Introduction to Classical and Quantum Computing

Chapter 1 Classical Information and Computation

Introduction

Bits

* + 1. Coins

When a fair coin is flipped in the air and lands on the ground, it takes on one of two states: H (heads) or T (tails) -> {H, T}.

When two coins are flipped the state of both coins is in {HH, HT, TH, TT}.

Three coins {HHH, HHT, HTH, HTT, THH, THT, TTH, TTT}.

To general this, *n* coins form a set with members.

Four coins forms a set with members.

Five coins forms a set with members.

* + 1. Dice

When a fair dice is rolled and comes to a stop, it takes on one of six states: 1, 2, 3, 4, 5, or 6 -> {1, 2, 3, 4, 5, 6}.

When two dice are rolled the state of both dice is in a set that has members.

To general this, *n* dice form a set with members.

Four dice forms a set with members.

Five coins forms a set with members.

* + 1. Encoding Information

The more states an object can represent, the fewer objects you need to encode information. How many coins would you need to encode the seven colors of the rainbow? How may six-side dice would you need encode the seven colors of the rainbow?

Something with two states carries the smallest amount of information possible.

How many coins would you need to encode the 26 letters of the alphabet? You would need 5 coins. How many dice would you need to do the same? You would need 2 dice.

* + 1. Physical bits

Regardless of the physical system used to present two sates (heads/tails, off/on, pits/lands, 0V/5V), it is easier to use the binary digits (bit) 0 and 1.

A bit is the smallest unit of classical information.

* + 1. Binary

If we convert two state physical system to binary, then this affords us the opportunity to use math on the bits and manipulate them.

Binary numbers are base 2 numbers and are also referred to as binary strings or bit strings. An example of a bit string is 11010 or equivalently 110102.

Know how to convert number between different systems – for example binary to decimal, decimal to binary, binary to hexadecimal and so on.

The leftmost bit is called the most significant bit and the rightmost bit is the least significant bit.

Negative numbers can be encoded in binary using two’s complement. In this system, the most significant bit is negative while the remain bits are positive. Understand why 101 in two’s complement is equivalent to the decimal number -3.

* + 1. ASCII

American Standard Code for Information Interchange (ASCII). The original ASCII standard encoded characters using a seven-bit string.

The most common modern standard is UTF-8 (Unicode Transformation Format) that uses 32-bit string with the first

* 1. Logic Gates
     1. Single-Bit Gates
     2. Two-Bit Gates
     3. Logic Gates and Physical Circuits
     4. Multiple Gates
     5. Universal Gates
  2. Adders and Verilog
     1. Adding Binary Numbers by Hand
     2. Half Adder
     3. Full Adder
     4. Ripple-Carry Adder
     5. Ripple-Carry with Full Adders
     6. Circuit Complexity
  3. Circuit Simplification and Boolean Algebra
     1. Order of Operations
     2. Association, Commutivity, and Distribution
     3. Identities Involving Zero and One
     4. Single-Variable Identities
     5. Two-Variable Identities and De Morgan’s Law
     6. Circuit Simplification
  4. Reversible Logic Gates
     1. Reversible Gates
     2. Irreversible Gates
     3. Toffoli Gate: A Reversible AND Gate
     4. Making Irreversible Gates Reversible
  5. Error Correction
     1. Errors in Physical Devices
     2. Error Detection
     3. Error Correction
  6. Computational Complexity
     1. Asymptotic Notation
     2. Complexity Classes
  7. Turing Machines
     1. Components
     2. Incrementing Binary Numbers
     3. Church-Turing Thesis
  8. Summary

Chapter 3 An Introduction to Customer Discovery

*No startup business plan survives first contact with customers.* – Steve Blank

*A journey of a thousand miles begins with a single step. – Lao Tzu*

**Overview of the Customer Discovery Process**

Customer discovery is the first step in the Customer Development process. Skipping this step can expose a startup to significant risk as demonstrated by the story of Iridium – one of the largest startup failures in history.

