make it clearly visible to all the team.

## Tactic #10 - Getting inspiration from other sources

It can be particularly hard to solve a problem that you’ve experienced for some time. In public services there might be a shortage of innovation or inspiration from other organisations. So it’s useful to look at very different services to understand what they do well, and what you can learn from them.

Lightning demos can help a team develop a shared idea of what good looks like. By taking examples they aren’t necessarily used to (or haven’t studied in detail) it can trigger a set of ideas that can be applied to the task.

#### Make a list

Ask everyone on your team to come up with a list of products or services to review. These should be inspirational. Remind people to think outside of local government;)

#### Give a demo

Each person who suggested a product or service should give a brief presentation to the whole team, showing what’s good about it.

#### Capture big ideas

Write on the whiteboard the big ideas behind the products that might be useful. Make a quick sketch of that inspiring component, write a simple headline above it and note the source underneath.

When set alongside your map, you’ve now got:

* an initial area of focus,
* insights about what matters most, what we already know, and what constraints we face
* external stimuli - what does a great product or service look like?

## Tactic #11 - Capturing ideas

People often have lots of existing ideas of how to solve a problem. When they’re described it can lead to confusion.

Sketching is really important as a means of expressing your ideas quickly, identify any key uncertainties and provide a means to evaluate the idea.

Work alone - it’s important to give people the time and space to come up with their own ideas. But share regularly: someone else’s idea will stimulate another person’s thinking.

Take notes (20 minutes)

Encourage the team to walk around the room, exploring the assets you’ve put on the wall, taking notes. These don’t have to be structured but should capture the key learnings and implications - gathering your thoughts in one place.

Generate ideas (20 minutes)

A free sketching session - doodles, experiences etc. The team might want to present back their ideas, in order to generate more. You don’t have to show the whole service - core components, particularly screens, ‘moments’ - whatever people wish.

Rapid variations (5-10 minutes)

Try to generate alternative ideas to the ones you produced first. Look for radically different rather than variations on a theme.

Solution sketch (30 minutes)

Call the team back into a group. Figure out the details of the product or service. If you’ve a strong sketcher, they might ‘hold the pen’ whilst the team talk through each step. With any format, there are some key rules to follow:

* Make it self-explanatory
* Keep it anonymous
* Ugly is ok
* Words matter
* Give it a catchy name

#### Making a decision

There are five steps to making a decision of which solution to take forward:

1. Display the sketches on the wall
2. Use dot voting to mark interesting parts
3. Discuss the highlights of each solution
4. Vote for one solution each (you might want an anonymous ballot if there are strong personalities in the room, or if people are used to a hierarchy)
5. The Decider makes the final decision

#### A rumble

If you can’t decide on one idea, you can have a competition by splitting the teams and each team works on their preferred prototype. Think ‘the Apprentice’ without the silly team names or crap management.

## Tactic #12 - Storyboard the journey

Create a storyboard to show the service or product from start to finish

1. Draw a grid
2. Choose an opening scene: how do your users find the service exists?
3. Complete the storyboard
4. Include sufficient detail that no one has to ask ‘what happens next?’ or ‘What goes here’
5. When in doubt, take risks - don’t look for small fixed but favour risky solutions to test
6. Keep the story to 15 minutes or less

### For inspiration

[How to build empathy maps](https://uxdict.io/design-thinking-empathy-map-c69ab5d6b22#.5vrtc0yg3)

## Tactic #13 - User stories

User stories are a device for capturing what the user needs from the solution. By explaining what the user needs, it enables a designer and developer to work together to find the best way of meeting this need. This can be preferable to detailed documentation that can inadvertently add extra complexity or fail to explain what you really mean.

A user story has three parts:

1. As a [type of user]
2. I want to [a thing the user wants to do]
3. So that [why they want to do it]

Once you’ve created lots of user stories, you can prioritise which are most important to develop in the next phase. The remainder are valuable to keep for the next stages of development.

### For inspiration

[Writing good user stories](https://barryoreilly.com/2013/10/21/how-to-implement-hypothesis-driven-development/?utm_content=buffer2880f&utm_medium=social&utm_source=twitter.com&utm_campaign=buffer)

## Tactic #14 - Prototyping

There are lots of ways of prototyping- from drawing pictures of what something might look like, [wireframing](https://en.wikipedia.org/wiki/Website_wireframe) or developing the web interface that a user would use, without the data or software integration that might be delivered in a full working service.

It’s important to develop a prototype that’s good enough that the user can provide useful feedback. This is often called a ‘minimum viable product’ - the least you can get away with vs enough to be viable.

Sometimes, startup companies will have just a webpage which ‘pitches’ the product as ‘coming soon’ and offers a box to sign-up for further information. If they get lots of sign-ups, they know it’s an attractive proposition and might develop it further. If there are few, they’ve saved the cost and time involved in actually building the product.

### For inspiration

[Prototyping for dummies](https://speakerdeck.com/stephengill/prototyping-for-dummies)

[How to prototype in your web browser](https://designnotes.blog.gov.uk/2016/10/03/how-to-prototype-in-the-browser/)

[Prototyping through chatbots](https://medium.com/ideo-stories/chatbots-ultimate-prototyping-tool-e4e2831967f3#.375klcrvo)

## Tactic #15 - Interviewing users

If you interview just five users, you will get important insights about how your product works. The following are essential to ensure that you get valuable insights from the interviews:

* Ask permission to record the session, but make clear that you will not keep their details unless they ask otherwise
* Explain that they’re helping test a service - make clear you’re not testing them!
* Make clear that you had no hand in building this - so that they don’t hold back from saying what they think
* Give them a task before they start - eg ‘I’d like you to try to apply for a passport’
* Ask open ended questions - eg ‘what do you expect to happen next’
* Ask them to talk to you about what’s going on in their head - eg ‘What do you think when you see this?’

