

Introduction to Software Innovation

INTRODUCTION TO SOFTWARE INNOVATION

TIM MILLER

The University of Queensland
Brisbane, Queensland



Introduction to Software Innovation Copyright © 2026 by The University of Queensland is licensed under a [Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License](https://creativecommons.org/licenses/by-nc-sa/4.0/), except where otherwise noted.

CONTENTS

Title page	vii
Acknowledgment of Country	ix
1. Introduction	1
<u>Part I. Customer Research</u>	
2. Customer Discovery	15
3. Interviews	27
<u>Part II. Defining the Opportunity</u>	
4. Business Model Canvas	43
5. Value Proposition Canvas	48
<u>Part III. Experimentation and Validation</u>	
6. Experimentation: Testing and Validation	59
<u>Part IV. Building the Solution</u>	
7. Prototyping	71
8. Minimum Viable Product	76
9. Customer Relationships and Channels	84
Open Textbooks @ UQ	95

TITLE PAGE

First published by The University of Queensland in 2026

The University of Queensland
Brisbane QLD 4072
Australia

© The University of Queensland, 2026

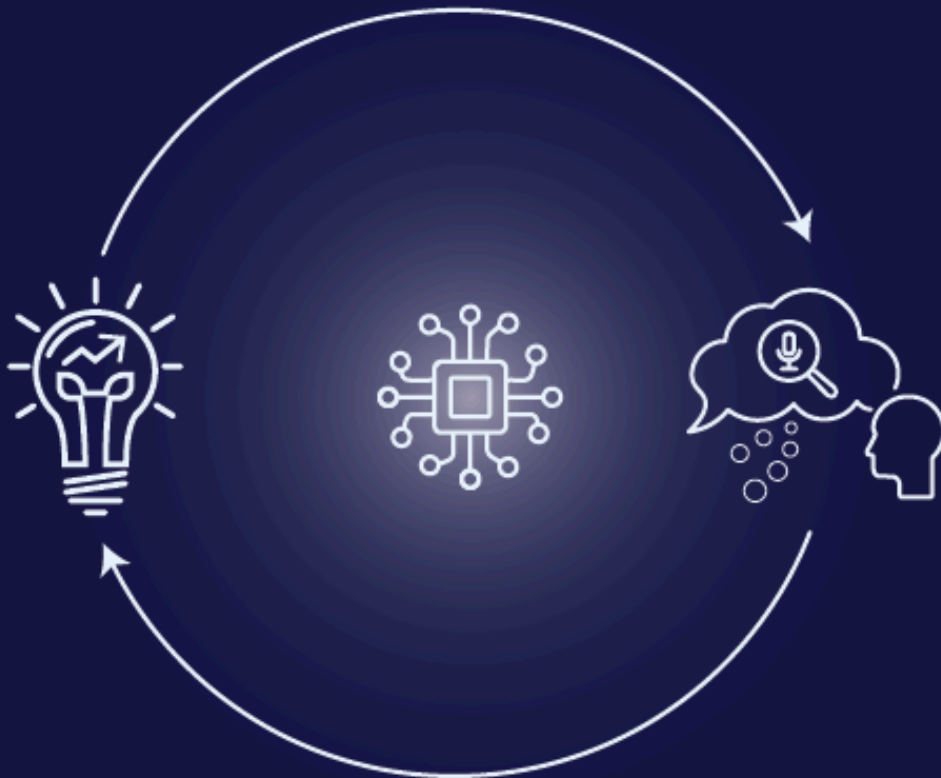
This book is published under a [Creative Commons Attribution-NonCommercial-ShareAlike 4.0 licence](#), unless otherwise noted.

DOI: 10.14264/3d66f1e

eISBN: 978-1-74272-485-0

Please cite as: Miller, T. (2025). *Introduction to software innovation*. The University of Queensland. DOI: 10.14264/3d66f1e

INTRODUCTION TO SOFTWARE INNOVATION



TIM MILLER

ACKNOWLEDGMENT OF COUNTRY

We acknowledge the Traditional Owners and their custodianship of the lands on which this project originated. We pay our respects to their Ancestors and their descendants, who continue cultural and spiritual connections to Country. We recognise their valuable contributions to Australian and global society.



A Guidance
Through Time by
Casey Coolwell
and Kyra
Mancktelow ©
The University of
Queensland

About the artwork

Quandamooka artists Casey Coolwell and Kyra Mancktelow have produced an artwork that recognises the three major campuses, while also championing the creation of a strong sense of belonging and truth-telling about Aboriginal and Torres Strait Islander histories, and ongoing connections with Country, knowledges, culture and kin. Although created as a single artwork, the piece can be read in three sections, starting with the blue/greys of the Herston campus, the purple of St Lucia and the orange/golds of Gatton.

The graphic elements overlaying the coloured background symbolise the five UQ values:

- The Brisbane River and its patterns represent our Pursuit of excellence. Within the River are tools used by Aboriginal people to teach, gather, hunt, and protect.
- Creativity and independent thinking is depicted through the spirit guardian, Jarjum (Child in Yugambeh language), and the kangaroo
- The jacaranda tree, bora ring, animal prints, footprints and stars collectively represent honesty and accountability, mutual respect and diversity and supporting our people.

Learn more about [The University of Queensland's Reconciliation Action Plan](#).

1.

INTRODUCTION

In this section, we define what we mean by **innovation**. This may seem strange – I’m sure you have all used the word. But in my experience, when I talk with people in technology-related fields saying they “do innovation”, they are often just using the latest software tools, techniques, or frameworks. This is, most definitely, not being innovative (although in doing innovation, we may sometimes use the latest tools).

What is innovation?

Definition — Innovation

Innovation is **creating new value for someone**.

What does it mean to create value?

This can be in the form of:

- New solutions to real-world problems.
- Old solutions given to a new group of people.
- Deriving new methods to solve new or existing problems.

The three important terms are: **new**, **value** and **someone**.

If a solution does not help people, they will not use it. It doesn’t matter how sophisticated or cutting-edge the solution is. People will only pay for and use products/services that help them.

“Quality in a product or service is not what the supplier puts in. It is what the customer gets out and is willing to pay for. A product is not quality because it is hard to make and costs a lot of money, as manufacturers typically believe. This is incompetence. Customers pay only for what is of use to them and gives them value. Nothing else constitutes quality.” – Peter Drucker, p. 129 [\[7\]](#).

The term **innovation** can also refer to the **process** for creating value. This is the focus of this course: the processes that people follow to create value for someone.

The Innovation Process

In this course, we will follow an innovation process called **Lean startup**, based on the book by Steve Blank [\[1\]](#), and with inspiration throughout the course from other innovation experts such as Tina Seelig [\[2\]](#) and Alexander Osterwalder & Yves Pigneur [\[3\]](#).

Why do we need to follow a process? Why can't we just come up with ideas? Ideas are simple. Implementation is hard. What happens if we take the time to implement our idea, and it turns out that *nobody wants it*? What a waste of time! (And perhaps a waste of money too!).

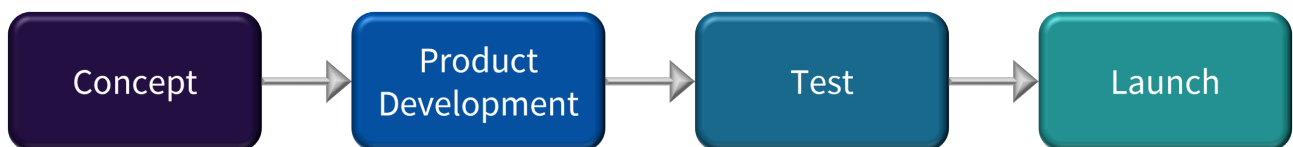
Innovation processes help us to test our ideas, abandon or refine them if they don't create value, and change them to something more valuable – but doing this **early** before we spend a lot of time and resources.

Importantly, just because **we think** our idea will bring people value does not mean it will. In fact, most ideas that people believe are great never do.

“Ideas are like babies because everyone thinks theirs is cute, therefore be objective when judging your own ideas.” – Tina Seelig [\[5\]](#)

The Lean Startup approach is a collection of methods for helping us objectively judge our ideas.

(Software) product development



The product development process. Adapted from “The four steps to the epiphany: successful strategies for products that win” by Steve Blank (2020) [\[1\]](#).

The figure above shows the standard product development process:

1. come up with a concept for a product that we think is good
2. develop the product
3. test and evaluate it

4. launch it.

Note

Throughout these notes, we use the term “product” to mean “products and services”, as a service is simply an intangible product.

This can work if the problem is well understood and the solution is clear. However, this is not often the case.

Often, the focus on product development becomes the launch date. Marketing, execution, and hiring are all predicated on a business plan that is making incorrect assumptions.

If we follow this product development process, **we won’t know we are wrong until we launch, and by then we are out of business.**

Product and business failure

According to the latest public [Standish Group report](#), about 20% of all software projects fail. About **35-40%** of projects are **successful**, which means that they are on time, on budget, and deliver a satisfactory result. The remaining **40-45%** are **challenged**, which means they run over time, over budget, or don’t deliver something satisfactory. Larger projects fail more often. These are high failure rates!

Results are worse for startup companies. Over [90% of startups fail](#), with 50% failing within 5 years. CB Insights data [\[8\]](#) indicates that startups fail mostly due to lack of cash flow and poor market fit:

Failure reason	Percentage
Cash flow	38%
No market fit	35%
Competition	20%
Business model	19%
Regulatory challenges	18%
Pricing issues	15%
Team	14%
Timing	10%

Interestingly, most of the issues above are NOT product development issues. Market fit accounts for one third; and one can argue that cash flow failures would be lower if teams released early product that fit the market or show their investors that they had a great market fit for their product. This latter argument

about “investors” holds where the “investors” are people in our organisation who have the ability to provide funding to our projects.

If products fail due to a lack of customers rather than product development failures, then why do we have:

- a process to manage product development; but
- no process to manage customer development.

There is an inexpensive and simple fix:

Focus on customers and markets from day one.

From (software) product development to (software) innovation

To use software as an example, innovation differs from product development in four different ways. It takes us:

1. **From a client to a market segment:** We focus much less on specific clients for whom we build the product. Instead, we have to identify a **market** – a segment of the population who would be paying customers for our idea. If we are innovating within an organisation, we still need to identify who our “market” is to justify spending budget on it.
2. **From a system to a value proposition:** We focus much less on a particular system or solution; instead, we focus on identifying what will bring value to our market.
3. **From requirements specification to a minimal viable product:** We focus much less on identifying the requirements for a product, to identifying the smallest amount of functionality that we believe can bring value, so we can test our ideas early.
4. **From development to experimentation:** We focus much less on the life cycle of developing a software solution (at least initially), and much more on **experimenting** with our ideas by “**getting out of the building**” and talking to people in our market segment, gaining feedback on our ideas.

A customer development process: the four steps

“What you will get wrong is that you will not pay enough attention to your users.

You will make up some idea in your own head that you will call your “vision”, and you will spend a lot of time thinking about your vision. In a cafe. By yourself. And build some elaborate thing without going and talking to users, because that’s doing sales, which is a pain in the ass, and they might say no.

You will not ship fast enough because you’re embarrassed to ship something unfinished, and you don’t want to face the likely feedback that you will get from shipping. You will shrink from contact with the real world, contact with your users. That’s the mistake you will make.” – Paul Graham (spoken at 40m 39s) [\[6\]](#)



From [Talking to humans](#). Credit to Tom Fishburne

Text version

Person A says: 'If I were our teenage girl target, I would love our new product'.

Person B says: 'Have you actually talked to any to make sure?'

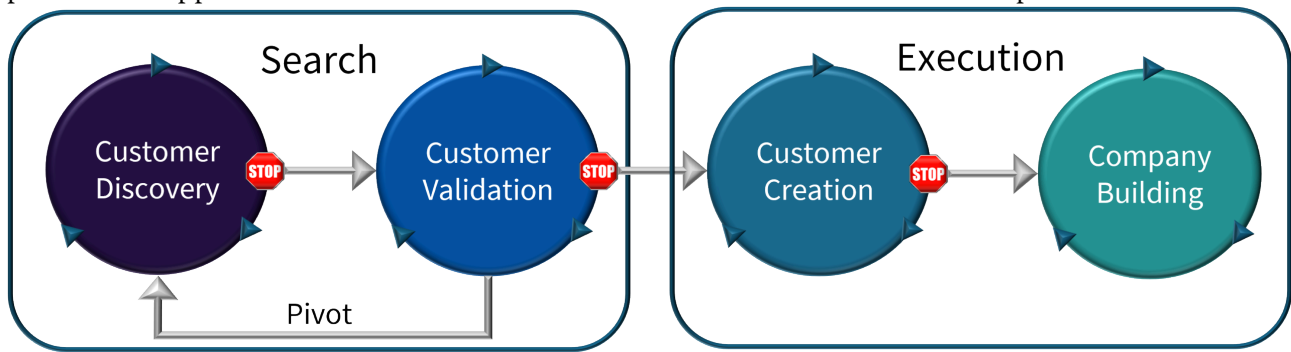
Person C says: 'What? And leave this room?'



The interactive version of this H5P content is available at:

<https://uq.pressbooks.pub/introduction-software-innovation/?p=577#h5p-1>

The customer development process, shown below, is not a replacement for the product development process – it supplements it. While the focus is on customers, we still need to deliver products.



The customer development process. Adapted from “The four steps to the epiphany: successful strategies for products that win” by Steve Blank (2020). [\[1\]](#)

Customer development has four steps:

1. **Customer discovery** finds out who our potential customers are and what their problems are. This captures the innovators’ vision and turns it into a set of testable hypotheses. Then it develops a plan to test those hypotheses and turn them into insights.
2. **Customer validation** evaluates whether we have actually found a group of customers and a product that they would pay for. This tests whether the resulting innovation is repeatable and scalable. If not, return to customer discovery.
3. **Customer creation** is the start of execution. It builds end-user demand and drives it into the sales channel to scale the business.
4. **Company building** transitions the organisation from a innovation unit to a company focused on executing a validated business model.

Note that, as outlined in the [customer development process model](#), each step is itself iterative — developing customers is not a one-off or even a linear process. Further, going back from customer validation to customer discovery is encouraged when we find that we have not understood our customers properly.

At the end of the process, we want to be able to answer five questions:

- Have we identified a problem that a set of customers wants solved?
- Have we identified a product that solves these problems?
- Do we have a set of customers that will pay for this product?
- Do we have a profitable and viable business model to produce this product?
- Do we know enough about our product and customers to go out and sell this?

Focus of this book

In these notes, we focus mostly on the first step: customer discovery, with some parts of customer

validation. These are the parts that are relevant to **everyone** working in technology, including people working in existing organisations and people who have no desire to start their own startup.

Customer discovery is crucial to any innovation project, whether to start up a new company, innovating new products in existing companies, or internal projects at organisations.

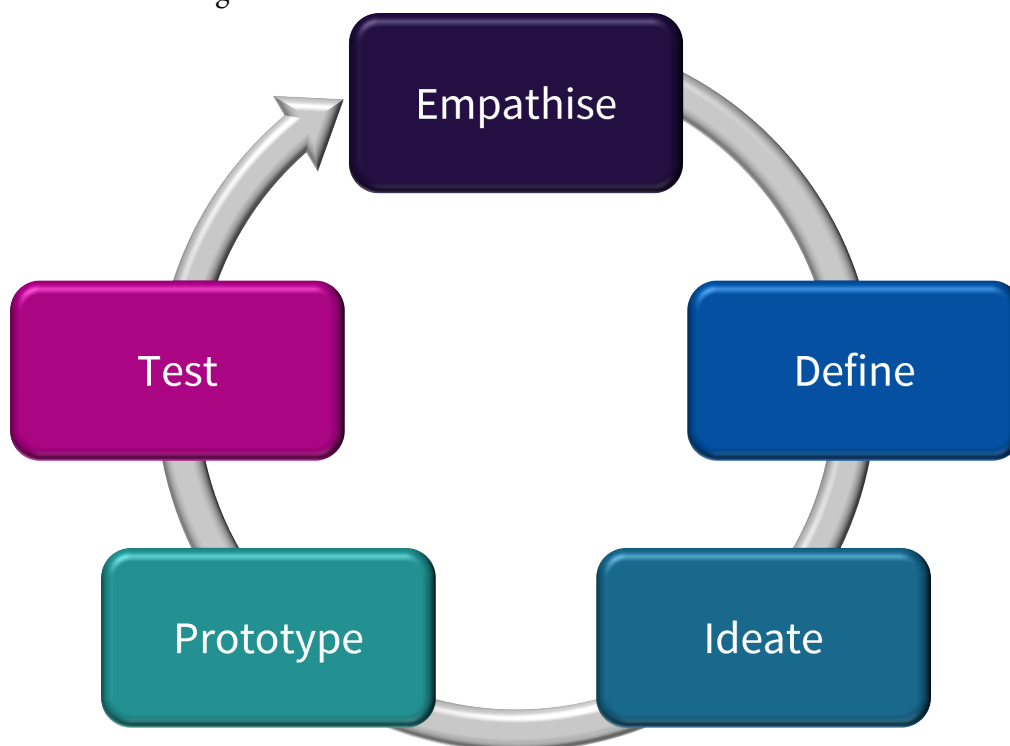
However, the ideas and processes around testing, learning, and failing early that are crucial in customer discovery are similar to the remaining steps — the activities that we use them on vary in other phases.

Design thinking

Note

There are no facts inside our building that can help us understand our customers, so we need to **get outside of the building**.

To execute each of the steps in the customer development process, we can adopt the **design thinking process**, as outlined in the figure below.



The design thinking process

The design thinking process is an iterative process broken into five steps:

1. **Empathise** with people who are our prospective users/customers. To **empathise** with someone is not to **sympathise** with someone; or just to **care** about someone. Empathy is an understanding that we develop about another person, and empathy is built by deeply **listening** to someone to

understand their thoughts and the reasons for their behaviour. In innovation, empathising is the process of understanding people’s problems, why they consider them problems, and how these issues impact their lives. Empathising helps use gain insights into the needs and desires of people, which help us develop meaningful solutions. Watch [this killer 5-minute video of what empathy really means \(Vimeo, 4m 15s\)](#) from my colleague Professor Tim Kastle, Director of the Andrew N. Liveris Academy for Innovation and Leadership at the University of Queensland:



One or more interactive elements has been excluded from this version of the text. You can view them online here: <https://uq.pressbooks.pub/introduction-software-innovation/?p=577#oembed-1>

2. **Define** the problems that our demographic has, based on the understanding we gained from empathising. Be sure to pay attention to the people and what they said; and importantly, this is not about coming up with solutions. We will discuss later how to make these definitions specific and actionable.
3. **Ideate** to come up with concrete ideas that address the problem. “Ideation” is a term that sounds like it comes from a charlatan’s toolbox because a lot of people use it meaninglessly, but it just means to generate and communicate ideas. We can use a range of creativity techniques, such as brainstorming, mind mapping, collaborative workshops, and co-design. The important part of this step is to first generate as many ideas as possible without critiquing them, and then after this, select the set of most promising ideas. A research collaborator of mine, Professor Marek Kowalkiewicz, Chair of Digital Economy at Queensland University of Technology points out that “creativity” is not a magic process — all of us can be and are creative. Marek talks about ideating by **crafting** ideas using simple techniques. Check out [Marek’s awesome TEDx talk “Creativity not required: how great minds craft ideas” \(YouTube, 11m 43s\)](#) on this topic.



One or more interactive elements has been excluded from this version of the text. You can view them online here: <https://uq.pressbooks.pub/introduction-software-innovation/?p=577#oembed-2>

4. **Prototype** the best ideas into simple, inexpensive prototypes. A prototype is a preliminary version of a product that is used to gain feedback on ideas. Even in software products, a prototype need not be an actual piece of software. Simple prototypes such as storyboards, or even simply statements of what the problem is, can be tested with our customers.
5. **Test** our prototype with potential customers to get feedback on whether we have understood the problem and generated useful ideas. It is important at this step that we **do not try to sell** our idea or

prototype — we simply want to get feedback.

Note

The design thinking process is **iterative**. When we test prototypes with our demographic, we can use this change to learn from them by empathising. From a feedback session, we should gain new insights and a better understanding of the demographic we are working with, and their problems.

Lean start-up vs. design thinking vs. product development

Factor	Lean Startup	Design Thinking	Product Development
Philosophy	Fail early, learn	User-centred	Product quality
Goals	Finding a market and business model	Solving user problems	Delivering a quality product
Approach	Hypothesis-driven	Empathy, ideation, prototyping	Stage-gate process, often iterative for software
Assumptions	Initially, we know nothing	We know our customers/users	We know what product to build
Iteration	Rapid, continuous	Regular, feedback-driven prototyping	Often limited, but can be agile
Time to Market	Short, focuses on quick feedback	Variable, depends on iteration cycles	Longer, detailed planning and development phases
Techniques	Hypothesis testing, MVP	Empathy maps, journey maps, brainstorming, prototyping	Requirements analysis, detailed design, testing
Outcome	Product with validated market fit	Innovative solutions that address user needs	Fully developed, feature-rich product

The above table summarises the differences between the Lean startup approach, design thinking, and product development. Each process has a different philosophy, goals, approaches, techniques, etc.

However, we needn't think of these as competing processes – they are **complementary**. We use design thinking and related techniques throughout the innovation process.

Once our product idea is well tested and we have a design that is both a good product-market fit and useful for users, we use product development processes to produce a quality product.