*One of the largest startup failures on record as they executed their business plan.*

*Customers don’t behave like your business plan.*

**The Customer Discovery Philosophy**

A startup begins with the vision of its founders: a vision of a new product or service that solves a customer’s problems or needs and of how it will reach its many customers. Customer discovery lowers the odds of spending zillions and getting zero in return.

The No. 1 goal of customer discovery is to turn the founders’ initial hypotheses about their market and customers into facts.

**Get Out of the Building**

Only after the founders have gotten out of the building and in front of customers repeatedly for weeks, if not months, will they know whether they have a valid vision or just a hallucination.

For a startup in customer discovery mode, the following facts are not in scope:

1. Understanding the needs and wants of all customers,
2. Making a list of all features customers want before they buy your product,
3. Handing product development a features list of the sum of all customer requests,
4. Handing product development a detailed marketing-requirements document,
5. Running focus groups and testing customers’ reactions to your product to see if they will buy.

Your pressing goal is to develop your product for the few, not the many.

In customer discovery you start building your product even before you know whether you have any customers for it.

What is it about a first product from a new company that’s different from follow-on products in a large company?

*On a startup’s first day, there’s limited – if any – customer input.*

**Search for the Problem/Solution Fit**

Have we on a problem that lots of people want us to solve or a need they want us to fill?

Does our solution (product, website, app) solve the problem in a compelling way?

At its core, the essence of customer discovery is to determine whether your startup’s value proposition matches the customer segment it plans to target.

This is the essence of problem/solution fit and it is only achieved when the revenue model, pricing, and customer acquisition efforts all match up with the customers’ needs.

**Develop the Product for the Few, Not the Many**

In a mature company, a market-requirements document is delivered to engineering with a sum of all possible customer features based on customer requests and complaints. The MRD drives a coordinated effort between product management, marketing, sales, and engineering. This effort goes towards modifying the product for the next release cycle.

Unlike this mature company, for startups on day one, there’s limited – if any – customer input for creating a formal product specification.

No startup can afford to build a product with every feature a mainstream customer needs all at once. The product would take years to get to market and be obsolete by the time it arrives.

Instead, successful startups solve this dilemma by focusing development and early selling efforts on a very small group of early customers who have bought into the startup’s vision – these are the earlyvangelists and they are the visionary customers that give the startup the feedback necessary to add features to the product over time.

*Earlyvangelists are willing to make a leap of faith and buy an early product.*

**Earlyvangelists: The Most Important Customers of All**

Evangelists are enthusiasts who spread the good news about your product to friends, family, or co-workers. They are driven to be early adopters and are willing to buy unfinished and untested products for a lot of different reasons including competitive edge and bragging rights.

Every industry has a small subset of visionaries willing to take a leap of faith on an early product.

One mistake that founders often make is to give away or heavily discount early alpha/beta products to blue-chip customers.

Earlyvangelists willing to pay is a critical part of the customer discovery process and you will use the insights in interactions with earlyvangelists to test the entire buying process.

In web/mobile apps, where multi-sided markets (separate users and payers) are often found, earlyvangelists can be users and payers. But even as nonpaying users, these earlyvangelists are willing or eager accelerators of your viral growth.

Earlyvangelists can be relied on for feedback and initial sales; they’ll tell others about the product and spread the word that the vision is real. They can also be potential advisory board candidates.

**Earlyvangelist Characteristics (Figure 3.1)**

Has a problem -> Is aware of having a problem -> Been actively looking for a solution and has a timetable for finding it -> Assembled an interim solution out of parts because the problem is painful -> Committed or can quickly acquire budget dollars to purchase

*Earlyvangelists are willing or eager accelerators of your viral growth.*

**Build a Minimum Viable Product (MVP) First**

The idea that a startup builds its products for a small group of initial customers rather than devising a generic mainstream spec is radical.