We’ve written a quick guide for how to conduct [usability testing](https://docs.google.com/document/d/1hEe7UQiFAzr4FRjQtcnQW7ci5m2iJg0wQLokSefrN-g/edit?usp=sharing).

## Tactic #16 - Capturing business rules

Our services are conditioned by the business and legislative rules that determine how we can design something. The ‘gherkin syntax’ provides a common way of capturing these in a simple format.

The structure of a Gherkin syntax is:

* Feature
* Scenario
* Given, When, Then, And, But (Steps)
* Background
* Scenario outline
* Examples

This is an example from a cash machine

* Feature: Account Holder withdraws cash
* Scenario: Account has sufficient funds
* Given the account balance is £100
* And the card is valid
* And the machine contains enough money
* When the Account Holder requests £20
* Then the ATM should dispense £20
* And the account balance should be £80
* And the card should be returned

## Tactic #18 - challenging the group

One of the difficulties of group-working is how rapidly people cluster around a solution. Often they’ll disagree vehemently about the details, but agree about the broad framework. This agreement helps you make rapid progress together, but makes it harder to view a potential solution from a more critical perspetive.

Break your team in two parts: one ‘takes’ a 'blue pill’ and the other a ‘red pill’. From the point of ‘taking’ the pill, that team has to play a role. The blue pill team have to produce a presentation about all the strengths of the solution. The red pill team have to produce a presentation about the weaknesses and risks of the solution. They should present their findings to each other and engage in ‘robust’ debate. If possible, ask someone independent to judge which team wins.

Use the findings of both teams to re-asssess your proposal:

* where might it fall short?
* where does it not work?
* what needs to be improved?

## Tactic #19 - Presenting your work

Working in this way prioritises producing things to show over written documentation. The speed at which you’re working favours regular, light-touch updated rather than less frequent, but longer, project board meetings.

A ‘show & tell’ should have two parts: What we’ve done over the last period and What we’re going to do over the next period. There are some key rules to running a successful show and tell:

* Keep it informal - preferably people standing-up
* Have something to show, which is work in progress
* Capture comments as you go

## Tactic #20 - Improving team performance

The closer a team works together, the greater the opportunity for friction. If this is left unaddressed it can slow down, or prevent progress on a project. Equally, if the project does not have the full commitment of every team member, it can lead to resentment. At the end of each phase of work it is valuable for the team to assess its performance and air any challenges.

Ideally, work with an independent facilitator - failing that, the team leader can facilitate but should not participate in the discussion. Before starting, it can be useful to read out a statement reminding the team that “we believe that everyone did their best, given the circumstances and what they knew at the time”. Remind people that the discussion should “remain in the room” rather than continuing afterwards.

Give everyone some post-it notes and ask them to write what went well - putting one thing per post-it. Put these on a wall and lead a discussion around each point. Avoid individuals ‘presenting’ their thoughts can help avoid issues becoming personal.

Next, ask everyone to capture thoughts on what didn’t go well. Facilitate the discussion in the same way.

Finally, ask everyone to capture thoughts on what they will do to make things better. Facilitate the discussion.

Then, encourage the team to take a break!

## Setting up a Google Sprint

A ‘sprint’ is a period of time, usually a fortnight, in which a team tries to produce a solution to a problem. A team at Google devised a more intensive schedule to go from problem to solution in a week. You can combine all of these tactics into an intensive, 5 day course, to develop a solution.

The advice below helps you set up a sprint.

### The bigger the challenge, the better the sprint

Google Sprints work best when you’re facing a big challenge and aren’t sure where to start. It works best when:  
1. The problem is high stakes - a big problem which is likely to require considerable investment of time and money to solve. A sprint can check the direction of travel to reduce risks of heading off in the wrong direction

2. Not enough time

Helping generate good solutions, quickly

3. Just plain stuck

Generate a fresh approach to problem solving

### Setting up the team

Ensure a decision maker is on the team. Sell it in the following terms:

1. We’ll make rapid progress
2. It’s an experiment
3. Explain what you will sacrifice to do this (in terms of existing commitments)
4. It’s about focus

The ideal team is seven people or fewer - enough so that everyone can contribute with different perspectives, but not so many that some people contribute less. Consider putting together people who don’t usually work together. Consider the following roles:

1. Decider
2. Policy expert
3. Comms expert
4. Frontline staff expert
5. user expert
6. Design expert

Pick a facilitator, and try to find a troublemaker - someone with a specific remit to think differently to the group and challenge its assumptions. If you need extra experts, schedule them for day one.

### Time and space

Block five full days in the calendar

Allow no devices in the room - it’s ok to leave to check a device

You’ll need:

Big whiteboards

Post-Its

Sharpies

A4 and A3 paper

### For inspiration

[How Zapier uses Google Sprints](https://zapier.com/blog/google-ventures-design-sprint/)

[Speed up your team with a service blueprint](https://blog.practicalservicedesign.com/speed-up-your-team-with-a-service-blueprint-399a011121a)

[Service design at the BBC](http://www.bbc.co.uk/blogs/internet/entries/4c06d6ef-3d11-4897-9588-7a55deaa1f32)

### The timetable

1. We have learnt from the GDS Service Design Manual, 18F’s playbook, the USDS, Australian Digital Transformation Office, Lean Startup and Google Sprint [↑](#footnote-ref-0)