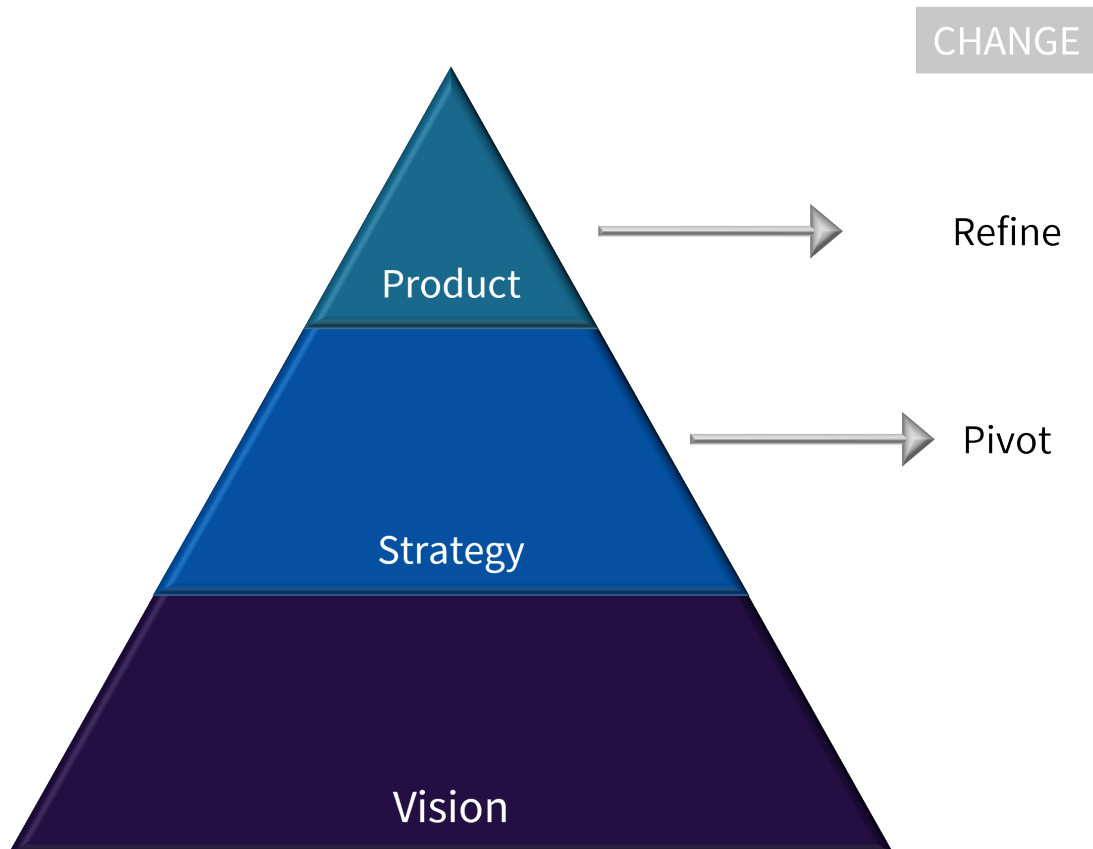
Pivot vs. refine

What happens if we test an idea and it turns out we are wrong about our customers, their problems, and/or our solution?

Well, we need to come up with new ideas!

But first, we need to be careful not to simply throw away everything we have learnt.

The following figure shows how Eric Ries [4] conceptualises product innovation. The **vision** is what he calls the “true north” – this is how we believe the world could be if we are successful. To achieve our vision, we need a **strategy**, which includes the understanding of our customers, partners, and competitors, the ideas we have about our customers’ problems, and the road map for our product. The **product** is the result of implementing the strategy.



Pivoting vs. refining when we learn our ideas won’t work. Adapted from “The lean startup: How today’s entrepreneurs use continuous innovation to create radically successful businesses” by Eric Ries (2011). [4]

When we test our ideas and find out they are wrong, **we should treat this as an opportunity to learn, not as a complete failure.**

Running example: Medical alert systems for older people living independently

Throughout these notes, we’ll use the following running example, which is a project that I worked on with some colleagues at Swinburne University during my time at The University of Melbourne.



An emergency alarm for older people living at home independently. Image: “[Medical alarm device](#)” by [Florian Fuchs](#), shared under a [Creative Commons Attribution 3.0 Unported licence](#).

Background

Older people living independently can occasionally have emergency medical issues that require medical support, but they are unable to get to a phone. For example, they fall and are unable to get up. Medical alert systems, such as the example in the image above, can be used to mitigate these issues. A medical alert has two components:

1. a wireless emergency alarm, such as a wrist band (as shown in [the figure above](#)) or a “necklace” worn around the neck that has a button that, when pressed, raises an alarm with a specialised care company, who calls for an ambulance; and
2. a base station with a *wellness check*. This often contains a red emergency button that connects the person to the care company (same as pressing the emergency alarm), and also a green button that they are asked to press each day before a fixed time. If they fail to press the green button before that time, the care company calls them to check that they are OK.

The main aims of this are: (a) keep the person safe; and (b) give peace of mind to their family that if anything happens, they will know quickly.

Challenge

The challenge is that *most older people hate these medical alert systems*. They are typically forced upon them by their family, who give them an ultimatum of having an alarm or going into a care home. Their family clearly care about them and want them to be safe, but older people feel that they are under surveillance. They also find that wearing emergency alarms around the wrist or neck is a signal that they are old, frail, and no longer independent. In our study, we found that most people hate the wireless transmitter they wear around their neck, calling it a “cowbell” (allowed people to keep tabs on them), and they often didn’t even wear it unless their family were coming to visit (the only time it was not useful). Many older people also disliked the concept of the green button. If they forgot to press it (which must be so easy to forget as it is not meaningful to them), they would receive a call and feel that they were forgetful and wasting people’s time. In short: they felt a lack of control/agency in their lives. The company providing the service didn’t view this as a waste of time; and in fact, the people calling them were trained specifically to take the time to talk to them and not rush them off the phone — quite unusual for most call centres. Some older people didn’t mind the system. We saw that the people who *opted in* (that is, they proposed the idea of getting a medical alarm) were much more accepting than those who were given an ultimatum by their family.

Our thoughts

When one of our team gained second-hand experience in these systems, having purchased one for his mother, he discussed his and his mother’s experience and noted that there must be an opportunity for better-designed medical alert systems that can achieve the goals of: (a) giving peace of mind to family members who are worried about older relatives living on their own; and (b) giving control/autonomy back to people living independently.

But what would this be?

Without learning about the experience, needs, and desires of older people who live independently and their families, any design we came up with was speculative. We had to **get out of the building and talk to people**.

We spoke with several groups of people:

1. Older people who lived on their own and had an existing medical alert system.
2. Older people who lived on their own and did not have an existing medical alert system.
3. Families of older people who lived on their own.
4. Two existing medical alert companies, including their executive staff and (quite informally) some call centre staff.

Throughout these notes, we’ll use our journey to make the innovation concepts discussed more concrete.

Chapter summary

The key takeaways for this chapter are:

1. Innovation is about **creating value for someone**. It is not about using the latest technology, wearing turtleneck jumpers at product launches, or just coming up with ideas and implementing them.
2. Most startups, businesses, and products fail because people **do not understand their market**.
3. Successful innovators **get out of the building** to **understand their customers and their customers' problems**.
4. Successful innovation is about **generating and testing ideas early** so we **fail early** if our ideas are wrong.
5. When our ideas fail, we can **use this as an opportunity to learn**, and then either **refine our product** or **pivot our strategy**.

References

- [1] (1,2,3) Blank, Steve. *The four steps to the epiphany: successful strategies for products that win*. John Wiley & Sons, 2020.
- [2] Seelig, Tina. *inGenius: A crash course on creativity*. Hay House Inc, 2012.
- [3] Osterwalder, Alexander, and Yves Pigneur. *Business model generation: A handbook for visionaries, game changers, and challengers*. Vol. 1. John Wiley & Sons, 2010. <https://www.strategyzer.com/>
- [4] (1,2) Ries, Eric. *The lean startup: How today's entrepreneurs use continuous innovation to create radically successful businesses*. Crown Currency, 2011.
- [5] Seelig, Tina. *What I wish I knew when I was 20: A crash course on making your place in the world*. HarperOne, 2009.
- [6] Y Combinator. [A conversation with Paul Graham – Moderated by Geoff Ralston](https://www.youtube.com/watch?v=4WO5kJChg3). 2018. <https://www.youtube.com/watch?v=4WO5kJChg3>
- [7] Drucker, Peter F. *The Essential Drucker*. Routledge, 2007. <https://doi.org/10.4324/9780429347979>
- [8] CB Insights. *The top 12 reasons startups fail*. 2021. <https://www.cbinsights.com/research/report/startup-failure-reasons-top/>

PART I

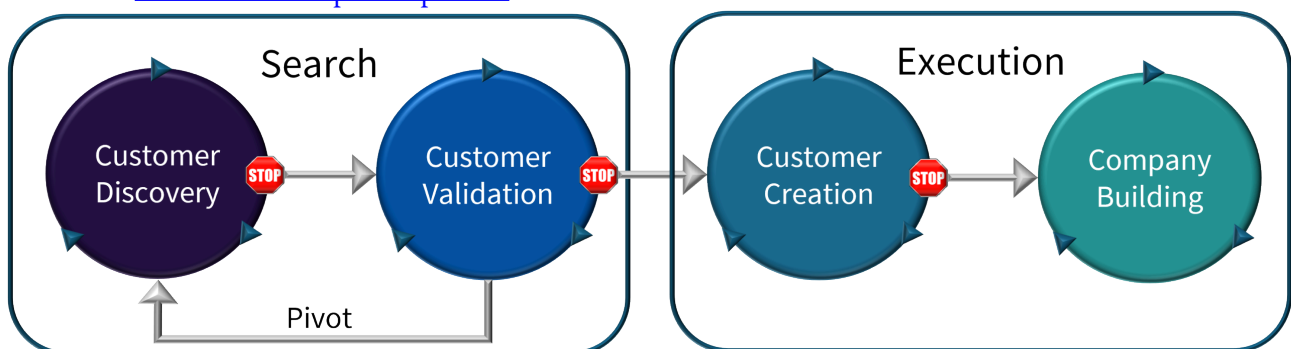
CUSTOMER RESEARCH

2.

CUSTOMER DISCOVERY

“If I had an hour to solve a problem and my life depended on the solution, I would spend the first fifty-five minutes determining the proper question to ask, for once I know the proper question, I could solve the problem in less than five minutes.” – Albert Einstein

Recall the [customer development process](#):



The customer development process from chapter 1. Adapted from “The four steps to the epiphany: successful strategies for products that win” by Steve Blank (2020) [\[1\]](#).

Innovation takes us from thinking about clients and users to thinking about **customers**.

For this, we need to **discover who our customers are** and **what they need**. This is step 1 in the customer development process.

Customer discovery doesn’t mean that we literally need to find the exact people who will buy or use our product. It means something different.

But first, let’s define what customer discovery does **NOT** mean:

- that we need to understand the needs and wants of everybody, or even all of our customers.
- that we need to find every problem they have.
- requirements engineering
- that we need to collect feature lists from customers.

- that we are running lots of focus groups.
- defining a product that we think is a good idea and trying to sell it.

Definition — Customer discovery

Customer discovery is the process of turning an initial idea or vision about a market or customers into a set of **facts**. The process tests whether the business model for a company solves real customer problems and whether the product features fit those customer problems.

When we think about building a product, we have an initial guess, called **hypotheses**, about people's problems and how we can solve them. This is best considered a **vision** rather than a specific product. We fail when we assume that our guesses are facts.

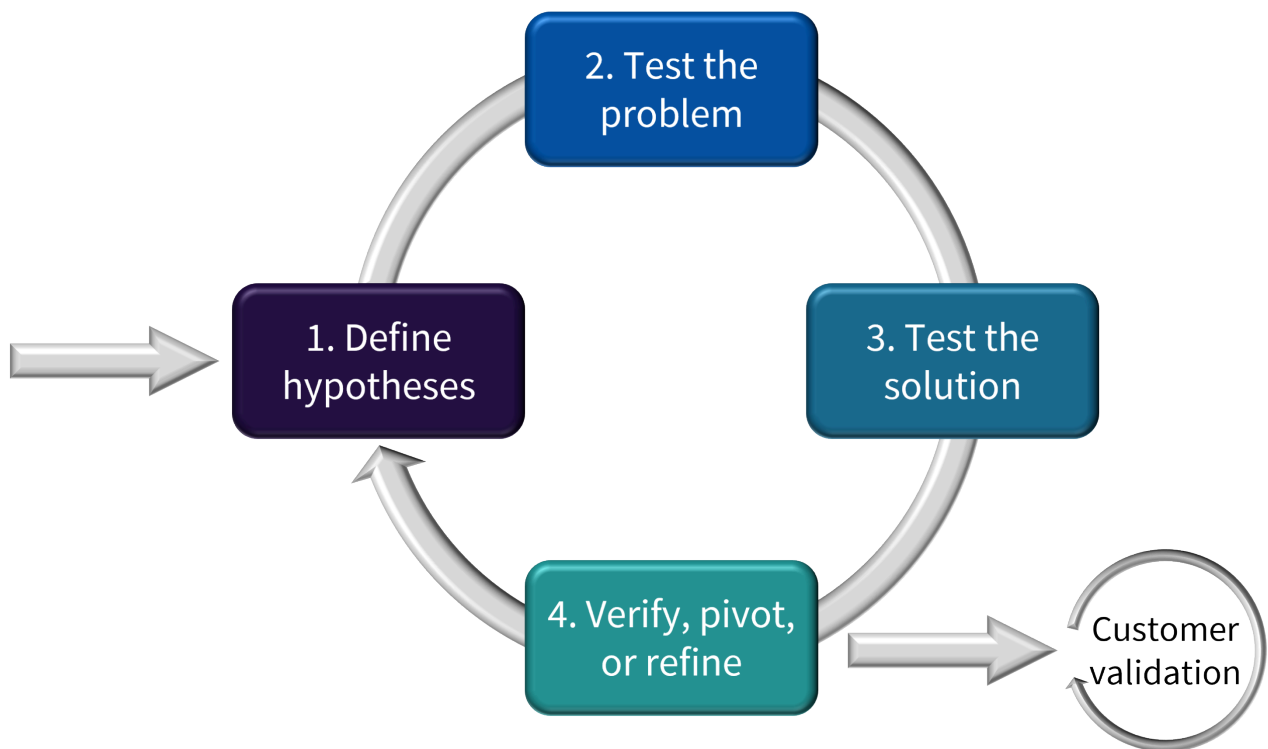
Note

As a side note, I dislike the word “facts” in this context, because even once we go out and find customers to confirm our hypotheses, they are not certain. However, we'll use the term “facts” to be consistent with the startup community.

The goal of customer discovery is to find out who our customers are and whether the problem we are solving is important to them. It emphasises discovery and iteration.

The [customer discovery process](#) consists of four phases:

1. **Define hypotheses:** State our guesses and assumptions as verifiable hypotheses (more on this later) and adding these on a business model canvas (also [more on this later](#)).
2. **Test the problem:** Get out and speak to our customers to see if we have understood their problem. If not, return to step 1.
3. **Create and test the solution:** Get out and speak to our customers to see if our solution solves their problem and find out whether they would be willing to use it and pay for it.
4. **Verify, pivot, or refine:** For those hypotheses that hold, turn them into **facts**. For those that don't, [pivot or refine](#).



The customer discovery process. Adapted from “The four steps to the epiphany: successful strategies for products that win” by Steve Blank (2020) [\[1\]](#).

Typically, we want to be able to move through this loop as quickly as possible, so that if we get some ideas wrong, we **fail early**.

Define our hypotheses

To define our hypotheses, we first have to **sketch our vision**.

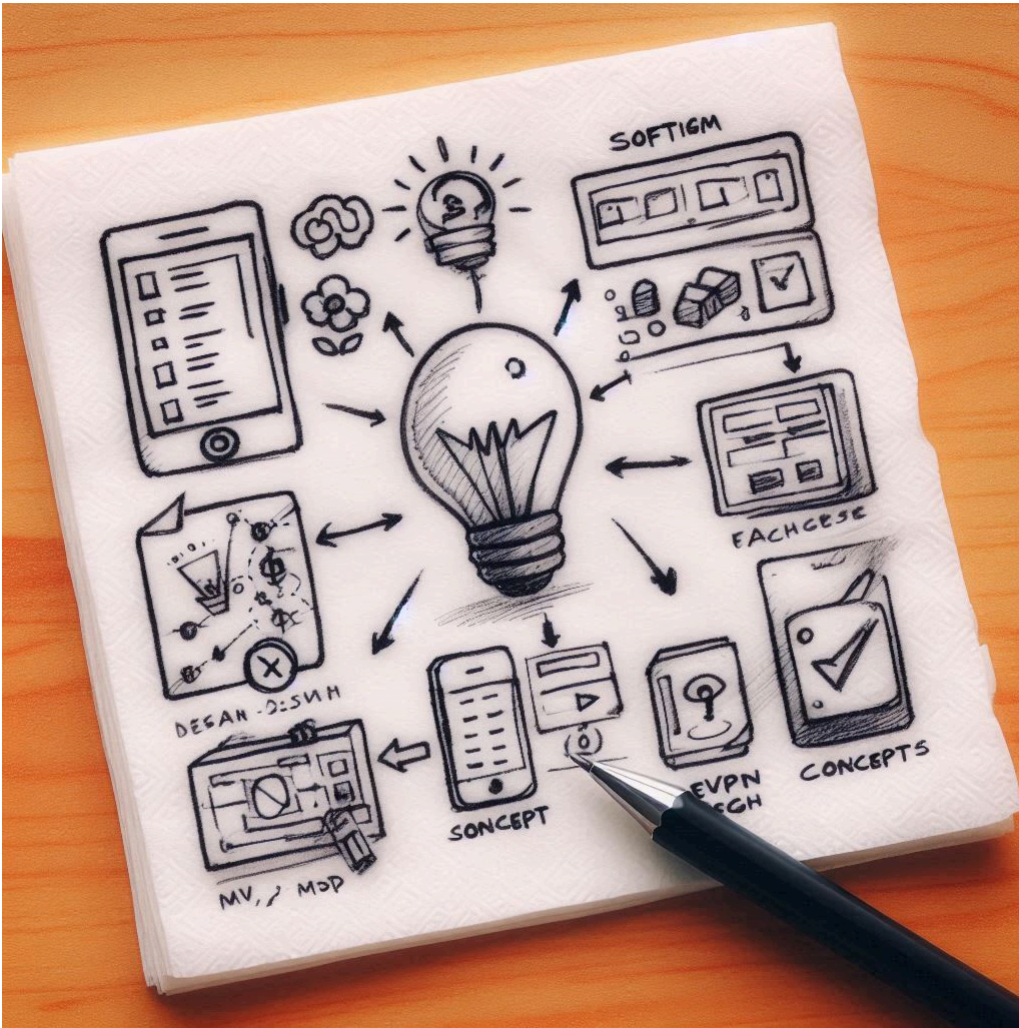
This means writing out answers to the following questions:

1. What are our customers’ top problems?
2. Does our idea help to solve these problems?
3. What would a day in the life of a customer look like before and after our product?

At this point, the answers to these are **guesses** or **assumptions**. We typically do not know if the answers are true. It may be “obvious” that they are, but many projects fail because team members don’t test “obvious” assumptions that turn out to be false.

This version really needs to just be a sketch. At this point, we shouldn’t be defining detailed solutions to the problem, as we may be wasting valuable time designing a solution to a problem that doesn’t exist.

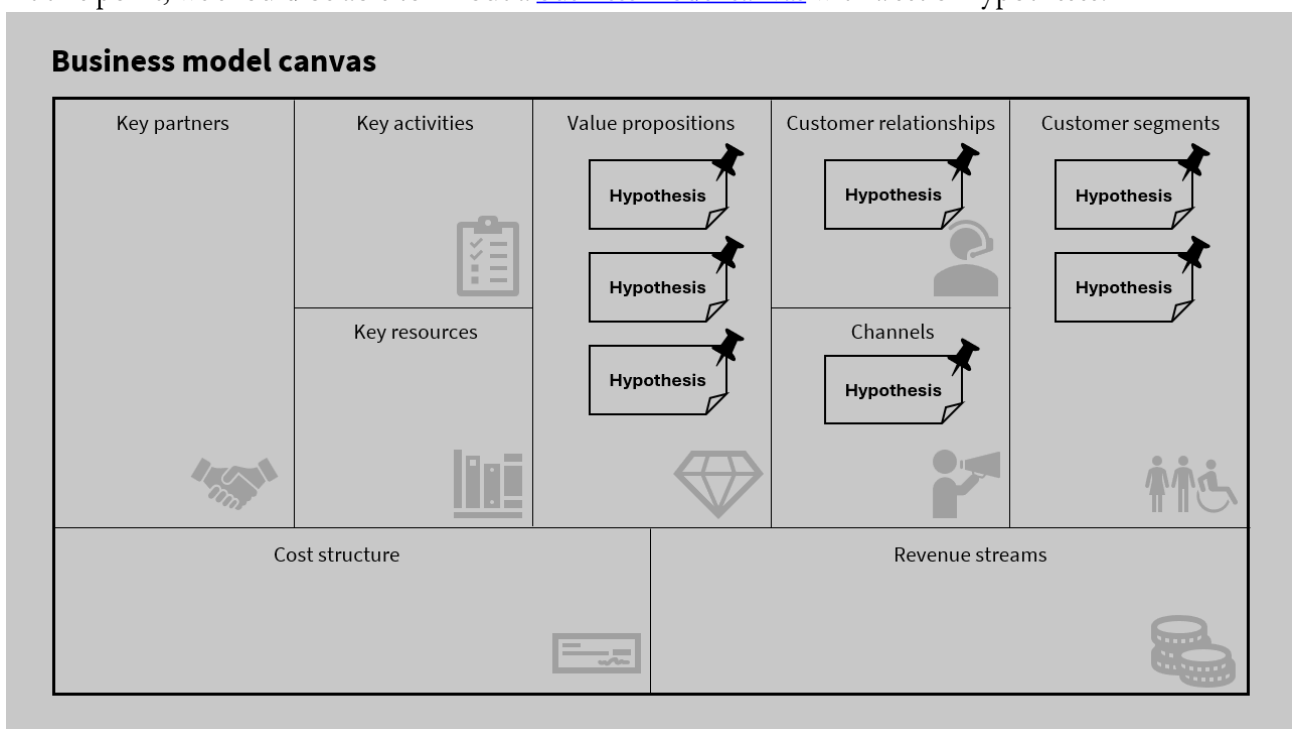
like this...



not like this...