On day one, the startup doesn’t know who its initial customers are or what features they’ll want. So, without front loading customer discovery, the startup risks overengineering their vision and this invariably leads to wasted engineering effort, time, and cash.

This being said, completely front-loading customer discovery without having anything in-hand is risking lost time and having no basis on which to garner customer feedback.

The most productive approach is to develop the core features of the product (incrementally and iteratively with agile engineering methods), with the feature list driven by the vision and experience of the company’s founders. This is a Minimum Viable Product.

The goal of customer discovery is to test your understanding of the customer’s problem and see if your proposed solution will prompt him to use or buy the product based on its most important features alone.

Most users want finished products, but earlyvangelists are the perfect target for the MVP. Tailor the initial product release to satisfy their needs.

If no one thinks your MVP solution is interesting or sufficient (most important and indispensable), iterate or pivot until an adequate number say “yes”. If, and only if, no customers can be found for the most important features of the MVP, bring customers’ additional feature requests to product development.

In the customer development model, feature requests to an MVP are by exception and iteration rather than by rule. This eliminates the endless list of feature requests that often delay first customer chip and drive product development teams crazy.

MVPs for Web/Mobile are different because the customer discovery process is conducted differently. This is primarily driven by the much larger scale of customer reach attained by combining online with face-to-face interactions. More emphasis is placed of customer acquisition, activation, and referrals. MVPs can be developed faster and earlier and iterated with shorter cycle times. Customer data can be harvested in greater volume with greater speed and granularity.

*The goal of the MVP is to build the smallest possible feature set.*

**Developing the Minimum Viable Product for a Web/Mobile Product (Figure 3.2)**

**Use the Business Model Canvas as The Customer Discovery Scorecard**

Often there’s a lack of shared and clear understanding of the business model throughout a startup. The business model canvas is a diagrammatic tool designed to show how the startup believes it will make money across nine components: 1) product (value proposition), 2) customers (customer segments), 3) channels, 4) demand creation (customer relationships), 5) key partners, 6) revenue models (revenue streams), 7) key resources, 8) key activities, and 9) cost structures.

In this customer discovery phase, you’ll develop a one- or two-page brief about each of the following boxes (components) of the business model canvas and dig deeper into things like:

* Market size: how big is the opportunity,
* Value proposition, part 1: the product/service, its benefits and MVP,
* Customer segments: who the customer is and what problems the product solves,
* Channels: how the product will be distributed and sold,
* Customer relationships: how demand will be created,
* Value proposition, part 2: market-type hypothesis and competitive set/differentiation,
* Key resources: suppliers, commodities, IP, or other essential elements of the business,
* Key partners: other enterprises essential to success of the business,
* Revenue streams: revenue and profit sources and size.

Customer Development uses the canvas as a “scorecard” to track how the canvas changes over time (weekly frequency) as the founders search and validate or invalidate components of canvas in their search for a business model.

Once a week update the canvas to reflect any pivots or iterations, highlighting in red the changes from last week.

When the canvas is utilized effectively, it allows founders to search for alternatives when necessary to survive another day.

As you complete the hypothesis development, your business model canvas quickly becomes multi-dimensional. You are three initial canvases:

* Core elements of the business model itself (value proposition, channels, etc…)
* Hypotheses you have for each element of the business model (“people will want these features”, “people will buy our product because…”, and so on)
* A layer outlining the key pass/fail tests you will use to get face-to-face with customers and use their feedback to convert your hypotheses into facts.

**Business Model Canvas (Figure 3.3)**

**Sample Business Model Canvas – Initial Hypothesis (Figure 3.4)**

**Using the Business Model Canvas as a Weekly Scorecard (Figure 3.5)**

**Customer Discovery: Overview of the Process (Figure 3.6)**

Customer Discover has Four Phases

In Phase 1

In Phase 2

In Phase 3

In Phase 4