At this point, we should be able to fill out a [business model canvas](#) with a set of hypotheses:



Business model canvas. Adapted from “[Business model canvas](#)” by [Strategyzer](#), modified under a [Creative Commons Attribution ShareAlike 3.0 Unported licence](#).

Of course, it may be that we already know the answers to some of these questions, either because there are other products that have demonstrated this – products from other organisations or even our own – or because someone has already collected the data for us. For example, Sony would have had an unverified hypothesis in the 1980s that people would like to have a portable music system with headphones to listen to music as they walked around. Once they sold millions of these, this hypothesis was confirmed for *other* organisations as well, who built on the idea.

However, there must be **some** questions we don’t have verified answers to. If we did already, it is because someone has already validated these by building the product, and we are therefore not really innovating at all. There is presumably something that we plan to do differently to competitors or to improve on earlier versions of our product, and those are based on customer problems that are usually not verified (yet).

Note – **Customer vs. user**

We need to think broadly when we say “customer”. Importantly, we need to consider both our users and our customers — these two segments may be different. The **users** of a product are the people who will use it as part of their job, their life, etc. The **customers** are the people who pay for it.

Consider the example of a person working in sales support, who would use sale support software, vs. the people who approve the purchase of this software, which may include the chief technology officer (CTO) or IT manager, the chief executive officer (CEO), or the chief financial officer (CFO); and maybe all of the above. We may need to talk to both the end users and the paying customers as part of customer discovery. We need to verify whether the end users really feel the pain of the problem we have identified; and also whether the CTO would be willing to pay for our solution.

We will use the term “customers” to refer to both end users and the paying customers.

Example – Medical alert system initial hypotheses

In our [medical alert system](#) example, we had some initial hypotheses that helped to define our vision:

1. Older people want to be safe.
2. Families of older people want to know, on a daily basis, that their older family member is safe.
3. Older people want to be able to initiate a call for help and cancel any call for help.
4. Older people want to maintain their independence/autonomy.
5. Older people want to be cared for.
6. Older people do not want to have to press the green button each day to let the care company know that they are alive.
7. Older people do not to wear a visible (and ugly!) emergency alarm around their neck or wrist.

These are quite high-level for a product vision. At this stage, we had many ideas for potentially products, but these were not our focus just yet – we first wanted to go out and test our hypothesis, that the issues our team member and his mother were having were widespread.

Test the Problem

Next, we need to get some data to test whether our customers really do have the problem we think they do.

To test these, we use experimentation, which we discuss further in [Experimentation: Testing and validation](#).

Effectively though, we gather data that can be used to confirm whether our guesses/assumptions are correct or not.

Typically, this involves going out and speaking to potential customers about their problems. However, we may also be able to gather some data via other means, such as existing data sets. However, it is highly unusual to be able to verify more than a handful of hypotheses without talking to potential customers.

If the data we gather **refutes** our guesses, we need to **re-visit** our hypotheses.

If the data we gather **supports** our guesses, move to the next step.

Test the Solution

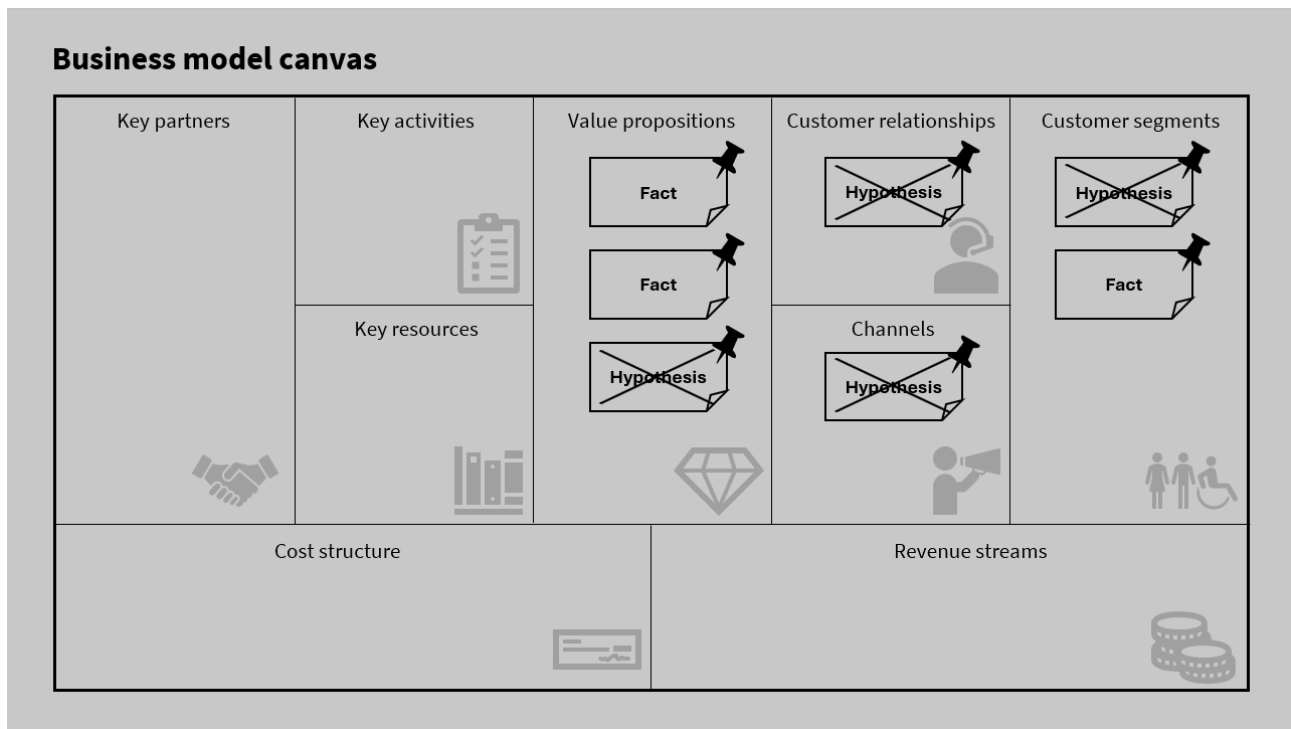
Next, we need to get some data to evaluate whether our solution really does solve the problem.

We test the hypotheses using the same process: going out to interact with customers.

What we take to these meetings depends on how much information we already have, how solid our idea is, and how easy it is to generate some material (e.g. a prototype) to gain feedback.

Verify, Pivot, or Refine

At this point in the process, we should have a set of hypotheses that are either supported (and become **facts**) or refuted:



A business model canvas showing some refuted hypotheses and some facts. Adapted from “[Business model canvas](#)” by [Strategyzer](#), modified under a [Creative Commons Attribution ShareAlike 3.0 Unported licence](#).

Similar to our initial questions, we end the customer discovery process when we have facts that answer ‘yes’ the following questions:

1. Do we know our customers’ top problems?
2. Do they recognise that they have this problem?
3. If there was a solution, would they buy it?
4. Can we build a solution at a price they would be willing to pay?

If we answer ‘no’ to any of these, we need to [pivot or refine](#):

- Pivot: Change our strategy and repeat the cycle of customer discovery again.
- Refine: Change our product and repeat the cycle of customer discovery again.

Note

Be careful to distinguish between an individual customer not recognising that they have a problem (which would, in fact, mean they are probably not really a customer), and an entire segment of the market not recognising that they have a problem. If an individual customer does not see a problem, we move on to the next person (and maybe return to this customer later once early adopters are using our solution). If an entire market segment does not see a problem, we need to refine or pivot: either find a new market segment who does have this problem, or work towards a different problem.

Once we have a set of facts describing our customers and solutions to their problems that are verified, we can move on to the customer validation step.

What is a verifiable hypothesis? We discuss this in the chapter on [Experimentation: Testing and validation](#).

Example – Medical alert system refined hypotheses

In our medical alert example, after some interviews with various stakeholders, we were able to verify:

1. Older people want to be safe.
2. Families of older people want to know, on a daily basis, that their older family member was safe.
3. Older people want to be able to initiate a call for help and cancel any call for help.
4. Older people want to maintain their independence/autonomy.
5. Older people do not want to have to press the green button each day to let the care company know that they are alive.

In the interview, hypothesis 5 was refuted:

5. Older people want to be cared for.

We found that older people wanted to be *cared about*, and otherwise rejected the idea that family and service providers should care for their safety, which is somewhat in conflict with the goal of being independent. Thus, we replace this with:

5. Older people want to be cared about.

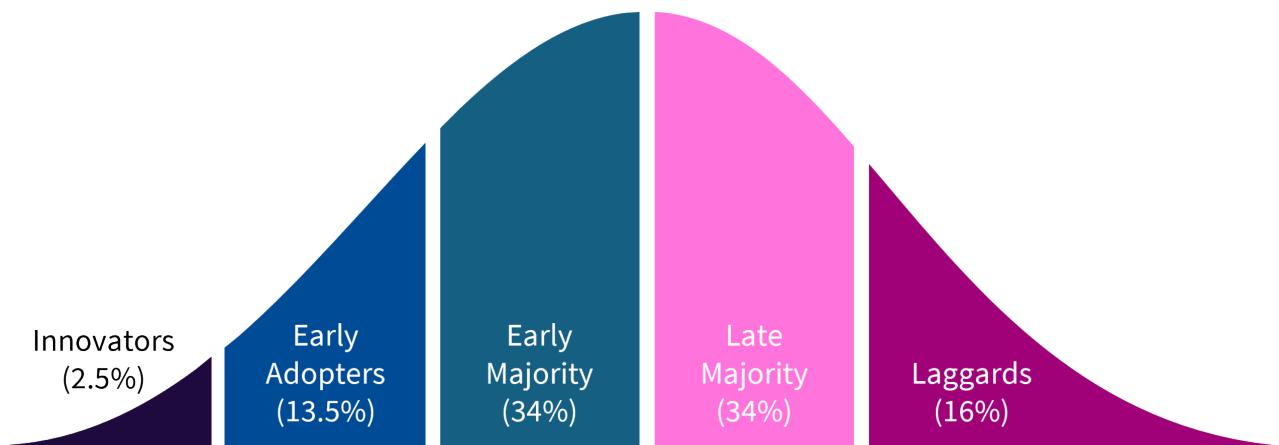
We also partially refuted hypothesis 7:

7. Older people do not want to wear an emergency alarm around their neck or wrist.

We discovered that:

8. Some older people wouldn't mind wearing a wireless transmitter if it was not visible to others.
9. Older people wanted to be *in touch* with their family members, rather than a service provider.
10. Older people would prefer a medical alert that allowed them to be mobile (the alarms were attached to a phone line and the wireless transmitters have a short range).

Customer adoption patterns



Customer adoption patterns. Adapted from “Crossing the chasm: Marketing and selling high-tech products to mainstream customers” by Geoffrey A. Moore (2014) [2].

The [model for customer adoption patterns](#) above shows the rough pattern by which customers tend to adopt new products. We have five broad types of customer in this pattern:

1. The innovators, who create new products themselves and make up a small amount of the market.
2. The early adopters, who typically adopt new solutions before others.
3. The early majority, who adopt once they see some people using a solution successfully.
4. The late majority, who adopt once they see many other people using a solution successfully.
5. The laggards, who typically don’t adopt or adopt very late.

Definition — Early adopter

Early adopters are people who typically adopt new solutions before others. They understand that they have a problem, want a solution, usually have an interim solution, and have the funds to purchase a solution. Often, they like to be on the cutting edge.

It’s important in innovation to initially **focus on early adopters**. Our goal in customer discovery is not to understand the needs of all of our customers. Instead, we want to develop our product incrementally: at first, just for the few early adopters, and once we have some paying customers, broaden to the early majority.

What makes early adopters so crucial? There are a few things:

1. They have a problem.
2. They know they have that problem.
3. They typically have an interim solution for that problem, or are actively searching for a solution.
4. They are willing and able to pay for a solution.
5. They tend to adopt things before others; while the rest of the market waits to see.
6. They tend to spread the word about your product if it works for them.

So, early adopters are “easier” to sell to, and help us to establish ourselves in the market. As such, focus on their needs first, and iteratively build a product that reaches the rest of the market.

In our medical alert project, many of our initial interviews were with people who were existing customers, and many of them (both older people and their families) told us stories to indicate that they thought there was sufficient room for improvement in these systems.

Customer Development Heuristics

There are a few good heuristics to go by when doing customer development:

1. There are no facts inside your building, so get outside.
2. Develop for the few, not the many.
3. Early adopters make your product...
4. ... And they know more than you about the problem.
5. Focus groups are for big companies, not startups.
6. Surveys don't give insight compared to interviews — follow-up questions are difficult.
7. The goal for release 1 is the minimum feature set for early adopters.

Chapter summary

The key takeaways for this chapter are:

1. Customer discovery is a process for setting out a vision based on some hypotheses, and turning (some of) those **hypotheses into facts**.
2. It requires us to **get out of the building** and work with potential customer segments.
3. It is an **iterative process**.
4. At the end of each iteration, we decide whether we can continue to customer validation, or whether the need to *pivot or refine*.
5. **Early adopters** will be our first customers, so focusing on them can be valuable.

References

[1] Blank, Steve. *The four steps to the epiphany: successful strategies for products that win*. John Wiley & Sons, 2020.

[2] Moore, Geoffrey A. *Crossing the chasm: Marketing and selling disruptive products to mainstream customers* (3rd ed.). Harper Business, 2014.

3.

INTERVIEWS

Interviews are a key way to gather data about your customers and their problems, as well as data about how well your solutions solve their problems.

Interviews allow us to check our assumptions early.

Ground rules for interviewing customers

Ground rule	Explanation
Adopt a beginner's mindset	<p>Listen with a "fresh pair of ears".</p> <p>Ask for clarification instead of assuming what the interviewee means.</p> <p>Explore things that you did not expect.</p>
Do interviews in person	<p>If possible! You can see how people react.</p> <p>Video conference is next best.</p>
One person at a time	<p>Focus groups tend to lead to group think</p> <p>You will struggle to get deep engagement for more than one person.</p>
Listen more than you talk	<p>Listen, don't tell.</p> <p>Do not talk about what you believe.</p> <p>Clarify misunderstandings, but don't try to convince.</p>
Ask open-ended questions	<p>Don't ask 'Do you...?' – this leads to yes/no answers that give minimal insight</p> <p>Do 'ask': "Tell me about the last time that you ... "</p>
Get them to tell stories	<p>Reflecting on their prior experience brings out more detail.</p> <p>Dig deeper when they talk about important parts of the story.</p>
Get facts, not opinions	<p>Don't ask "Would you ... ?"</p> <p>Do ask "How much did you ... ?"</p>
Ask "why" questions	<p>Motivation matters and helps uncover new facts.</p> <p>Ask: "Why do you need ... ?", "Why is this a pain?"</p>
Don't sell	<p>We are not interviewing to sell a solution.</p> <p>Don't explain our solution: "Wouldn't you like ... ?"</p> <p>Ask: "What are the most painful ... ?"</p>
Follow up	<p>Ask for contact details.</p> <p>Follow up with more questions if we need to later.</p>
Always open doors at the end	<p>Ask: "Who else should I talk to?"</p>
	Always!

Crafting interview questions

Identifying a good set of questions to gather insight is key to innovation.

Step 1: Define your audience – who do you want to learn about?

As part of our project on emergency alarms for older people, we gathered data using interviews from four different segments:

1. Older people who had existing emergency alarms.
2. Older people who had never had an emergency alarm.
3. Families of older people.
4. Existing emergency alarm companies.

In this section, we give some examples for the first segment—older people who had an existing emergency alarm with pendant, necklace, or wrist band, and a base station.

Step 2: Define your goals – what do you want to learn about?

It is good practice to first list the goals for interviews — what do you want to learn?

Example – Goals for interviews

For our segment of older people who have an existing emergency alarm, our goals were:

1. Learn how older adults actually use and experience the alarm system day-to-day.
2. Understand their feelings of safety, trust, and any challenges with the alarm.
3. Explore their support network and other ways they stay safe at home.

We can see here that the focus is not just on the alarm itself, but their experience, their feelings (safety, trust), and the larger ecosystem that they use for safety.

Step 3: Define your questions – how are you going to learn it?

Then, define a set of questions to achieve your goals. There is no fixed amount – it depends what you want to learn. However, we want to respect our participants by not taking up too much time, and not probing into personal details unless truly necessary.

Note: The interviews should be considered a **guide**, not a script.

Example interview questions – Medical alert system: Older people with an existing alarm

0. Do I have your consent to record this interview? I will automatically transcribe it and analyse the transcript later. You can choose not to answer questions, stop at any time, and ask me to delete the recording.
1. Can you tell me the story of how you decided to get the alarm system?
2. Can you tell me about the last time you used the pendant alarm? What happened, and what was it

- like for you?*If they haven't used it:* If you ever needed to use the alarm, how do you imagine it would help you?*Follow-up for all:* Are there any ways you worry it might not work as expected?
3. Can you walk me through what usually happens each morning with the green button? Have there ever been times you forgot to press it — can you tell me what that was like?
 4. How does having the alarm affect your day-to-day life? Why?
 5. Have you ever had a time when the alarm company contacted you? Can you tell me what that was like?
 6. When it comes to staying safe at home, how involved is your family?
 7. What other ways do you use to stay safe at home, besides the alarm?
 8. Is there anything we haven't talked about that you think we should know about using the alarm?
 9. Who else do you think I should talk to?

We can see that these questions are open-ended, focus on factual things mostly, plus some subjective experience. There are very few 'why' questions, but these are more prominent as follow-up questions later during the interview.

Importantly, these are interview questions early in the project when discovering our customers pain points, etc. – they do NOT ask about any solution ideas that we have.

Example of bad interview questions – Medical alert system: Older people with an existing alarm

Here is an example of interview questions that will do a really poor job of gaining insight into your customers. These questions are the same theme as the previous nine, but simply don't offer the same amount of exploration.

0. I'm recording this, ok?
(Does not give the participant a clear opportunity to decline.)
1. Why did you get the alarm system?
(Too broad and can lead to vague answers. Does not encourage a story.)
2. Have you used the pendant alarm before?
(Closed question that invites a yes/no answer and shuts down deeper discussion.)
3. Do you press the green button every day?
(Closed question that invites a yes/no answer and does not ask for detail or context.)
4. Tell me why you find the alarm helpful.
(Leading question (assumes the alarm is helpful) and vague (invites opinion without depth).)
5. When the alarm company calls you, do you usually answer?
(Closed question that invites a yes/no answer, does not ask about feelings or stories.)
6. Does your family help you with the alarm?
(Closed question that invites a yes/no answer, and does not get detail about how family is involved.)
7. Do you prefer the alarm or other safety methods?
(Closed question that invites a yes/no answer, and is a forced choice, no nuance.)

8. Is there anything else you want to add?
(Vague prompt that often gets a no.)
9. Can you give me the names of others with alarms?
(May feel intrusive, and closed as it does not encourage them to think broadly — of family, people without alarms, etc.)

Finding and approaching customers for interviews

Once we have the interview questions, we need to find potential customers and discuss with them.

Step 1: Find customers! This may involve cold-calling/emailing people in companies/organisations, such as a CIO or a potential user. For some projects, this could simply be approaching people in public—for example, if innovating in the rental market, approaching people waiting at a rental inspection would be perfect. For others, you could try attending conferences or trade shows. Often I organise workshops that potential customers come to out of curiosity.

Step 2: Ask them for their time. This is often the most difficult step for people new to interviewing, but it is quite straightforward. There are two parts to this:

- a. Start by asking for their help/time.
- b. Then explain why.

Don't start with explaining why— just straight out ask them and don't hide detail.

Example — Approaching an enterprise customer via email.

Dear [Name], Would you be open to a 15 minute Zoom call in the next week to discuss [topic] ? My name is [Name] and I'm part of a startup that is innovating in the sector and would like to learn about your experience to better understand the challenges and opportunities in this area. <etc. etc> I'm not trying to sell you anything and I'm happy to work around your schedule.

Note

You will get a positive response more often if you make it clear that you are not trying to sell anything

Example — Approaching a customer in public.

Hi. My name is [Name]. I was wondering if you have 10-15 minutes of your time to discuss your experience about [topic]? I'm in a startup that is innovating in the sector and would like to better understand the challenges and opportunities in this area. I'm not trying to sell you anything.

Step 2a: Refusals. If someone says 'no', just thank them and move on.

Example — Handling refusals

Be brief and polite. Here are some examples:

1. No worries! Thanks for your time.
2. I understand. Thanks! I'll let you get on with your day.
3. Thanks anyway, I appreciate it!

Conducting interviews

When conducting interviews, remember to think back to the goals of the interview, and direct the conversation to achieve those goals.

Step 1 — The interview

Make sure your recording equipment is running and you have your question ready.

Step 2 — During the interview

Some tips:

1. Listen!
2. Start broad (“Tell me about the last time...”) and then probe for more details.
3. Follow the flow of the conversation, not the strict ordering of questions. You can revisit any skipped questions later.
4. Wait 4-5 seconds after an answer before moving on to the next question. That pause often prompts people to add more insight, and that insight is often more useful than the first part of their answer!
5. If you find something interesting, follow up: “Why was that?”, “How?”, “What happened next?”.
6. Direct people back on topic if they drift off topic: “That’s helpful – let’s return to ...”.
7. Watch their behaviour – especially if you are getting them to show you how they do something, such as demonstrate how they use an existing product, or how they interact with your [MVP](#).

Step 3 – After the interview

Transcribe your recording (there are many free and excellent automatic transcription tools available), store the transcript AND the original recording **securely**. Anonymise the transcript if suitable for a given project.

Example of bad interviewing – Medical alert system: Older people with an existing alarm

In this example, I interviewed ChatGPT-4o, asking it to play the role of an older person with an emergency alarm.

I use the good questions from above, but do a poor job of interviewing as I simply ask the questions, get the answer, and move on — something that I see often when people don't take interviewing seriously.

Here we go:

Tim: Do I have your consent to record this interview? I will automatically transcribe it and analyse the transcript later. You can choose not to answer questions, stop at any time, and ask me to delete the recording.

Diane: Yes, that's fine. You can go ahead and record.

(Fail! I should have thanked her and acknowledged once the recording started.)

Tim: Can you tell me the story of how you decided to get the alarm system?

Diane: Well, it wasn't really my decision. My kids wanted me to get it after I had a fall. They said it would help them worry less. So... I got it.

(Fail! I should have followed up. Why did her kids want her to get one? What happened during that fall? How did she feel about getting the alarm at that time?)

Tim: OK. Can you tell me about the last time you used the pendant alarm? What happened, and what was it like for you?

Diane: I haven't used it, really. Not the red button anyway. Never needed to.

(Fail! I could have explored her feelings around this. How does it feel for her to have something she has never used?)

Tim: Ok. If you ever needed to use the alarm, how do you imagine it would help you?

Diane: I suppose I'd press the button and someone would come help. That's the idea, I guess.

(Fail! I could have explored her expectations around what would happen. Who do you think would respond? What do you think would happen? What would you want to happen?)

Tim: Yes. Are there any ways you worry it might not work as expected?

Diane: Hmm... maybe if I forgot to wear it, or if the battery went flat or something. But I don't think about it much.

(Fail! This was an opportunity to dig deeper into any concerns or any past experiences related to this.)

Tim: Ok. Can you walk me through what usually happens each morning with the green button? Have there ever been times you forgot to press it — can you tell me what that was like?

Diane: I'm supposed to press it before 11. Sometimes I forget. Then they call. It's a bit embarrassing, really. Makes me feel like I've done something wrong.

(Fail! Big fail! I could have asked why this is embarrassing? How does this embarrassment affect her use of this system, and her feelings towards it? Why does she feel like she has done something wrong?)

Tim: I see. How does having the alarm affect your day-to-day life? Why?

Diane: I don't know... I guess it's just... there. I don't wear the wristband unless the kids are coming over. Feels a bit silly, to be honest.

(Fail! I missed a huge opportunity to explore why it feels silly. Why does she only wear it if her children are coming over? What does wearing this wrist band mean to her?)

Tim: Ok. Have you ever had a time when the alarm company contacted you? Can you tell me what that was like?

Diane: Yeah, when I forget the green button. They call, ask if I'm alright. I say yes. That's about it. Bit annoying.

(Fail! I could have asked more about those calls. Who are "they"? What do they say? How long are the calls? What impact does it have on her feelings of her own safety? How do the calls make her feel about the relationship with the alarm company?)

Tim: Ok. When it comes to staying safe at home, how involved is your family?

Diane: Oh, my kids check in every day with a quick call. It's nice to know they care, even if it's brief.

(Fail! I could have explored that these calls are like and how they compare to the emergency alarm.)

Tim: Ok. What other ways do you use to stay safe at home, besides the alarm?

Diane: Well, I just have my smartphone with me. If something happens, I'd call them. That's about it.

(Fail! Missed opportunity to explore how she uses her smartphone and how comfortable she is using it. How does this compare to using the emergency alarm?)

Tim: Is there anything we haven't talked about that you think we should know about using the alarm?

Diane: No, I think we covered it. Nothing else really comes to mind.

Tim: Who else do you think I should talk to?

Diane: I don't know.

(Fail! What about her children? Anyone at the alarm company? What about friends who have had a fall but don't have an alarm? Friends with an alarm?)

Tim: Ok.

(Fail! I didn't thank her!)

In the above example, I fail multiple times by just simply not following up. Diane (the fake participant) was a bit closed, but I just took her answers. I only learnt from surface-level things such as whether she liked the alarm, whether she had used it, etc. Nothing that really helps me to design a better emergency alarm.

Let's try again!

Example of better interviewing – Medical alert system: Older people with an existing alarm

In this example, I interviewed ChatGPT-4o using the same questions and asking it to answer these in the same way. This time, I asked follow-up questions.

In the prompt, I asked ChatGPT-4o to annotate (the text in the square brackets) the benefits of any of the follow-up questions that I asked.

Tim: Do I have your consent to record this interview? I will automatically transcribe it and analyse the transcript later. You can choose not to answer questions, stop at any time, and ask me to delete the recording.

Diane: Yes, you have my consent.

Tim: Thanks! Ok. I'm recording now.

Can you tell me the story of how you decided to get the alarm system?

Diane: Well, I had a fall once, and after that, my children insisted I get one so I could stay in my own home safely.

Tim: I'm sorry to hear that you had a fall. Can you tell me what happened?

Diane: I slipped in the kitchen and wasn't able to get up for a while.

[This follow-up encouraged the participant to share more context and personal experience, building empathy and deeper understanding of the event.]

Tim: That sounds awful. How long were you on the floor?

Diane: It was a couple of hours before someone found me.

[Asking for specifics like duration helps highlight the severity and impact of the incident.]

Tim: Yes, I can imagine. What was going through your mind at the time?

Diane: I was scared and a bit helpless. I didn't know if anyone would come.

[Understanding emotional state reveals deeper feelings that influence decisions.]

Tim: So you said your children wanted you to get the alarm: what were your thoughts about that at the time?

Diane: I felt a bit pressured, honestly. Like they didn't trust me to manage on my own.

[This follow-up uncovers feelings about family dynamics and trust, key to understanding user motivation.]

Tim: How did their lack of trust make you feel?

Diane: It made me feel less independent, which I really don't like.

[Exploring feelings helps reveal barriers and pain points related to product acceptance.]

Tim: I can imagine.

Can you tell me about the last time you used the pendant alarm? What happened, and what was it like for you?

Diane: Actually, I haven't really had to use it. It's just there.

Tim: Well that's good to hear. How does it feel to be paying for this if you've never used it?

Diane: It feels a bit like I'm paying for something I don't really need.

[This question probes the perceived value and acceptance of the product, revealing potential dissatisfaction.]

Tim: Can you tell me more about what you mean by 'another thing you have to deal with'?

Diane: I just mean it feels like an extra hassle, something I have to remember every day, which I'm not always good at.

[This follow-up clarifies vague statements and identifies practical frustrations.]

Tim: Remember what?

Diane: I'm supposed to press the green button before 11. Sometimes I forget and they call. It's a bit embarrassing, really. Makes me feel like I've done something wrong.

Tim: Who are "they"?

Diane: The people at the alarm company.

Tim: How many times have you forgotten to press the green button?

Diane: Yeah—one morning I forgot to press the green button, and they rang around midday. I picked up and they were polite, just checking I was okay. But I felt embarrassed. Like I'd failed some simple task.

Tim: How long did the call last for?

Diane: It lasted a few minutes. They asked if I was feeling alright, if anything had happened, and reminded me to try and press the button every day. I said I was fine, just forgot. They were nice about it, but still—it felt a bit... awkward.

Tim: How did it make you feel that you had 'failed a simple task'?

Diane: I felt a bit stupid, honestly. Like I was letting everyone down, especially my kids. It's not a big deal, but in the moment it just sticks with me.

Tim: And you can't shake it for a while?

Diane: No, it lingers for the rest of the day. I keep thinking about it and feeling like a burden. It's a tough feeling to shake off.

Tim: I'm sorry to hear that.

Diane: Thank you. It's not easy to talk about, but I'm glad you're listening.

[*These explore the emotional impact of the alarm company's call, encouraging the interviewee to express their feelings, which deepens understanding of the personal challenges behind routine safety checks.]

Tim: It's what I do!

Tell me a bit more about the wristband being ugly.

Diane: It's bulky and it makes me feel old and frail. I don't like wearing it. Actually, I only wear it when my kids are coming over.

[*This reveals design and emotional issues that affect usage and acceptance.*]

Tim: I assume your kids don't know about that last part. How do you think they'd react if they knew?

Diane: They'd probably be upset because they want me to be safe.

[*This uncovers possible unspoken tensions or compromises in user behavior.*]

Tim: And you hate feeling frail and needy?

Diane: Yes, it's not a feeling I want to have.

[This exposes emotional resistance to the product's implications. The question builds rapport with the user.]

Tim: I can understand that.

If you ever needed to use the alarm, how do you imagine it would help you?

Diane: I guess it would alert someone that I need help.

Tim: Who do you think would respond?

Diane: Probably the alarm company first, then my children.

[This clarifies expected response chains and real-world functioning.]

Tim: What do you think would happen if you pressed the red button?

Diane: Someone would come to help me, I hope.

Tim: What would you want to happen?

Diane: I'd want my children to come if possible, and an ambulance if needed.

Tim: So you'd prefer your children to come first?

Diane: Yes, definitely.

[This helps understand preferences and priorities in emergency response.]

Tim: Indeed. Are there any ways you worry it might not work as expected?

Diane: Sometimes I worry if I forget to press the green button and it won't catch that I'm in trouble.

Tim: Do you worry about it not working in other ways?

Diane: Not really, mostly just forgetting the green button.

[This narrows down the main user concerns to specific failure modes.]

Tim: The anxiety about it failing brings back memories of your earlier fall?

Diane: Yes, it reminds me of that scary time.

[Connecting past experiences to current worries deepens understanding of user emotions.]

Tim: Can you walk me through what usually happens each morning with the green button?

(Note: In the list of questions, at this point the question should be "Have you ever had a time when the

alarm company contacted you? Can you tell me what that was like?”, but the participant already covered this, including how it was embarrassing. Instead, I asked a more specific question about the daily routine, and skipped the rest.)

Diane: I try to press it by 11 am every day to let them know I’m okay.

Tim: And you remember to press it most mornings?

Diane: Most days, but sometimes I forget if I’m in a rush or distracted.

Tim: So when you are in a rush or distracted, you forget to push the button sometimes?

Diane: Exactly.

Tim: I see. How does having the alarm affect your day-to-day life? Why?

Diane: It feels like one more thing I have to manage and a reminder that I’m not as independent as I used to be. I feel checked on.

Tim: Checked on?

(Such a simple follow-up question! I wanted to learn more, so just reflected their comment as a question.)

Diane: Yeah, my children call sometimes, and the alarm company calls if I forget the button.

Tim: What does freedom from this look like?

Diane: Just feeling like I can live without all these reminders and gadgets telling me I need help.

Tim: Who does it feel like is watching?

Diane: Mostly the alarm company, but also my kids in a way. Like the whole setup is there because they don’t fully trust I’ll be okay on my own.

Tim: It’s like the roles have reversed from when they were young and they were parent!

Diane: Exactly! That’s what it feels like sometimes. Like they’re parenting me now, and I didn’t really sign up for that.

[This interaction moves beyond surface-level facts to explore the emotional impact, frequency, quality, and personal meaning of family involvement, revealing important nuances about support and connection that would otherwise be missed.]

Tim: When it comes to staying safe at home, how involved is your family?

Diane: They check in pretty regularly, mostly by phone. Not a lot of face-to-face, but they want to make sure I'm okay. Sometimes they stop by, too.

Tim: How often do they call?

Diane: They used to call every day, especially right after my fall. Now it's more like a few times a week. They probably trust the alarm system helps a bit.

Tim: What are these calls like? How long do they last and what do you discuss?

Diane: They're usually pretty short, just checking in to see how I'm doing. Sometimes it's a quick "Hi, are you okay?" and other times we might chat a little about my day. Mostly it feels a bit routine, like a checkbox.

Tim: How does that make you feel?

Diane: It's nice they care, but sometimes it feels a bit impersonal. Like they're just doing what they think they should, rather than really wanting to connect.

Tim: You'd like to connect more?

Diane: Yeah, I think so. It would be nice to have conversations that feel a bit more genuine, rather than just quick check-ins. Maybe talk about other things besides my health sometimes.

Tim: How do these calls compare to the emergency alarm?

Diane: They feel different. The family calls are more personal, even if they're brief. The alarm calls feel formal and a little cold—just a job they have to do. Both are important, but in different ways.

[This interaction uncovers the frequency, nature, and emotional tone of family interactions, revealing a desire for deeper, more genuine connection beyond routine check-ins, and contrasts the warmth of family calls with the formality of alarm company contact, enriching the interviewee's lived experience around safety and care.]

Tim: What other ways do you use to stay safe at home, besides the alarm?

Diane: I mostly rely on my phone. It's always with me, and I can call anyone if I need help.

Tim: The phone is key for you?

Diane: Yes, it feels more natural and less intrusive than the alarm.

Tim: More natural how?

Diane: Like, it's something I'm used to and comfortable with. I don't have to remember extra buttons or wear something I don't like.

Tim: I think smartphones are part of all of us, these days!

Diane: Definitely.

[These follow-up questions clarify the interviewee's preference for familiar technology over specialised devices, exploring feelings of comfort and intrusion, and help highlight the importance of natural, user-friendly solutions in safety design.]

Tim: I agree with that!

Is there anything we haven't talked about that you think we should know about using the alarm?

Diane: I guess sometimes I just don't want to complain or cause trouble by saying I don't like it.

Tim: You feel that you shouldn't complain?

Diane: Yes, I don't want to seem ungrateful or difficult.

Tim: I see.

Who else do you think I should talk to?

Diane: You could talk to some of my children, and maybe some of my friends who also use alarms. Also, some people at the alarm company would have insight.

Tim: I really appreciate you giving your time to me today. This was really helpful and insightful. I hope the whole experience improves for you and that you can get back some of that feeling of independence! Thanks.

Diane: Thank you, Tim.

Tim: I'm ending the recording now.

Chapter summary

The key takeaways for this chapter are:

- Interviewing is about **listening** and **learning**. It is **not** just about asking questions.
- **Open-ended questions** elicit more valuable information than closed questions.
- Asking interviewees to tell **stories/anecdotes** elicits more valuable information than asking direct questions.
- Clearly defining the **goals** for your interview—what you want to learn—is more important than defining the questions.

PART II

DEFINING THE OPPORTUNITY

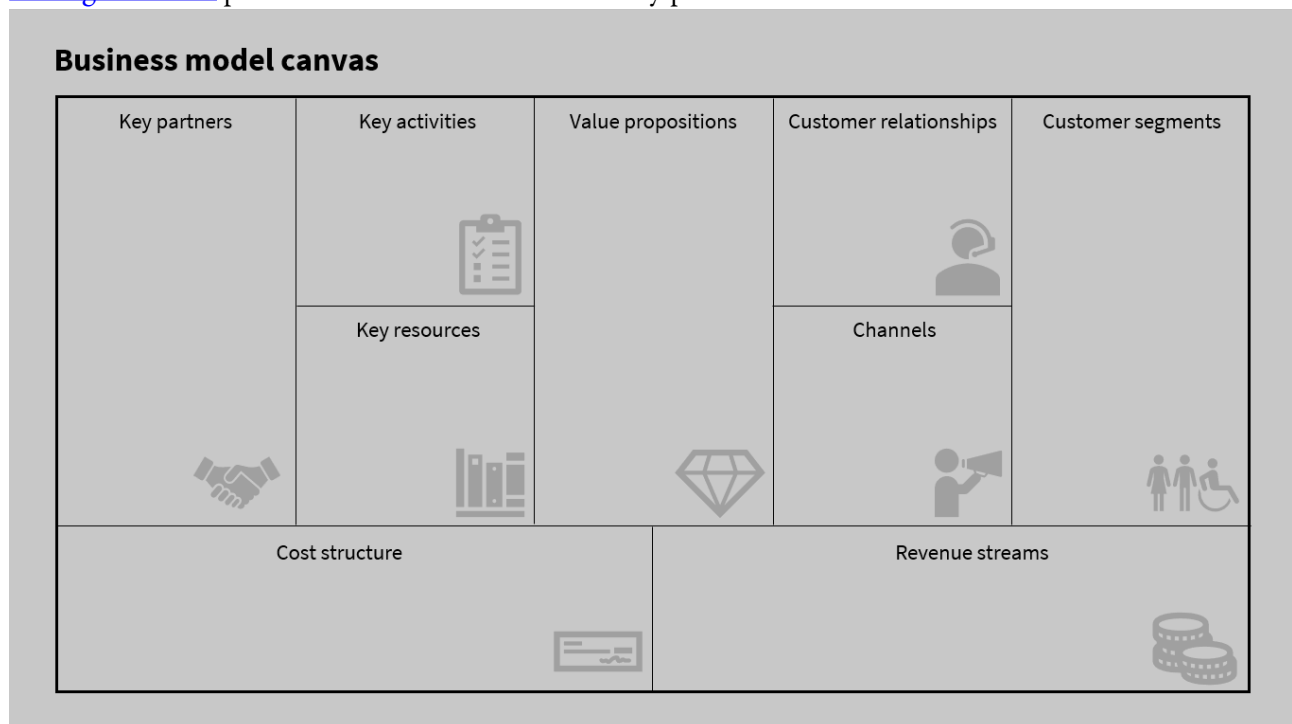
4.

BUSINESS MODEL CANVAS

In this chapter, we look at the **business model canvas**, which is a simple yet useful tool for describing and visualising a business model.

Business models and customer development

[The figure below](#) provides an overview of the nine key parts of a business model:



The structure of a business model canvas. “[Business model canvas](#)” by [Strategyzer](#), shared under a [Creative Commons Attribution ShareAlike 3.0 Unported licence](#).

Let’s define each of these:

1. **Customer segments** define one or more groups of people who (we think) will be our customers, and a description of the problems that they are experiencing related to our vision. Typically, very few products appeal to everyone because people don’t all have the same problems. Instead, we look at groups of people who share common characteristics. We can define customer segments based on their occupation, hobbies, demographics (e.g. age, location, gender, sexuality, etc.), living situation (alone, share house, family house, renting, owning), and many others. We specify what the problems are for each of these segments. For this, we need to answer the question: “Who are our customers

and what are their problems?”

2. **Value propositions** are statements (either guesses or verified insights) that describe how to solve customer problems. For this, we need to answer the question: “What are the solutions that we can make to solve our customers’ problems in each segment?”
3. **Channels** are the mechanisms by which solutions are delivered to customers; e.g. a web application, via distribution to small businesses, via YouTube. For this, we need to answer the question: “How do our customers in each segment want to be reached to learn about this and use it?”
4. **Customer relationships** are the mechanisms that are used to engage and build trust between organisations and customer segments. These may be different for different customer segments. For this, we need to answer the question: “How do our customers want to interact with us and how do we want them to ‘see’ us?”
5. **Revenue streams** are to identify how value propositions will generate revenue from the customer segments to the organisation. For this, we need to answer the question: “Which value propositions are customer segments willing to pay, and how much?”
6. **Key resources** identify what is needed to successfully deliver our value propositions; e.g. cloud services, physical resources (delivery vehicles, hardware, etc.), service subscriptions. For this, we need to answer the question: “What are the resources required to deliver our value propositions?”
7. **Key activities** identify what needs to be done to successfully deliver our value propositions. For this, we need to answer the question: “What are the things we need to do to deliver our value propositions?”
8. **Key partners** identify groups outside of the organisation that help the product succeed; e.g. suppliers, distributors, etc. These are typically people or organisations who can help do key activities, or give access to key resources. If we look at our channels, we often see delivery occurs via *indirect channels*, meaning not via us directly, but via another organisation. These organisations are partners. For this, we need to answer the question: “Who are the people we need to work with to deliver our value propositions?”
9. **Cost structures** identify the expenses that are needed to operate; e.g. for salaries, rent, utilities, services, and technology. For this, we need to answer the question: “What are the costs that we will incur to deliver our value propositions?”

At each point, any of these could be either a hypothesis (a guess that we have made using some assumptions) or an insight based on data (something we have gone out and verified).

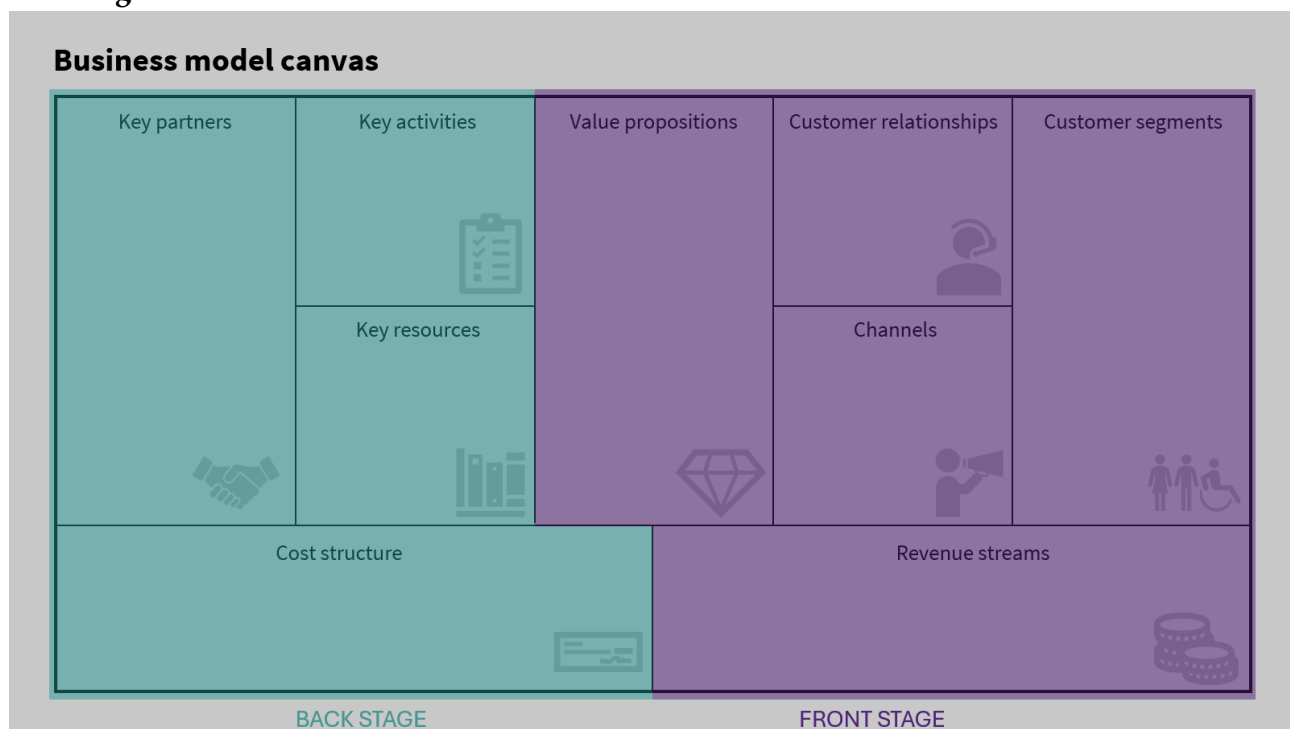
We can see that this is **customer focused**, not product focused. It targets the problems of the customers directly.

Will we need to do all of these in this course?

In this course, we will focus mostly on the first five items:

1. Customer segments: who are our customers?
2. Value propositions: what can we do for them?
3. Channels: how will we deliver our product/service?
4. Customer relationships: how do we interact with our customers?
5. Revenue streams: will customers pay for it? If yes, how?

These are the so-called **front stage** of the business model canvas. The remaining steps 6-9 are called the **back stage** of the business model canvas.



Front stage vs. back stage of a business model canvas. Adapted from “[Business model canvas](#)” by [Strategyzer](#), modified under a [Creative Commons Attribution ShareAlike 3.0 Unported licence](#).

The back stage activities are crucial in building and scaling a business. We will be discussing these but will only have time to scratch the surface in going out and verifying them.

The business model canvas

Definition — Business model canvas (BMC)

A **business model canvas** is a visual description of the nine key parts of a business model. It consists of (a mix of) hypotheses and facts about the business model. It can be used to describe existing business models, describe new business models, and help us focus on the key questions of a new business.

The business model canvas is something that we should complete for any innovation project. The value of a business model canvas is not just in *having* it (although it is useful to visualise our business model). The value comes from *doing* the activities that we need to fill out the different sections. By asking the right

questions and gathering the right data to fill in the canvas, we learn about our customers, their problems, what we can do to solve them, and what we need to deliver those solutions.

We do not need to use this format directly – business models canvases can just be represented on pieces of paper, as notes on a Miro board, as cards on a Trello board, or as activities on a Jira board. However, the format in [Fig. 9](#) allow us to easily visualise the core components of our business model, so can be useful to get a high-level snapshot and to help other people (e.g. investors, collaborators) quickly understand the business model.

Further, this format of the canvas has been designed carefully so that the related parts of the model are next to each other. For example, value propositions serve customer segments, but they do so via customer relationships and channels; the key activities and key resources are required to deliver value propositions; and so on.

Customer development and the business model canvas

Effectively, the customer development process is what helps us to “fill out” our business model canvas.

Focusing on customer discovery, we start with our vision, and brainstorm a set of hypotheses around our vision. We should have at least one in each of the cells of the business model canvas — and often, multiple. There is no set number or “standard” number of hypotheses – it depends on your vision and what you have found. Typically though, the number in each cell is just a handful (2-7) in each cell, with customer segments and value propositions often containing more than others – many value propositions can be ‘delivered’ using the same activities and resources.

Customer development helps us to do three main things:

1. Turn some of our hypotheses into verified hypotheses (sometimes called “facts”), which effectively means that we collect enough data such that we are confident that our hypotheses (our guesses) are accurate.
2. Turn some of our hypotheses into rejected hypotheses, which effectively means that we collect enough data such that we are confident that our hypotheses (our guesses) are inaccurate.
3. Create new hypothesis from what we learn.

Throughout the course, the business model canvas and the related [value proposition canvas](#) are used to guide what we do. We express what we think is true (hypotheses), we add them to the canvases, we test them by [experimentation](#), and then we iterate until we have a business model canvas with a set of accepted hypotheses that are sufficient for us to move on to the next step.

Chapter summary

The key takeaways for this chapter are:

1. **Business model canvases** are a way to understand and represent business models.
2. They are **customer focused**.
3. A business model canvas can be separated into **front stage** (things the customer sees) and **back stage** (things the customer does not see).
4. We use the business model canvas as a way to **guide our thinking** and our activities, as well as a way to **represent and visualise** what we have found

5.

VALUE PROPOSITION CANVAS

Definition — Value proposition canvas

A **value proposition canvas** is a visual description of our customer segments, and our value propositions to solve the customers' problems.



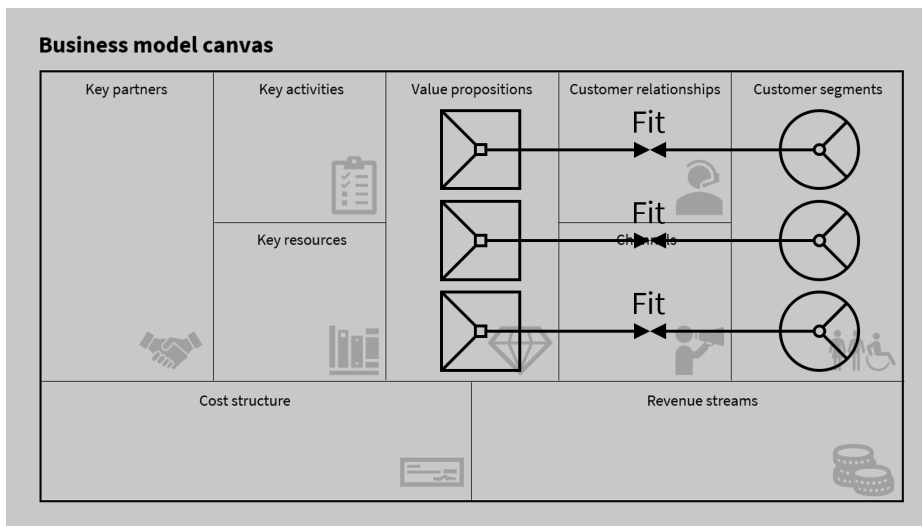
One or more interactive elements has been excluded from this version of the text. You can view them online here: <https://uq.pressbooks.pub/introduction-software-innovation/?p=68>

The [Value Proposition Canvas](#) by [Strategyzer](#).

The figure above shows the structure of a value proposition model. It contains two parts:

1. A **customer profile**, which describes the customers within a single segment, including their problems.
2. A **value map**, which describes the value propositions that potential solutions bring to that segment.

Value proposition canvases are used to define the value propositions and customer segments of a business model canvas. Effectively, for every segment, we map the segment to the value propositions that solve that segment's problems. Therefore, there can be **multiple** value proposition canvases within a single business model:



The relationship between a value proposition canvas and a business model canvas. Adapted from “[Business model canvas](#)” by [Strategyzer](#), modified under a [Creative Commons Attribution ShareAlike 3.0 Unported licence](#).

The four types of startup markets

When we are starting up a new product or company, there are four broad types markets:

1. **Entering an existing market:** The market is well-established and we are simply providing something with a different slant.
2. **Creating an entirely new market:** There is currently no market and we are establishing it.
3. **Re-segmenting an existing market as a low-cost provider:** There is a well-established market, and we are providing a service at a lower price than competitors.
4. **Re-segmenting an existing market as a niche player:** There is a well-established market, and we are appealing to a niche part of the market via e.g. higher quality.

The features of a product are driven by how well they satisfy a market: the **product/market fit**. Larger companies tend to be market-driven, while startups tend to be product-driven.

Customer profile



One or more interactive elements has been excluded from this version of the text. You can view them online here: <https://uq.pressbooks.pub/introduction-software-innovation/?p=68>

The structure of a customer profile. [The Customer Profile](#) by [Strategyzer](#).

Customer jobs: Trigger questions

1. What is the one thing that your customer couldn't live without accomplishing? What are the stepping stones that could help your customer achieve this key job?
2. What are the different contexts that your customers might be in? How do their activities and goals change depending on these different contexts?
3. What does your customer need to accomplish that involves interaction with others?
4. What tasks are your customers trying to perform in their work or personal life? What functional problems are your customers trying to solve?
5. Are there problems that you think customers have that they may not even be aware of?
6. What emotional needs are your customers trying to satisfy? What jobs, if completed, would give the user a sense of self-satisfaction?
7. How does your customer want to be perceived by others? What can your customer do to help themselves be perceived this way?
8. How does your customer want to feel? What does your customer need to do to feel this way?
9. Track your customer's interaction with a product or service throughout its lifespan. What supporting jobs surface throughout this life cycle? Does the user switch roles throughout this process? [\[1\]](#)

Customer pains: Trigger questions

1. How do your customers define too costly? Takes a lot of time, costs too much money, or requires substantial efforts?
2. What makes your customers feel bad? What are their frustrations, annoyances, or things that give them a headache?
3. How are current value propositions under-performing for your customers? Which features are they missing? Are there performance issues that annoy them or malfunctions they cite?
4. What are the main difficulties and challenges your customers encounter? Do they understand how things work, have difficulties getting certain things done, or resist particular jobs for specific reasons?
5. What negative social consequences do your customers encounter or fear? Are they afraid of a loss of face, power, trust, or status?
6. What risks do your customers fear? Are they afraid of financial, social, or technical risks, or are they asking themselves what could go wrong?
7. What's keeping your customers awake at night? What are their big issues, concerns, and worries?
8. What common mistakes do your customers make? Are they using a solution the wrong way?
9. What barriers are keeping your customers from adopting a value proposition? Are there upfront investment costs, a steep learning curve, or other obstacles preventing adoption? [\[1\]](#)

Customer gains: Trigger questions

1. Which savings would make your customers happy? Which savings in terms of time, money, and effort would they value?
2. What quality levels do they expect, and what would they wish for more or less of?
3. How do current value propositions delight your customers? Which specific features do they enjoy? What performance and quality do they expect?
4. What would make your customers' jobs or lives easier? Could there be a flatter learning curve, more services, or lower costs of ownership?
5. What positive social consequences do your customers desire? What makes them look good? What increases their power or their status?
6. What are customers looking for most? Are they searching for good design, guarantees, specific or more features?
7. What do customers dream about? What do they aspire to achieve, or what would be a big relief to them?
8. How do your customers measure success and failure? How do they gauge performance or cost?
9. What would increase your customers' likelihood of adopting a value proposition? Do they desire lower cost, less investment, lower risk, or better quality? [\[1\]](#)

Prioritisation

Once we have a list of jobs, pains and gains, we prioritise and rank them:

- Jobs: what are the high-value jobs?
- Gains: what is the relevance of the gain?
- Pains: what is the significance of the pain?

Value map



One or more interactive elements has been excluded from this version of the text. You can view them online here: <https://uq.pressbooks.pub/introduction-software-innovation/?p=68>

The structure of a value map. [The Value Map](#) by [Strategyzer](#).

Pain relievers: Trigger questions

Could your product:

1. produce savings? In terms of time, money, or efforts.
2. make your customers feel better? By killing frustrations, annoyances, and other things that give customers a headache.
3. fix under-performing solutions? By introducing new features, better performance, or enhanced quality.
4. put an end to difficulties and challenges your customers encounter? By making things easier or eliminating obstacles.
5. wipe out negative social consequences your customers encounter or fear? In terms of loss of face or lost power, trust, or status.
6. eliminate risks your customers fear? In terms of financial, social, technical risks, or things that could potentially go wrong.
7. help your customers better sleep at night? By addressing significant issues, diminishing concerns, or eliminating worries.
8. limit or eradicate common mistakes customers make? By helping them use a solution the right way.
9. eliminate barriers that are keeping your customer from adopting the value propositions? Introducing lower or no upfront investment costs, a flatter learning curve, or eliminating other obstacles preventing adoption. [\[1\]](#)

Gain Creators: Trigger Questions

Could your product:

1. create savings that please your customers? In terms of time, money, and effort.
2. produce outcomes your customers expect or that exceed their expectations? By offering quality levels, more of something, or less of something.
3. outperform current value propositions and delight your customers? Regarding specific features, performance, or quality.
4. make your customers' work or life easier? Via better usability, accessibility, more services, or lower cost of ownership.
5. create positive social consequences? By making them look good or producing an increase in power or status.
6. do something specific that customers are looking for? In terms of good design, guarantees, or specific or more features.
7. fulfil a desire customers dream about? By helping them achieve their aspirations or getting relief from a hardship?
8. produce positive outcomes that match your customers' success and failure criteria? In terms of better performance or lower cost [\[1\]](#)

Six ways to innovate from a customer profile

Alex Osterwalder outlines six ways to innovate from customer profiles [\[1\]](#):

Strategy	Explanation
Address more jobs?	Can we help customers complete more jobs in their day?
More important jobs?	Can we help customers do something more valuable?
Beyond functional jobs?	Can we help customers create value by fulfilling important social and emotional jobs?
Help more customers?	Can we help more customers by making jobs easier or cheaper?
Do a job better?	Can we help customers perform existing jobs at a higher quality?

Competitor Analyses

Once we have a solid idea of our value proposition, we need to analyse our competitors.

Why is a competitor analysis important?

There is no point trying to sell a product if a competitor does it as good or better than we do. A product is only successful if it offers a **unique** value proposition.

If you think: “we don’t need a competitor analysis” – we know our value proposition is unique, a clear response is: how do you know it is unique if you haven’t done a competitor analysis?! There are **many** “unique” apps on the Apple and Android app store that all do much the same thing.

Further, analysing how our competitors solve their customers’ problems can give us insight in two ways:

1. If they do something well, we can mimic it (provided it is not patented or copyrighted).
2. If they do something poorly, we can learn from this and try to do it better.

What is a competitor analysis?

A competitor analysis is conceptually simple, but can be challenging to do.

It typically has the following parts, which correspond to the process that we can follow:

1. **Define unique value proposition:** Write a short, clear statement of what we believe our unique value proposition is.
2. **Find competitors:** Search for competitors who are solving the same problems.
3. **Analysis:** Complete a **competitor table** and a short *Strengths, Weaknesses, Opportunities, Threats*

(SWOT) analysis of the top few competitors.

4. **Insights and strategy:** Summarise what we learnt and how it changes our plans.

Define unique value proposition

We should have already defined our value proposition by this point, but it helps to start a competitor analysis with a short, clear statement that specifies it:

Our project, **[Project Name]**, aims to solve **[brief problem statement]** for **[target customer segment]**. We believe that our unique value proposition is **[value proposition]**.
The following competitor analysis includes a comparison table, SWOT analysis of key competitors, and strategic insights based on our findings.

Find competitors

Next, find our closest competitors. There are two broad types of competitors:

1. **Direct competitors:** Offer the *same type of solution* to the same customer segment and their same problems. For example, someone offering a product that connects families via mobile devices is a direct competitor to the emergency alarm ‘picture frame’.
2. **Indirect competitors:** Offer a *different type of solution* to the same customer segment and their same problems. For example, existing emergency alarm companies offered a service rather than connecting families directly – same pain, very different solution.

How do we do find competitors?

Here are 5 ways:

1. **Customer interviews:** We can look at our customer interviews! If we did our job properly, we should have been probing how our customers currently solve their problems. This gives us insight into our competitors.
2. **Search engines:** We can run the types of queries we expect that our customers would run when trying to solve their problems.
3. **Customer channels:** Look at distribution channels, such as app stores or other product platforms. For physical products, we could look at online sellers or even go into stores.
4. **Social media:** Browse forums, etc., about your customers’ problems to see what products people currently use.
5. **Industry trade shows, blogs, etc:** Look for public forums about the specific industry, and do

things like attend trade shows to learn about competitors.

Analysis

Put together all of the competitors that we have analysed into a **competitor table**:

Competitor	Direct/ indirect	Target Customers	Key Features	Pricing Model	Strengths	Weaknesses
Competitor A	Direct	Small businesses	Automation, dashboard	Subscription	Easy to use	Limited integration
Competitor B	Direct	Enterprises	Custom workflows, analytics	Tiered pricing	Scalable	Complex setup
Competitor C	Indirect	Freelancers	Simple UI, mobile app	Freemium	Affordable	Few advanced features
...
Our solution	Direct	Freelancers	Simple UI, mobile app	Freemium and premium	Affordable	Mobile only

Next, for the most closely related competitors, that is, those whose market share you are hoping to steal, do a **SWOT** analysis:

- **Strengths:** What do they do well? What unique capabilities do they have?
- **Weaknesses:** Where do they struggle? What are people's complaints about their product?
- **Opportunities:** In the market they are in, what are trends or gaps that they can benefit from?
- **Threats:** In the market they are in, what challenges do they have? What is changing that could make them less valuable?

Strengths may be similar to opportunities, and weaknesses to threats. The difference is:

- Strengths and weaknesses focus on the properties of the organisation or product.
- Threats and opportunities focus on the properties of the market they are in.

Example – Competitor analysis

Competitor A

- **Strengths:** Intuitive interface, strong customer support.
- **Weaknesses:** Limited scalability for larger teams.
- **Opportunities:** Growing demand for lightweight tools.
- **Threats:** New entrants with better integration capabilities.

Competitor B

- **Strengths:** Robust feature set, enterprise-grade security.
- **Weaknesses:** High learning curve.
- **Opportunities:** Expansion into mid-market.
- **Threats:** Price-sensitive customers switching to cheaper alternatives.

Our product

- **Strengths:** Enterprise-grade security, low cost.
- **Weaknesses:** Minimal features.
- **Opportunities:** Expansion into low-cost market.
- **Threats:** Existing providers offering low-cost options in future.

Tip

You should know your closest competitors so well that you should be able to fill out a business model canvas about their product(s)!

Insights and strategy

Now that we understand our competitors, what can we learn from how they operate? A nice way to think about this is:

1. What did we learn about our competitors?
2. What does that change about our own understanding or hypotheses?

Example – Insights and strategy

Insight	Strategic implication
Most competitors focus on scalability	Can we focus on being niche and quality to differentiate ourselves?
All charged more than we expect them to	Are we under-estimating the costs to do this? We need to test price more/better
There is a gap in direct family-to-older-person communication	We need to test the hypothesis that people want this
Most competitors had a complicated sales and sign-up process	We need to avoid over-complicating this.

Example – Medical Alert System Value Proposition

Based on the [refined hypotheses](#) defined previously during our customer discovery, and informed by the value proposition canvas and competitor analysis, we can express the value propositions as follows:

Value Propositions – Families Members (Primary Paying Customers)

- Provide daily reassurance that their older family member is safe, without requiring constant calls or intrusive monitoring.
- Reduce anxiety through simple, regular confirmation of wellbeing rather than relying only on emergencies.
- Enable meaningful connection through everyday interactions.
- Offer peace of mind without undermining the older person’s independence or dignity.

Value Propositions – Older People (End Users)

- Support a sense of safety while maintaining independence and autonomy.
- Feel cared about, rather than cared for or monitored.
- Avoid compulsory daily “check-in” behaviours that feel controlling or medicalised.
- Encourage simple, enjoyable interaction with family as part of everyday life.
- Preserve dignity by avoiding visible or stigmatising safety devices.

Chapter summary

The key takeaways for this chapter are:

1. A **value proposition canvas** describes our customers, their problems, and the value propositions designed to address them.
2. It contains two parts: a **customer profile** and a **value map**.
3. A customer profile describes the **jobs, gains, and pains** of customers in a segment.
4. A value map describes the **pain relievers, gain creators, and products/services** behind these.
5. We analyse **competitors** to identify **unique value propositions** and to **learn** from them.

References

[1](1,2,3,4,5,6) Alex Osterwalder. *Achieve product-market fit with our brand-new Value Proposition Canvas*. <https://www.strategyzer.com/library/achieve-product-market-fit-with-our-brand-new-value-proposition-designer-canvas>

PART III

EXPERIMENTATION AND VALIDATION

6.

EXPERIMENTATION: TESTING AND VALIDATION

In this section, we discuss a process for running experiments to test hypotheses on our business model canvas.

Experimentation

Why experiment? We experiment to **gain insights** about customers, their problems, our solutions, and any other important assumptions we make about our product.

There are two types of experiments:

- **Generative/exploratory:** These experiments help us to explore a space.
- **Evaluative:** These are hypothesis-driven and aim to find data to support or refute a hypothesis.

There are two **reasons** that we experiment:

- To evaluate our understanding of our customers and market fit.
- To evaluate products.

The process of experimentation is summed up with this excellent process model from Steve Blank:

<https://twitter.com/AlexOsterwalder/status/525527176272039936>

The process model for experimentation.

Step 1: Extract the hypotheses from your business model canvas (or define them if you haven't yet!)

Step 2: Prioritise the hypotheses from what you believe are the most valuable to the least valuable. Based your prioritisation on things like:

- The risk to your product, where risk is probability times impact. If the impact on your product of the hypothesis being wrong is low and the probability of it being right is high, it is lower priority. If the impact is high and the probability low or unclear, then it is higher priority.
- How much of a problem the hypothesis is for customers.

- How much the customers would be willing to pay to have the problem solved.

Step 3: Design tests to gather data for your hypotheses and capture these as **test cards**.

Step 4: Prioritise your tests. Base your prioritisation on things like:

- The priority of the hypothesis.
- How easily you can test the hypothesis.

Step 5: Run the tests!

Step 6: Capture the learnings in **learning cards**.

Step 7: Decide what to do based on what you have learnt.

Then **repeat!**

Hypotheses

At this point, our canvases are full of *hypotheses*, not *facts* (yet!).

But what exactly is a hypothesis?

Definition – Hypothesis

In the context of innovation, a **hypothesis** is a falsifiable statement or prediction about a business idea, product, or customer that can be investigated through empirical observation and experimentation.

This is very similar to the definition of “hypothesis” in science.

Note this important property: **falsifiable**. This means that we must write the hypothesis in such a way that we are able to demonstrate it is false using empirical data.

“In so far as a scientific statement speaks about reality, it must be falsifiable: and in so far as it is not falsifiable, it does not speak about reality.” – Karl Popper, *The Logic of Scientific Discovery*, p. 316.[\[2\]](#)

Example – Non-Falsifiable hypothesis

“This solution will help this segment of customers improve their health over the long term.”

This is not falsifiable for three (well, two and a half) reasons:

1. The timeframe “over the long term” is not specific enough to measure. If we find it has had no impact within 10 years, anyone could claim “10 years is not long term”.

2. The group “this segment” is not specific enough. If it improved 95% of customers’ health within a year, is this hypothesis supported or refuted? 95% is a lot, but the hypothesis states “this segment”, which could be interpreted as “everyone in this segment” or perhaps “a majority of people in this segment”.
3. It is not testable *right now* because it talks about the long term. If we mean 10 years, we cannot really test it for 10 years.

Example – Falsifiable hypothesis

“We believe that adding an inventory management function to this product will achieve a customer retention rate of 75% over the first year.”

This is falsifiable:

1. It specifies what will happen: 75% retention rate.
2. This is measurable: we can measure retention by seeing which customers at the start of the year are still there at the end of the year.
3. The timeframe is clear: the first year.

Framing our assumptions like this treats innovation as a scientific process. Just like designing a scientific experiment, we design an experiment to test our hypotheses; and further, the aim of the experiment should be to try to prove that the hypothesis is wrong!

“The point is that, whenever we propose a solution to a problem, we ought to try as hard as we can to overthrow our solution, rather than defend it. Few of us, unfortunately, practice this precept; but other people, fortunately, will supply the criticism for us if we fail to supply it ourselves.” – Karl Popper, *The Logic of Scientific Discovery*, p. xix.^[2]

Test cards

Test cards, a product of [Strategyzer](#), are a good way to provide some structure to the hypotheses and experiments that we run. The format is below:

Each test card has four parts:

- **“We believe that ...”** is the hypothesis that we have — the assumption that we have about a customer segment, their problem, etc. – anything on the business model canvas!

- **“To verify that, we will ...”** what we will do to test the hypothesis, which almost always involves finding data; e.g. talking to customers, finding existing data, etc.
- **“And measure ...”** specifies exactly what we want to measure; e.g. number of customers who say X, or number of sales of existing product Y.
- **“We are right if ...”** specifies the criteria for either accepting or rejecting the hypothesis.

In addition, I often encourage people to add a fifth element outlining the questions to be asked, if talking to customers is the data collection:

- **“Therefore, we will ask ...”** defines what interview questions we will ask our interviewees to test the hypothesis.

This final part (interview questions) is not typically included on test cards, such as those used by Strategyzer. It was an innovation that one of the early COMP1100 teams added to their test cards that I thought brought value, so I adopted it.

Note

Note that we often gather data from sources other than interviews, but if we cannot, it helps to have a clear set of initial questions to ask.

Structuring hypotheses

A good hypothesis statement (the “We believe that ...” statement) should identify a target, a behaviour or observable phenomenon, and a reason for the behaviour/phenomenon:

We believe that **[target]** are **[showing behaviour / displaying interest in]** **[for reason Y]**.

Structuring verification statements

A good verification statement (the “To verify that ...” statement) should describe a particular behaviour that we will undertake to collect the right data to test the hypothesis:

To verify that, we will **[undertake behaviour]** to **[identify behaviour]**.

Structuring measurement statements

A good measurement statement (the “And measure ...” statement) should identify exactly what data will be measured:

And measure **[particular variable]**.

Structuring accept/reject statements

A good accept/reject statement (the “We are right if ...”) should have a clear and measurable statement that makes it easy to determine whether the hypothesis is accepted or rejected:

We are right if **[some measurable statement]**.

Structuring interview question statements

A good set of questions (the “Therefore, we will ask ...” statement) should identify the questions that we need answered, in order to gather the data

And ask **[specific questions]**.

We can see here that these statements make it clear to determine whether to accept or reject the hypothesis: we count the number of people who said something, and then calculate the percentage.

Steve Blank states that we should make bets with ourselves (or with each other if in teams) to help make the success/failure definition clear. By framing it as a bet, it helps us to make these concrete.

Example – Medical alert system test card: **users**

We believe that our end users are people over 65 who live on their own and would use technology to keep in touch with their family more often because they do not see them much.

To verify that, we will interview 50 people over 65 who live on their own.

And measure how many would like more regular digital interaction with their family.

We are right if 70% of older people say that they would like more regular digital interaction with their family.

Therefore, we will ask “how often would you like to interact with your family members?”

Example – Medical alert system test card: **paying customer**

We believe that our paying customers are close family members of our end users who would use technology to keep in touch with older family members using a photo-sharing app rather than delegate to a service because they would like assurance that their family members are doing well in a more personal manner.

To verify that, we will interview 50 people who have family members over 65 that live on their own and have an existing emergency alarm system.

And measure how many would prefer regular digital interaction with their family member rather than delegating to a service.

We are right if 70% of family members say that they would like more regular digital interaction with their family member rather than delegating to a service.

Therefore, we will ask “how often would you like to interact with your older family member?”

We can see above that we don't need to use an actual card: we can just use standard text and Markdown to record these. The format of the card is what helps us to structure our thinking and to document our findings.

Learning cards: capture learnings

Similar to test cards, we can structure the outcomes from our experiments in a format called **learning cards**:

The structure for these is pretty clear: it reflects the language in the test card:

Each learning card has four parts:

- **“We believed that ... “** is the hypothesis that we stated on the test card.
- **“We observed that ... “** is the outcome, derived from the data that we collected.
- **“From this we learnt that ... “** specifies exactly what was learnt, including things that we learnt that were not part of the original hypothesis.
- **“Therefore, we will ...”** determines whether we will accept/reject the hypothesis, and/or collect further data for this hypothesis, and/or investigate some new hypotheses.

Example – Medical alert system learning card: **users**

We believed that our end users are people over 65 who live on their own who would like to interact digitally with their family more often because they do not see them much.

We observed that 44 out of 50 people we interviewed confirmed that they would like more regular digital interaction with their family.

From this we learnt that our end users would consider using our product.

Therefore, we will accept this hypothesis.

Example – Medical alert system learning card: **paying customers**

We believed that close family members would like to interact digitally with older family members using a photo-sharing app.

We observed that 12 out of 50 people we interviewed confirmed that they would use such an app.

From this we learnt that family members would not use the app. **However**, we learnt that some family members would send photos from their smartphones to such an app.

Therefore, we will reject this hypothesis, and test a new hypothesis about how family members would take and send photos for their older family member.

Note in the above example that at some point during the interviews, some additional data (about using smartphones) was found. We haven't yet tested this hypothesis formally – for now, it becomes a new hypothesis (which we eventually did test).

Types of experiments

Experiments gather evidence to transform hypotheses into facts – or into refuted hypotheses.

There are several ways to collect evidence:



Interviews

Talk with potential customers



Observations

Observe people while doing their 'jobs'



Data collected by others

Find existing data that tests your hypothesis



Immersion

Working in your customer's environment



Low fidelity prototyping

Paper prototypes or low-fidelity digital prototypes.



High-fidelity solutions

Implementing new functionality in an existing product

Types of experiments

Tips for designing and executing experiments

Experiments should:

- Be as small and comprehensible as possible
- Have clear expected outcomes.
- Have a clear definition of success/failure, which everyone agrees to.
- Be as brief as possible when interviewing to collect data; e.g. preferably less than 30 minutes.

We can test multiple hypotheses at once; e.g. in a single interview or prototype session, but we must do this carefully. We really want to focus on just a few things in each experiment.

We need to prepare for the experiment: research the market and other competitors, research a business (if they are a customer), learn about people in the segment using public data, find out a bit about an interviewee, etc.

Remember that we should:

- **Listen!**
- Observe people's behaviour, tone, and reactions, not just what they say.
- Ask "why" and "why not"
- Ask who else we should talk to.
- Ask if we can follow up with them later.

Remember that we should NOT:

- "Correct" people (except to clarify if they have misunderstood a question).
- Convince people or sell to them.

Learning from experiments

Once we have executed an experiment, we need to gain insights.

If hypotheses have been supported, then great!

If not, what insights can we gain?

- If people would not pay for this, what would they pay for?
- If these are not our customers, then who is?
- Did we see any outliers that could open up new ideas?
- Pay attention to why our hypotheses are rejected.

What type of insights would we gain? It can be insights about:

- Our target market.
- Our value propositions.
- Our market fit.
- Our product features.

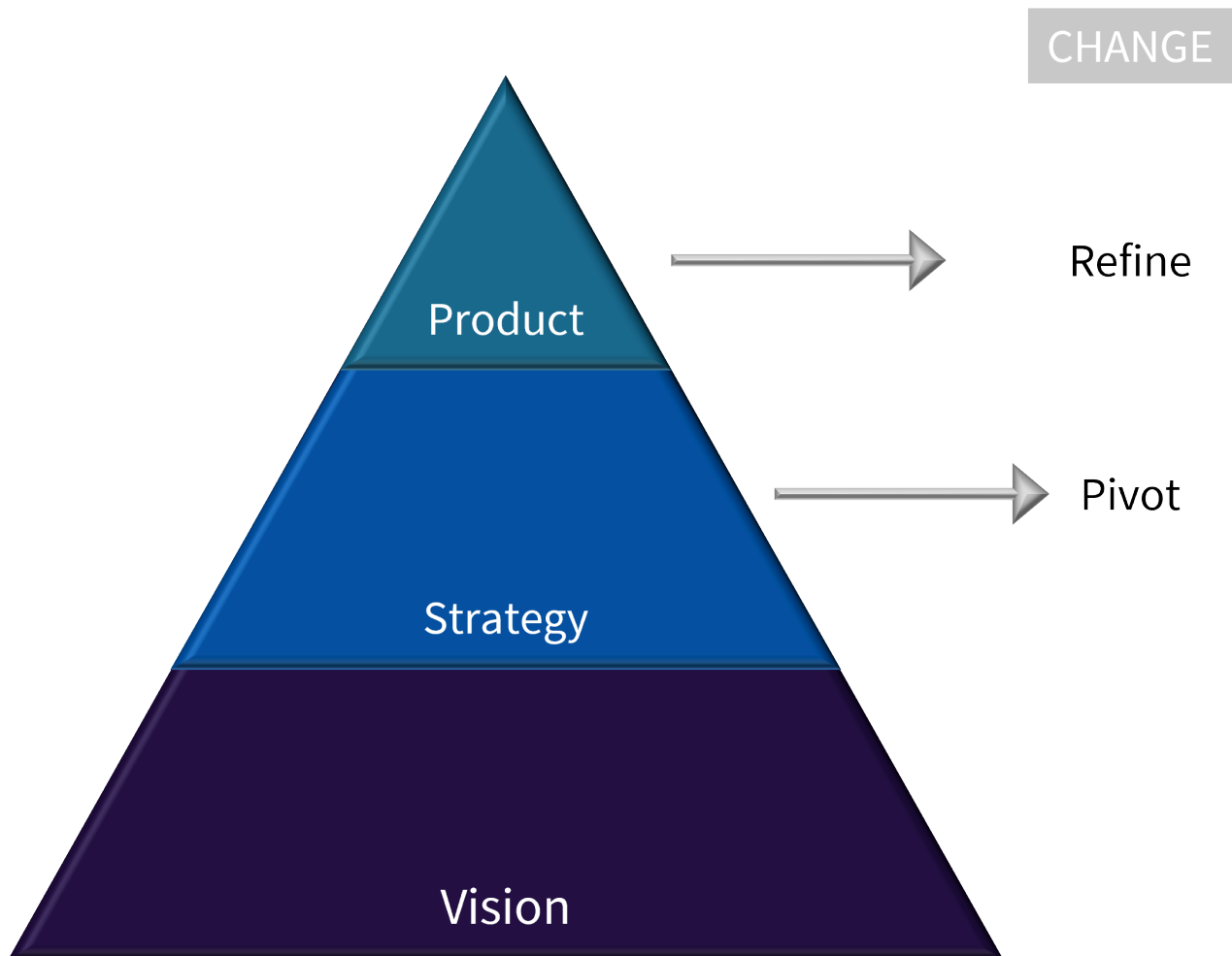
Accept, pivot or refine

Once we have gained some insights, recall that there are a few things we can do for each hypothesis that we have:

- accept the hypothesis as a fact;
- refine our hypothesis; or

- pivot our product entirely — for cases where it appears our value proposition does not have a good market fit.

Recall [Eric Ries's model for innovation](#) from [the introduction](#), re-produced below:



Pivoting vs. refining when we learn our ideas won't work. Adapted from “The lean startup: How today's entrepreneurs use continuous innovation to create radically successful businesses” by Eric Ries (2011) [\[1\]](#).

In cases where a pivot is required, this typically indicates a setback – we have rejected some hypotheses and have concluded that we do not have a market fit.

We can react to a setback in three ways:

1. Ignore what our testing showed us and press on with our idea. This is commonly done; but also, note that businesses and products commonly fail, and this is a key reason why. Why do people press on despite the facts telling them they are wrong? Because it is difficult to give up on an idea that we have invested so much time and effort on.
2. We can **refine** our product ideas. Typically, we can have a good understanding of our customers' problems, but our product idea did not quite solve their problem; or at least, not enough that they would buy it or invest in it. However, we can fine-tune our product ideas, prototype these, and test them again.

3. We can **pivot** our strategy. This happens when we have fundamentally misunderstood what our customers' problems were or what we thought could help solve them. Again, we use what we learnt from our tests to improve our understanding and come up with new ideas.

Ries^[1] notes that it is unusual for organisations to change their **vision** – this is usually the dream to which they are committed, and often the insights gained from data are sufficient.

Chapter summary

The key takeaways for this chapter are:

1. **Experiments** help us gain insights about our customers and products by testing our hypotheses.
2. Hypotheses must be **falsifiable** to be of value.
3. **Test cards** are a useful tool to structure our hypotheses.
4. **Learning cards** are a useful tool to structure our learnings from the data.
5. If a hypothesis is refuted, we can still gain insights by asking “why?” and paying attention to the data we gathered.

References

[1] Ries, Eric. *The lean startup: How today's entrepreneurs use continuous innovation to create radically successful businesses*. Crown Currency, 2011.

[2] Popper, Karl. *The logic of scientific discovery*. (2nd ed.) Taylor and Francis, 2002.

PART IV

BUILDING THE SOLUTION

7.

PROTOTYPING

The purpose of prototyping is to test ideas about solutions. These ideas can be:

- **Conceptual:** testing high-level conceptual ideas about solutions
- **Technical:** testing technical ideas about solutions, such as feasibility

Types of prototype

There are many different types of prototype, both in software and in hardware. For example, [Craig Borysowich](#) lists several different types of prototype in software, including conceptual prototypes, feasibility prototypes, functional storyboards, horizontal prototypes, and vertical prototypes.

In this book, the main two prototypes that we discuss are **conceptual prototypes** and **feasibility prototypes**.

Conceptual prototype

Definition — Conceptual prototype

A **Conceptual prototype** is a layout of a solution that shows the key elements that will be created.

Typically done in two ways:

1. **Paper prototypes:** drawing the ideas on paper/cardboard to demonstrate conceptual ideas
2. **Digital prototypes:** (AKA wireframes): digitally created and look more like a real application.

The purpose of a conceptual prototype is to **fake it before you build it**.

They are an inexpensive approach for testing out ideas without building something, and are key tools Human Computer Interaction (HCI).

Typically, early efforts give us more value because they **clarify ideas and uncover insights**.

Low- vs. high-fidelity conceptual prototypes

We need to consider the fidelity of the experience vs. the fidelity of the prototype. Conceptual prototypes are quick sketches of the system, so they are low fidelity compared to the final product, but provides a high fidelity experience to the customers we test on.

Low fidelity	High fidelity
Show:	Moving towards UI design territory
<ul style="list-style-type: none"> • Major navigation elements • Major content elements 	<ul style="list-style-type: none"> • Closer to what the final version would look like • Include navigation as click through
Don't really show:	Realistic data and images are important
<ul style="list-style-type: none"> • Colours • Images • Meaningful content • Accurate timing 	Useful only as we are finalising solutions

It is the fidelity of the experience, not the fidelity of the prototype, sketch, or technology that is important for ideation and early design. The earlier we prototype, the more valuable the insights tend to be.

Paper prototypes

Paper prototypes are conceptual prototypes that are quick “drawings” of a solution that allow us to test our ideas with customers.

They are typically used to explain or simulate a **scenario** of how the system would be used.

When we use a paper prototype, we show them to customers and then:

- Show or ask them to perform a particular task or scenario
- Encourage them to talk aloud about what they are understanding and what they think is missing
- Ask them to explain their intentions
- Ask them questions
- Take notes about what we are learning!

We can also use paper prototypes for generating ideas (ideation) — they allow us to quickly explore a concept. We can show them to friends, family, others in our team, etc., get comments, and **feedback to incorporate on the spot**, iterating a new version immediately.

Digital prototypes

Digital prototypes are a natural next step after paper prototypes. They are used quite extensively in user experience design.

Like paper prototypes, digital prototypes can be used to show navigation to prospective customers/users to ensure expectations are met.

Unlike paper prototypes, this enables people to interact with a solution, and is higher fidelity than paper prototypes.

The fidelity of the experience that the user gets is dependent on how closely the digital prototype resembles a complete MVP product.

Wizard of Oz prototypes

Definition — Wizard of Oz prototype

A **wizard of Oz** prototype looks like a real solution, but the underlying functionality is a person responding to user interaction.

Spoiler alert! This definition comes from the famous movie [The Wizard of Oz](#), in which the all-powerful “wizard” is in fact just a sophisticated puppet, controlled by a man behind a curtain. When Toto the dog pulls the curtain back to reveal the man, his charade is revealed.

In the same way, when we prototype systems, we can set up some sophisticated-looking technology, but in the background, the response from user interaction is simply a person.

Note

This does not mean that we *deceive* our customers – we can be perfectly honest that the functionality is not implemented as the responses are given by a person.

As part of prototyping, we can use anything that we want to conjure up experiences for customers.

*“It is much easier, cheaper, faster, and more reliable to find a little old man, a microphone, and some loud speakers than it is to find a real wizard. So it is with most software products. Fake it before you build it.” – Bill Buxton. *Sketching User Experiences*, p. 239.[\[1\]](#)*

For example, instead of building a complete mobile app to send photos, a server to process photos, and a

photo frame to display and interact with photos, we can simply implement some interfaces, and move the photos onto a tablet manually when our users are testing it.

Practical tips for conceptual prototypes

Here are some practical tips for conceptual prototypes:

- Keep them simple – they eventually will be thrown away
- Use a grid to ensure that everything is well aligned
- Add meaningful and concise annotations – people refer to it all the time
- Encourage feedback
- People often don't tend to criticise designs
 - "I'm a negative person"
 - "That's a lot of work, I don't want to be rude"
 - "I'm not good with computers"
 - "Responsibility for bad performance will fall on me"It is harder to criticise something that looks polished.
- Beware of tools for prototyping unless you want a high-fidelity, polished look

Feasibility prototypes

Definition — Feasibility prototype

A **feasibility prototype** explores the technical constraints of design solutions, and are useful at early product development stages.

Feasibility prototypes are both useful and important for innovation, but are not useful until we have found our customers, their problems, and have validated that our solution solves their problem. At that point, we use feasibility prototypes to test out technical solutions, and try them with customers.

Chapter summary

The key takeaways for this chapter are:

1. Prototypes are used to help **test ideas**.
2. **Conceptual prototypes** help us test conceptual ideas, while **technical prototypes** help us test technical approaches.
3. Prototyping is an **iterative process**.
4. **Paper prototypes** and **Wizard of Oz** prototypes are cheap and quick ways to test out conceptual ideas and gain feedback.

5. **Digital prototypes** are more useful for testing out solutions and workflows.
6. **Low-fidelity** prototypes are often more useful than **high-fidelity** prototypes to gather feedback because they are not so polished, so people are more prone to criticising.

References

Buxton, Bill. *Sketching user experiences*. Morgan Kaufmann, 2007. <https://doi.org/10.1016/B978-0-12-374037-3.X5043-3>

8.

MINIMUM VIABLE PRODUCT

“By far the dominant reason for not releasing sooner was a reluctance to trade the dream of success for the reality of feedback.” – Kent Beck in [Approaching a Minimum Viable Product](#)

Definition – Minimum viable product (MVP)

A **minimum viable product** (MVP) is a version of a new product that contains the fewest number of features (or least amount of effort) required to collect feedback about customers.

Purpose of the MVP

Quiz – Purpose of an MVP

A quick quiz question for us all: *What is the primary purpose of a minimum viable product (MVP)?*

1. To test the ability of product to meet minimal customer needs.
2. Generate as much revenue as we can as early as we can.
3. Identify bugs in product with a fixed feature set that meets a set of requirements.
4. All of the above.

Well, what did you answer?

All of these can be useful to us, but item 1 is the **primary** purpose of an MVP. Let's discuss.

The **primary** purpose of an MVP is to test our hypotheses.

If we test a hypothesis about market demand and this test *also results in us gaining revenue*, this is great, as it provides vital funding to continue with our efforts. However, our priority here is to build a valuable product in the long run. Knowing whether we have a product that meets customer needs *in the longer run* is more valuable during customer discovery than building a product that may not be suitable, but gets a few early wins. Of course, early wins are great, but they don't necessarily sustain us.

Identifying bugs in a product is also good, but as above: it is not as valuable at this early stage as testing whether our product meets customer needs.

“You’re selling the vision and delivering the minimum feature set to visionaries, not everyone.” – Steve Blank [2]

Principles for an MVP

Principle 1: KISS (Keep it simple, stupid!). Do not add “bells and whistles” – just add enough features to test the hypotheses that you want to test.

Principle 2: Fast to build. We should spend weeks, not months, to build an MVP.

Principle 3: Positive user experience. Although it is simple, make an MVP a positive experience for the people we test it on. It should be usable, simple, and intuitive – remember that these people are our future customers!

Principle 4: Early adopters. An MVP should appeal primarily to early adopters.

Principle 5: Iterate and Refine. Pay attention to user feedback and be prepared to refine the product and iterate multiple times. It is rare to get everything right in even the first few versions.

Principle 6: Set success metrics. An MVP is still used to [test hypotheses](#), so set metrics for those hypotheses. It is not just something for users to play with.



Minimum viable product.



Product vision.

Keep the MDP simple!

Example – Dropbox

[Dropbox](#) founder Drew Houston famously used a marketing video to demonstrate what Dropbox *would*

do, before it was built, as a way to test whether there was demand for this product. He asked people to sign up and pay, and received 70,000 sign ups on the first day, which gave him data to secure funding from investors.

Watch [this excellent video \(YouTube, 16m 52s\)](#) from Michael Seibel, partner at [Y Combinator](#) and co-founder of [Twitch](#) discuss how to build an MVP:



One or more interactive elements has been excluded from this version of the text. You can view them online here: <https://uq.pressbooks.pub/introduction-software-innovation/?p=76#oembed-1>

Seibel lists three important properties of an MVP:

1. **Fast to build.** We should spend weeks, not months, to build an MVP.
2. Limited functionality. We should implement just enough functionality to test out our hypotheses.
3. Appeal only to early adopters. Focussing on the earlier.

Example – Airbnb

In the [video from Michael Seibel \(YouTube, 16m 52s\)](#) that we reference above, he cites the example of Airbnb — a website for finding holiday accommodation rented out by individuals. Seibel notes that the original Airbnb MVP:

- Did not accept payments — once a match was made, the owner and customer had to sort out payment themselves.
- Did not have a map view — customers couldn't see a list of available properties on a map view; it was just a list.
- Insisted that owners rented out rooms with an air bed only!
- Was only set up for conferences — that is, the Airbnb creators would find cities hosting large conferences, set up a site, and remove it again once the conference was over.

However, this Airbnb site was crucial for testing out the founders' initial hypotheses, which were presumably about whether a website that allowed people to list holiday accommodation would attract paying customers.

I still don't understand the requirement of only giving air beds – perhaps that was an early marketing trick.

Medical alert system – In Touch

To fulfill the ‘keep in touch’ idea for our cohort of older people and their families, we implemented a simple photo-sharing app to be run on a tablet. We recruited some customers from our initial interviews.

Family members each had an email address that they could send photos to, allowing them to capture photos of their daily life; e.g. photos of their children (the grandchildren of their older family members). This allowed us to use existing mobile applications on the family side, so came at little cost and was easy to implement.

The older family members received a tablet with a small stand and were asked to place this in a visible place, much like a photo frame. The tablet would show photos that have been sent. Any time new photos were sent, these would be displayed. If no new photos were sent on a day, a previous photo that the older person had liked would be shown. However, to access the photos each day, the older person needed to touch the screen, where there was a notification letting them know a new photo was ready. When this occurred, a nominated family member would receive a message, thus letting them know their family member was ok.

This allowed us to test the hypotheses that both the older people and their family members would stay in touch using this type of device.

Note that this particular MVP did not allow the older person to send an emergency alert, nor did it enable mobility.

However, it took just a couple of weeks to implement and ran on standard tablets and smartphones, enabling us to test the idea. Further, as this was part of a research project, we had functionality that collected data about interaction frequency between family members (with ethics approval and consent!).

Defining features for MVPs

Once we have a basic idea for an MVP, we need to break it into a set of features. This helps us to gain feedback from our customers, as we can identify some concrete things our product will do. But importantly, it also helps our team to break work up among members and to prioritise features.

We want to act fast, so we should keep feature definitions simple and minimal. The following format is sufficient for most projects of a small team of 2-5 people:

1. **Feature name.**
2. **Description:** What the feature will do at a high level.
3. **Who:** The person responsible for implementing the feature. If you need more than one person, break into small features.
4. **When:** The deadline for feature delivery.
5. **Priority:** The priority of this feature; e.g. using [RICE](#), [MoSCoW](#) (Must Have, Should Have, Could Have, Will Not Have), or a ranking from 1 to N.
6. **Value:** A justification for the priority of this feature, using data from learning cards.

7. **Dependencies:** Features that must be completed before this feature can be released. NOTE: This does **not** mean features that must be completed before we can start implementing this feature: we can start implementing a feature and e.g. send dummy data. Dependencies should never block development; only release.

Example – Defining MVP Features for *In Touch*

Feature 1: Photo Submission via Email

Description: Allows family members to send photos to the system using their existing email applications.

- Parse incoming photos
- Associate photos with corresponding user

Who: Imogen

When: Wednesday 20th May 3:00pm

Priority: 4

Value: Essential for receiving photos. Less important than displaying fallback photos as this feature is not useful if photos cannot be displayed.

Dependencies: Must be completed before Feature 2 and 3 can function meaningfully.

Feature 2: Photo Display on Tablet

Description: Displays received photos on a tablet placed in the older person's home.

- Full-screen photo display
- Automatic refresh when new photo arrives

Who: Zac

When: Wednesday 27th May 7:00pm

Priority: 5

Value: Less important than feature 1 (photo submission) because it can still operate with a batch of photos.

Dependencies: Works in conjunction with Feature 3 to determine which photo is shown.

Feature 3: Fallback Photo Display

Description: Displays a previously uploaded photo if no new photos have been received that day.

- Store “liked” photos
- Select fallback image automatically

Who: Zac

When: Week 9 Wednesday 27th May 7:00pm

Priority: 3

Value: Provides value if photos are not sent; e.g. with bulk uploaded photos.

Dependencies: Works in conjunction with Feature 2, using the same display mechanism.

Feature 4: Interaction Prompt

Description: Requires the older person to touch the screen to access photos each day and acknowledge they have seen the photo.

- Notification indicating a new photo has arrived
- Simple, low-precision interaction to view the image and acknowledge they have seen it

Who: Joshua

When: Week 9 Wednesday 27th May 7:00pm

Priority: 1

Value: Core mechanism to signal wellbeing, so highest priority.

Dependencies: Works in conjunction to Feature 2 to trigger display image

Feature 5: Interaction Notification to Family

Description: Notifies the family member when the photo is accessed.

- Trigger notification on photo access
- Send email to nominated contact

Who: Amelie

When: Week 10 Monday 1st June 12:00pm

Priority: 2.

Value: Addresses a key pain point of families wanting assurance.

Dependencies: Relies on Feature 4, as notifications are triggered by user interaction.

Types of MVP

Pre-Order: e.g. [Kickstarter](#). As in the Dropbox example above, asking customers to pay for a product before it is made is a way to test market demand. Kickstarter campaigns that don't reach a pre-set fundraising target don't go ahead, and customers get a full refund.

Audience Building: e.g. Social Networks. By setting up a strong social media presence, we can track whether people will be interested in our product.

Show and Tell: Explainer Video. Again, the Dropbox example above is an example MVP – so simple, yet so effective.

Partial Product: Perhaps the best MVPs are working products, perhaps with just some (not all) of the features of a product – the most important ones (minimum viable!) – such as landing-page, single-feature websites. In the software world, these can be cheap and easy to build for many MVPs, and we can then release an MVP to early adopters to test the market.

MVP vs Prototyping

In terms of partial products, a minimum viable product is a **product**, not a prototype. That is, it must be something that we can sell/give to early adopters that they can use to solve their problems. A prototype designed using wireframes or even paper prototypes are not products.

While [software prototypes are useful](#) for gaining feedback about the features, look and feel, etc., of an MVP, ultimately they are not products. For example, tools such as [Figma](#) can be used to create wireframes, which can be used to gain concrete feedback, they are not products that customers can use to solve their problem. Consider the emergency alarm system, discussed above and [in the earlier chapter](#). A wireframe prototype can be shown to potential customers to gain feedback about the design, features, etc. However, they cannot download this and use it to stay in contact with their families, so it is not a product. Ultimately, using a prototype, **we cannot gain feedback about important parts of our value proposition**, such as whether it gives older people the independence/autonomy that they crave; or whether their family members get a sense of whether the older person is safe; or whether either will pay for it!

Below, we compare prototypes and partial-product MVPs, across five dimensions:

Prototype	Product
Aims to test and refine test design ideas	Aims to test market fit
Visualises core product ideas	Implements core product features
Often not actually functional	Must be functional (even if minimal)
Super fast to build and test	Requires working code, deployment, etc. so slower to build
Ideas tested in a ‘controlled’ environment	Released to the market

The final point is perhaps the one that makes it easy to tell whether something is a partial-product MVP: if we are testing our ideas in a ‘controlled’ environment; that is, showing a customer features and gaining feedback, then it is a prototype. On the other hand, if a customer is installing an application or accessing a web application and using it to help solve their pains, it is a product.

Chapter summary

The key takeaways for this chapter are:

1. A **minimum viable product** (MVP) has just the most important features to test our main hypotheses.
2. We should keep them **simple, iterate**, and if possible, **return to customers** to test refinements.
3. Different MVPs have different purposes, based on the hypotheses we test with them.
4. Prototypes are typically not MVPs.

References

[1] Beck, Kent. *Approaching a minimum viable product*. 2009.

[2] Blank, Steve. *Perfection by subtraction - The minimum feature set*. March 4, 2010.
<https://steveblank.com/2010/03/04/perfection-by-subtraction-the-minimum-feature-set/>

9.

CUSTOMER RELATIONSHIPS AND CHANNELS

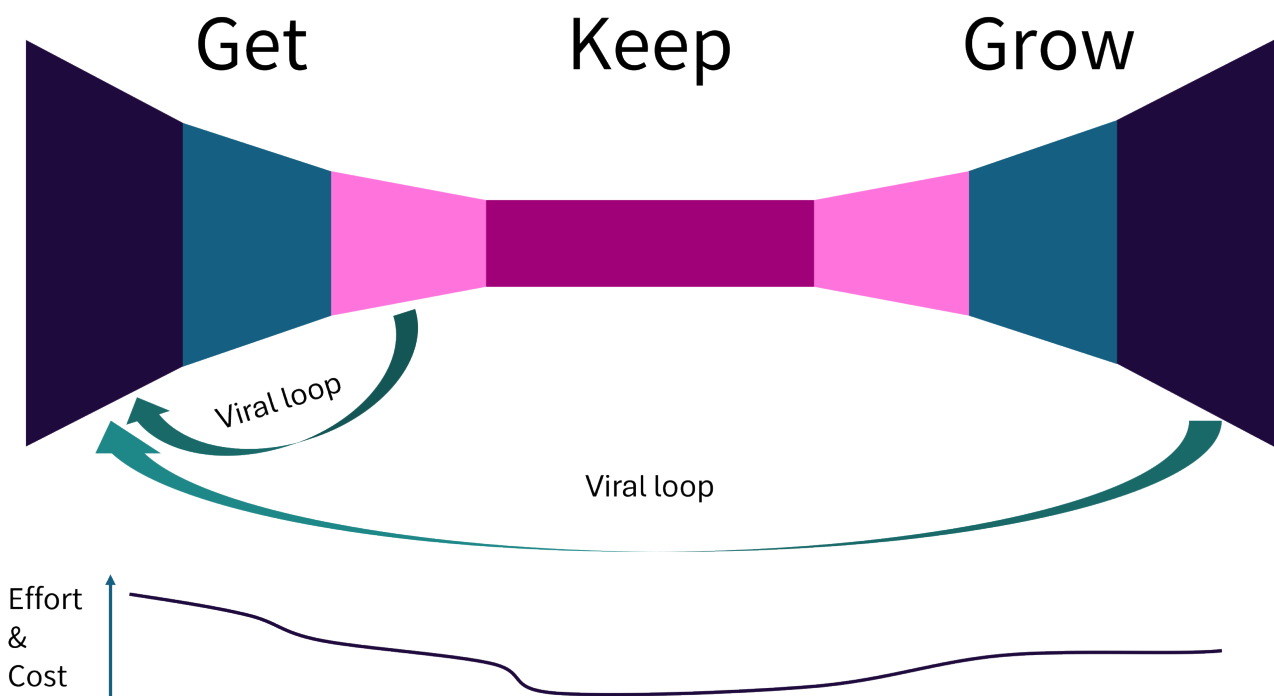
Recall from the [chapter on the business model canvas](#) that the customer relationship and channels are the links between our value propositions and our customer segments:

- **Channels:** the mechanisms by which solutions are delivered to customers.
- **Customer relationships:** the mechanisms that are used to engage and build trust between organisations and customer segments.

Customer relationships: Get, keep, and grow

The figure below outlines the three main stages to gain customers via customer relationships:

1. **Get:** bring initial customers into our product; e.g. early adopters.
2. **Keep:** keep the customers that we have.
3. **Grow:** grow into new customers; e.g. the early and then late majorities.



An overview of the ‘get, keep, grow’ model. Adapted from “[How to raise money – It’s a journey not an event funnel](#)” by [Steve Blank\[1\]](#).

Typically, getting the initial customers takes a lot of cost and effort. Keeping them is less effort and cost, provided we have a good product. Growing is more costly than keeping, but if we have a good product and this spreads via word of mouth from existing to new customers, this growth can be sharp. In fact, this can often cause problems for startups as they cannot keep up with the new demand. While this is less of a problem for software applications than say, physical products, it can still be difficult to e.g. find the necessary computational resources, or find the staff to provide customisations for particular customers.

The two viral loops indicate situations in which ‘getting’ and ‘growing’ happen with minimum effort from us: our customers using the product raises awareness with new customers. For example, customers telling other people about the product; or if our customers are businesses using our product to serve their own customers, their customers may see our product, raising awareness.

Foundations for customer relationships

Customer relationships are important for getting, keeping, and growing our customer base. This includes both users and paying customers.

If we want strong customer relationships, we need to be able to answer the following questions:

1. Who are our customers?
2. What is their motivation for using our product? What matters to them?
3. Where do they get their information about products?
4. What (or who) influences them?
5. How do they buy things?

Hopefully, our customer discovery process has given us a clear idea of the first two questions.

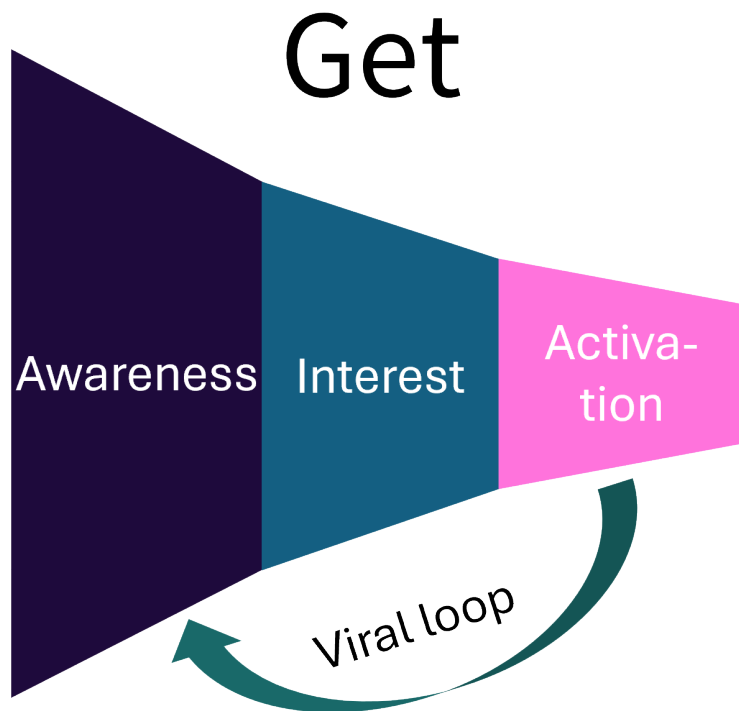
The latter questions can be framed as hypotheses and tested, in the same way that we frame and test hypotheses about customer segments and value propositions.

However, we can see that different types of customers have different needs. An older person using a medical alarm is less likely to use social media to research products than younger users of educational apps. Similarly, they are influenced by different people, and buy things differently. For example, younger demographics are much more likely to buy on their mobile device, while generation X would be more likely to use a laptop/desktop computer for large purchases. Older people may prefer to purchase things in physical stores — although this preference is starting to reduce as everyone becomes more digitally literate.

Example: A web and mobile product

Getting customers

Consider the example of a product we are releasing on the web to be used on desktop and mobile devices. What does the ‘get’ phase look like?

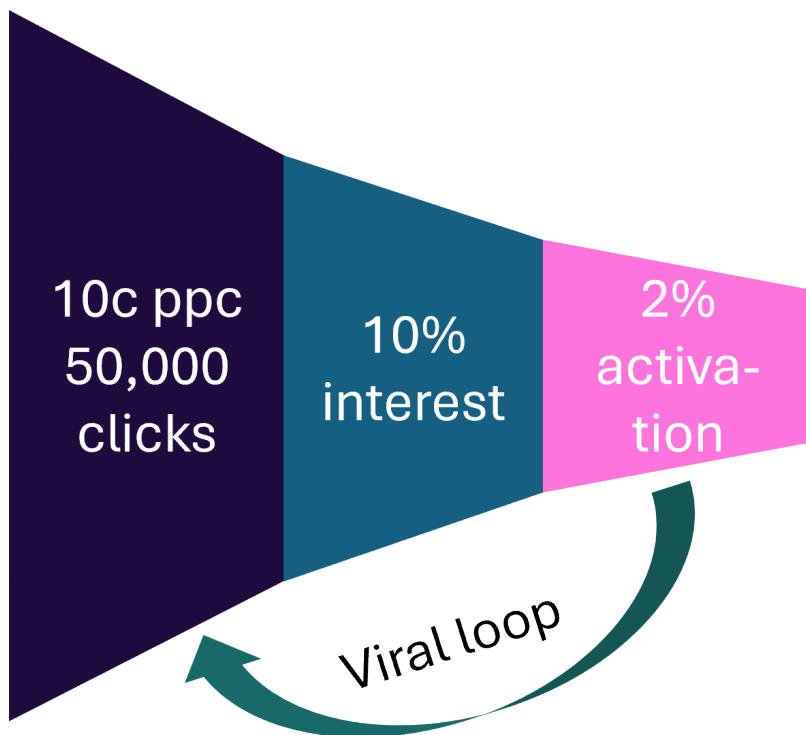


An example strategy for the ‘get’ stage of a web and mobile application. Adapted from “[How to raise money – It’s a journey not an event funnel](#)” by [Steve Blank\[1\]](#).

The figure above shows the strategies we could take:

1. **Awareness:** this represents the initial stage where customers become aware of a product or service.
2. **Interest:** the middle stage where those who are aware of the product or service show interest in it.
3. **Activation:** the third stage where interested individuals take action, such as signing up, purchasing, or engaging further.

How do we implement these strategies?



An example of the 'get' stage of a web and mobile application. Adapted from "[How to raise money – It's a journey not an event funnel](#)" by [Steve Blank\[1\]](#).

The figure above outlines some approaches we could use:

1. Pay for Internet advertising at 10c per click, at a maximum of 50,000 clicks.
2. 10% of people who see the advert show some interest.
3. 2% of people who see it activate by signing up to our product.

There are other ways to raise awareness. Some strategies are **paid** demand creation:

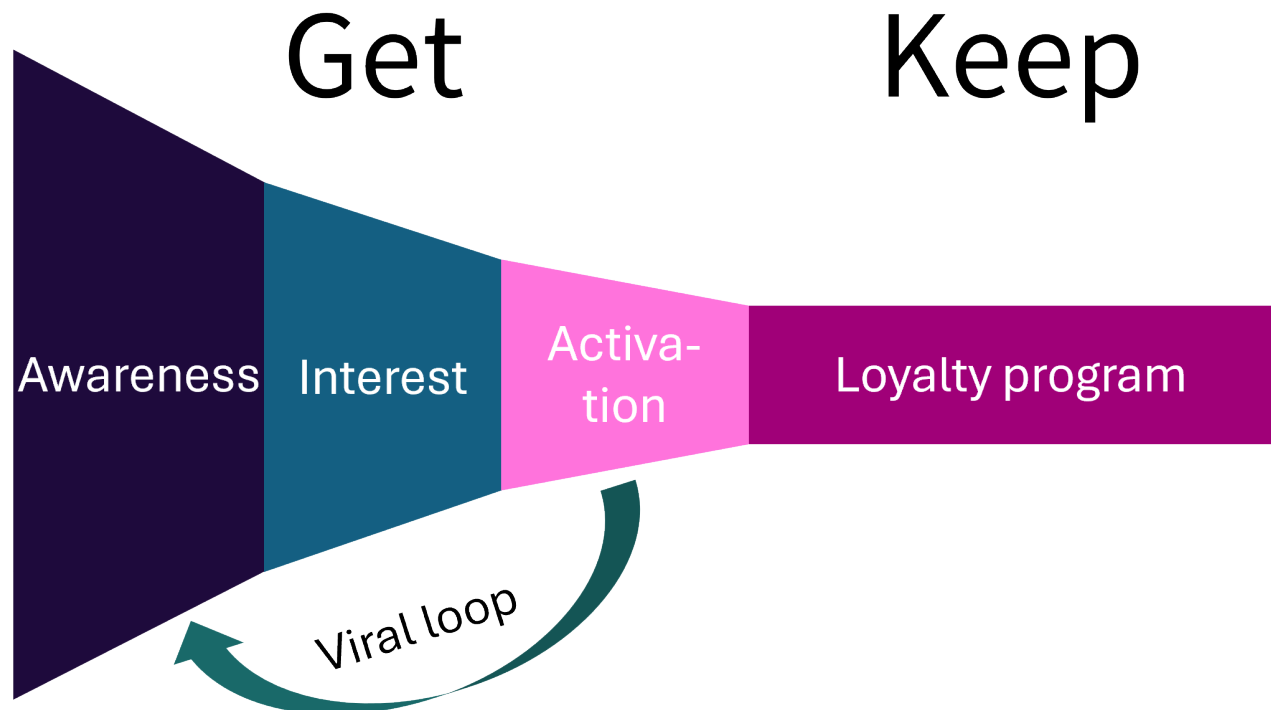
1. Use a public relations agency.
2. Attend trade shows.
3. Use email campaigns.
4. Social media engagement.

We can also use **earned** demand creation:

1. Publication in peer-reviewed journals.
2. Conference appearances.
3. Blogging and guest articles for media.
4. Search Engine Optimisation.
5. Social media presence.

Keeping customers

As noted, if we have a good value proposition, keeping customers is lower cost and effort than getting them initially. In short: the customers we have are easier to keep than acquiring customers we don't have. The reason is that these customers have already acted on their problem, so they have signalled clearly that they are in our customer segment.



An example strategy for the 'keep' stage of a web and mobile application. Adapted from “[How to raise money – It’s a journey not an event funnel](#)” by [Steve Blank\[1\]](#).

There are several strategies that we can use to keep customers:

1. Loyalty programs.
2. Product updates to keep their interest and continue adding value.
3. Contests and events to keep engagement.
4. Horizontal and vertical offerings.
5. Customer engagement and satisfaction programs; e.g. blogs, community support, email.
6. Social media.

However, this leads to a question: **Why should we focus on ‘keeping’ existing customers that we already have, instead of gaining new ones that bring new income?**

The answer is: it is more expensive to get new customers than keep existing ones.

This is known as **churn**, and it comes at a cost.

For example, consider monthly churn of 1%, 5%, and 10%. After 3 years, how many of the original customers will be left?

We can calculate the remaining customers as:

$$\text{Remaining percentage} = (1 - \text{Monthly attrition rate})^n,$$

where n is the number of months.

Monthly customer churn	Remaining customers after 3 years
1%	$(1 - 0.01)^{36} = 69.8\%$
5%	$(1 - 0.05)^{36} = 16.6\%$
10%	$(1 - 0.10)^{36} = 3.14\%$

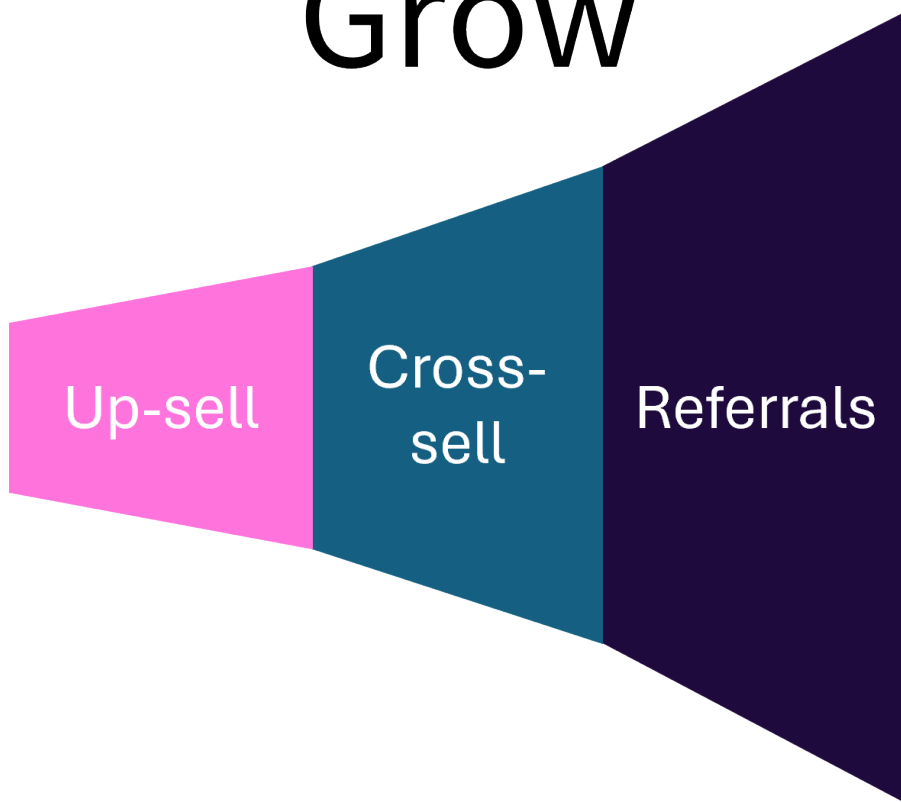
This is a massive difference: reducing attrition from 5% to 1% will result in over four times as many of our original customers, and from 10% to 1%, over 22 times as many.

Similarly, we can calculate the average duration that a customer stays with different churn rates:

Monthly customer churn	Average Duration (Months)
1%	$\frac{1}{0.01} = 100$ months
5%	$\frac{1}{0.05} = 20$ months
10%	$\frac{1}{0.10} = 10$ months

Growing customers

Grow



An example strategy for the ‘grow’ stage of a web and mobile application. Adapted from “[How to raise money – It’s a journey not an event funnel](#)” by [Steve Blank](#). [1]

To grow customers with our web and mobile product, we can use several different tactics, such as:

1. Un-bundling: split products into several smaller products.
2. Freemium: give away a basic product at no cost, but a premium product that customers pay for.
3. Up-sell and next-sell (vertical): one base product with multiple brands and features.
4. Cross-sell (horizontal): companion products to go with a product.
5. Referral programs: give incentives for customers to introduce new customers.

Recall from the [‘get, grow, keep’ model](#) above that good strategies for growing can result in viral loops and grow further – which can be great (more income) and painful (the demand can grow quicker than our ability to service it).

Channels

With regards to distribution channels, there are two main types: **direct** distribution and **indirect** distribution.

Direct Distribution Channels

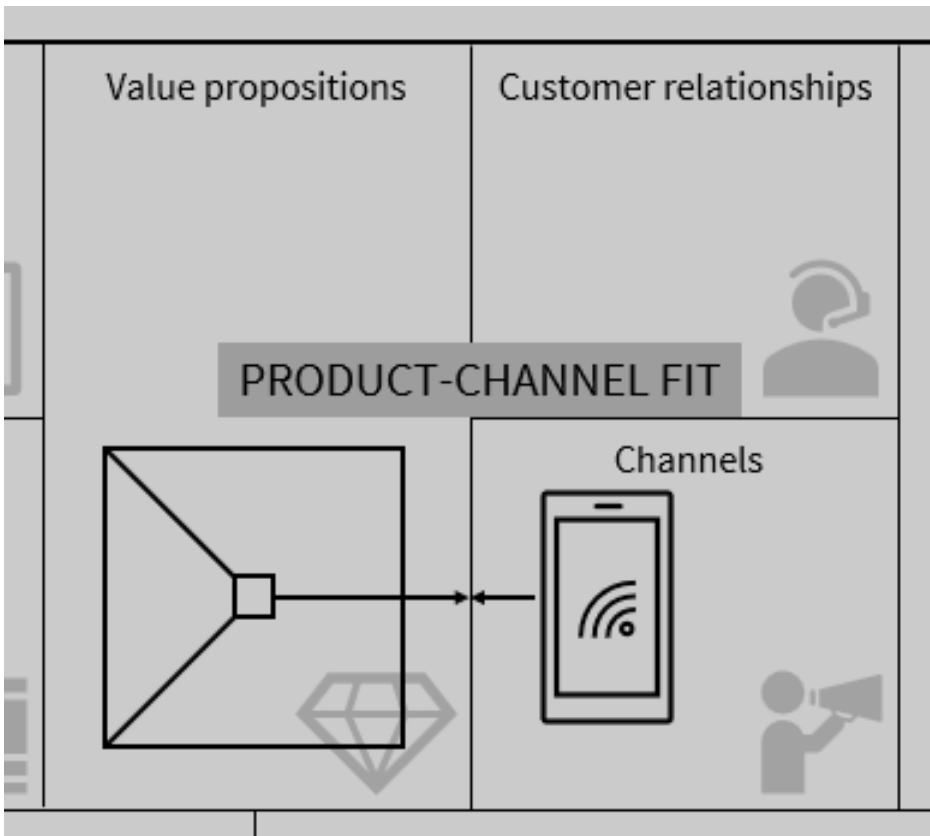
1. **Dedicated e-Commerce:** This involves selling products or services directly to customers through a company's own online platform, such as a dedicated website or mobile application. This channel allows for direct customer interaction and control over the sales process.
2. **Social Commerce:** This refers to the use of social media platforms to promote and sell products directly to consumers. Social commerce leverages the reach and engagement of platforms like Instagram, Facebook, or TikTok to drive sales directly within the app or by linking to external e-commerce sites.

Indirect Distribution Channels

1. **Marketplace (e.g., App Store):** Selling products or services through third-party platforms like an app store (e.g., Apple's App Store, Google Play, Microsoft Software Centre). These provide access to a large customer base but usually involve sharing revenue or paying fees to the platform provider.
2. **Two-Step Distribution:** This involves selling products through intermediaries, such as wholesalers or retailers, who then sell the products to the end customers. Common in industries where large-scale distribution networks are necessary to reach a broad audience.
3. **Aggregators:** These are platforms that gather offerings from multiple providers into a single platform (for a price), making it easier for customers to compare and purchase products. Examples include travel booking sites or food delivery apps.

Product-channel fit

Much like we have a product-market fit in our business model, we should also have a product-channel fit — that is, the product (value proposition) and the distribution channels must fit together, as illustrated below.



Product-market fit. Adapted from “[Business model canvas](#)” by [Strategyzer](#), modified under a [Creative Commons Attribution ShareAlike 3.0 Unported licence](#).

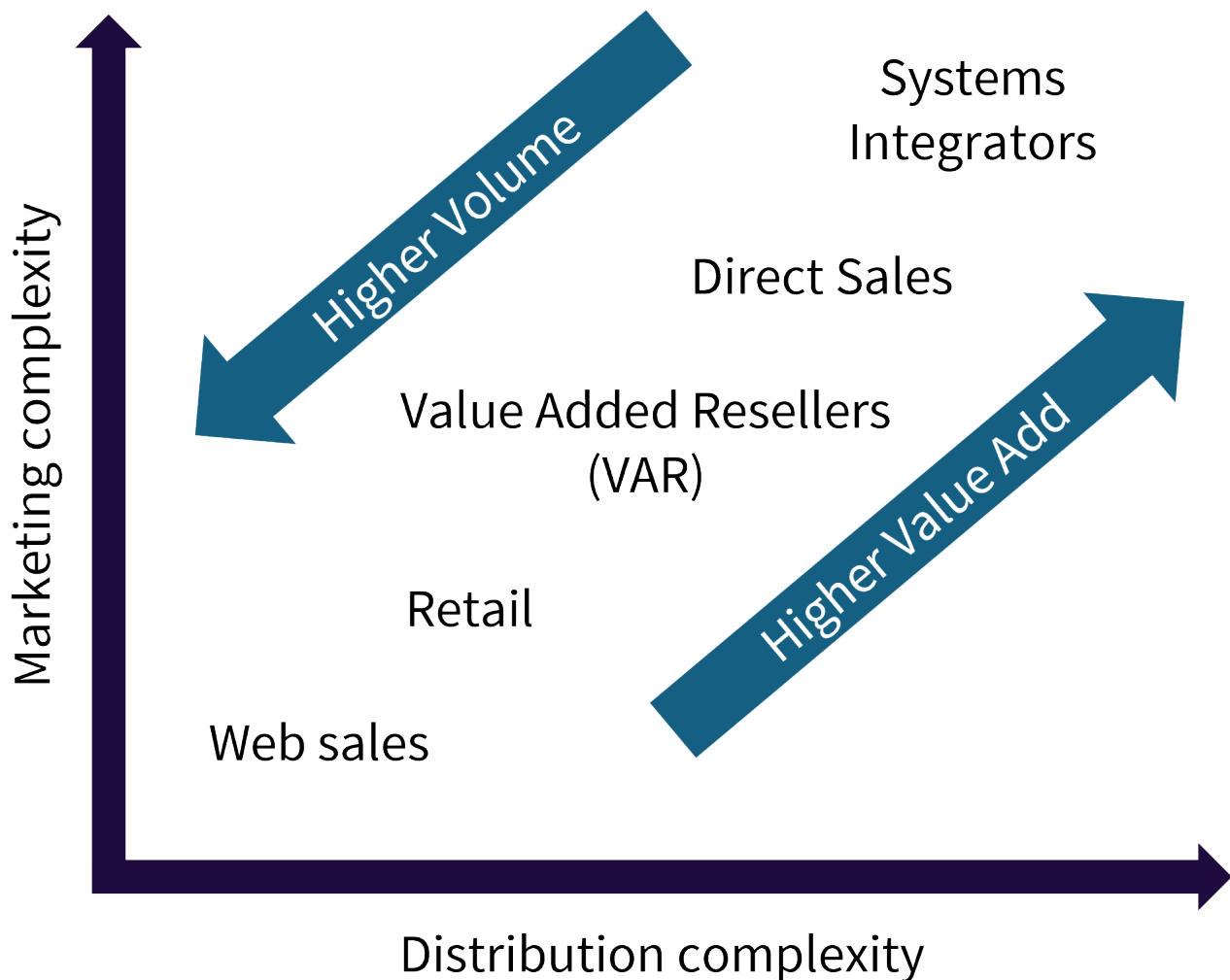
While software was once sold on DVDs, people will not buy mobile apps on DVDs! In fact, they are highly unlikely to download one and install if there is an alternative via an app store. It is sometimes easy to see a product-market fit for software-related solutions; e.g. the easiest way to sell mobile applications is to place them on app stores. However, desktop applications may require a set of different channels, such as direct purchase from a company website, upload to a software centre, via package managers, etc.

	Physical Channel	Virtual (e.g. Web/Mobile) Channel
Physical Product	Cars, Clothes, Books, Groceries	Clothes, Books, Groceries
Virtual Product	PC Games, DVDs	Digital music, Online Games, Podcasts, eBooks

Channel complexity

One final lesson that we can learn from the experience of others, is that there is a relationship between solution complexity and marketing complexity.

The model below illustrates that as distribution complexity increases, often marketing complexity does too. Further, this correlates with a higher value add, and negatively correlates with the ability to scale volume.



Relationship between distribution complexity, solution value, and marketing complexity.

For example:

- Web sales: Simple products that can be easily sold online can usually be advertised online with simple marketing via online ads and social media.
- Retail: This requires a bit more effort to market due to the need for physical presence and in-store promotions.
- Value Added Resellers (VAR): A value-added reseller takes another organisation's product and adds new features/services before reselling; e.g. they sell someone's product and offer support, installation, etc. This often requires more detailed marketing to explain the added value.
- Direct Sales: This involves personalised selling, requiring tailored marketing approaches.
- Systems Integrators: Organisations who integrate other systems often handle the most complex solutions, needing specialised marketing strategies to communicate the benefits of the integration. Often the integration can be personalised to specific organisations/people, meaning much higher complexity in marketing.

Of course, this is not a law of physics: it can be that some simple solutions are difficult to market, and vice-versa, but the trend is common.

What does this mean for our startup product? It is a lesson that keeping the solution simple, especially early on, helps us to get marketing up and running more quickly. This leads to earlier sales to our early adopters and early majority customers.

Customer feedback

Finally, it is important to note that, just like every other part of our customer discovery process, **we should actively seek out input from our customers about our distribution channels and relationship strategies**. This includes testing hypotheses early with e.g. interviews, and analysing customer feedback after release.

Chapter summary

The key takeaways for this chapter are:

1. **Customer relationships** are required to help you **get** customers, **keep** customers, and **grow** our customer base.
2. **Keeping** existing customers is **easier and cheaper** than getting new customers – we shouldn't neglect our existing customers in the race to gain new customers.
3. **Customer channels** allow us to distribute our products. We should aim for a good **product-channel fit**.
4. More complex solutions are often more complex to **market and distribute**. Another reason to **keep it simple** for our [minimum viable product](#)

References

[1] Blank, Steve. *How to raise money – It's a journey not an event*. February 26, 2020. <https://steveblank.com/2020/02/26/how-to-raise-money-its-a-journey-not-an-event/>

OPEN TEXTBOOKS @ UQ

Introduction to Software Innovation is made possible by the Library's Open Textbooks @ UQ program. The Library supports the adoption or creation of open educational resources, such as this open textbook, that are free for everyone. Open textbooks provide an alternative to commercial textbooks and benefit you, your students and the community. Find out how you can [author, adapt or adopt open textbooks](#).

Your feedback is always welcome

Contact pressbooks@library.uq.edu.au if you have questions or feedback about Open Textbooks @ UQ.

GLOSSARY

Business model canvas (BMC)

A **business model canvas** is a visual description of the nine key parts of a business model. It consists of (a mix of) hypotheses and facts about the business model. It can be used to describe existing business models, describe new business models, and to help us focus on the key questions of a new business.

Customer

The individual or organisation that makes the purchasing decision and pays for a product or service. The customer is not always the person who ultimately uses the product. (e.g., in our example of the medical alert system, a family member is the customer, purchasing a service for an elderly relative).

Customer discovery

The process of turning an initial idea or vision about a market or customers into a set of **facts**. The process tests whether the business model for a company solves real customer problems and whether the product features fit those customer problems.

Early adopter

People who typically adopt new solutions before others. They understand that they have a problem, want a solution, usually have an interim solution, and have the funds to purchase a solution. Often, they like to be on the cutting edge.

End user

The individual who directly interacts with and uses a product or service. The end user may differ from the person who purchases it.

Falsifiable

In this context, a statement (hypothesis) that can be tested and refuted using empirical data.

Hypothesis

In the context of innovation, a **hypothesis** is a falsifiable statement or prediction about a business idea, product, or customer that can be investigated through empirical observation and experimentation.

Innovation

Innovation is about **creating new value for someone**.

Minimum viable product (MVP)

A **minimum viable product** (MVP) is a version of a new product that contains the fewest number of features (or least amount of effort) required to collect feedback about customers.

Pivot

A structured change in strategy based on validated learning, while maintaining the overall vision.

Product-Channel Fit

Alignment between a value proposition and the distribution channels used to deliver it.

Product-Market Fit

The degree to which a product satisfies strong market demand.

Value proposition canvas

Value proposition canvas is a visual description of our customer segments, and our value propositions to solve the customers' problems.

Wizard of Oz

A **wizard of Oz** prototype is a solution that looks like a real solution, but the underlying functionality is a person responding to user interaction.