Core Principles and Features of Software Platform

CHAPTER 3

When you have completed your study of this two chapters, you will be able to

1. Understanding ecosystem, platform lifecycles using dominant designs, S-curves, and diffusion curves.
2. Platform properties: multi-sidedness, network effects, multi-homing, architecture, and governance.
3. Platform dynamics: tipping, locking-in, competitive durability, and envelopment.
4. How platform businesses differ from product and service business.
5. Why platforms need a mindset different from products and services.

Why do Big 7 occupy in Top 10 ?

3.1 Core Concepts

Ecosystem

- An environment consists of all the mutually dependent stakeholders, business units, and companies working in a particular area.
- In a typical ecosystem, there are producers and consumers, where the consumers add value to the consumed resources.
- In a software ecosystem, a supplier of an application working with companies doing the installation and support in different regions. Neither one could exist without the other.
- Ecosystems can be permanent or temporary. Software engineering economics provides the mechanisms to evaluate alternatives in establishing or extending an ecosystem, for instance, assessing whether to work with a specific distributor or have the distribution done by a company doing service in an area.

3.1 Core Concepts

Platform Lifecycle

The lifecycle of most technology innovations has three parts:

- The emergence of the ideal technology solution itself.
- The progression along a technology maturity curve.
- The penetration in its pool of prospective end-users.

These lifecycle characterizations apply both to platforms and Apps, we generically refer to both as technology solutions in describing platform lifecycles.

3.1 Core Concepts

Platform Lifecycle

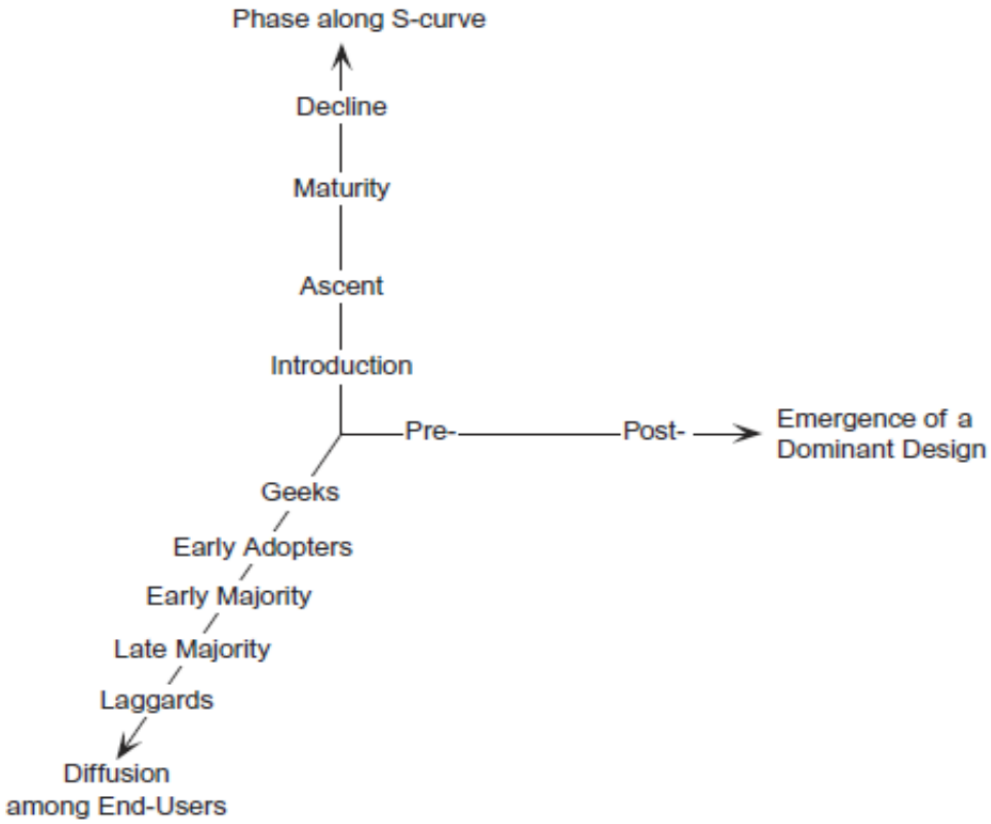


FIGURE 2.1

The three dimensions of the platform lifecycle.

3.1 Core Concepts

Emergence of a dominant design

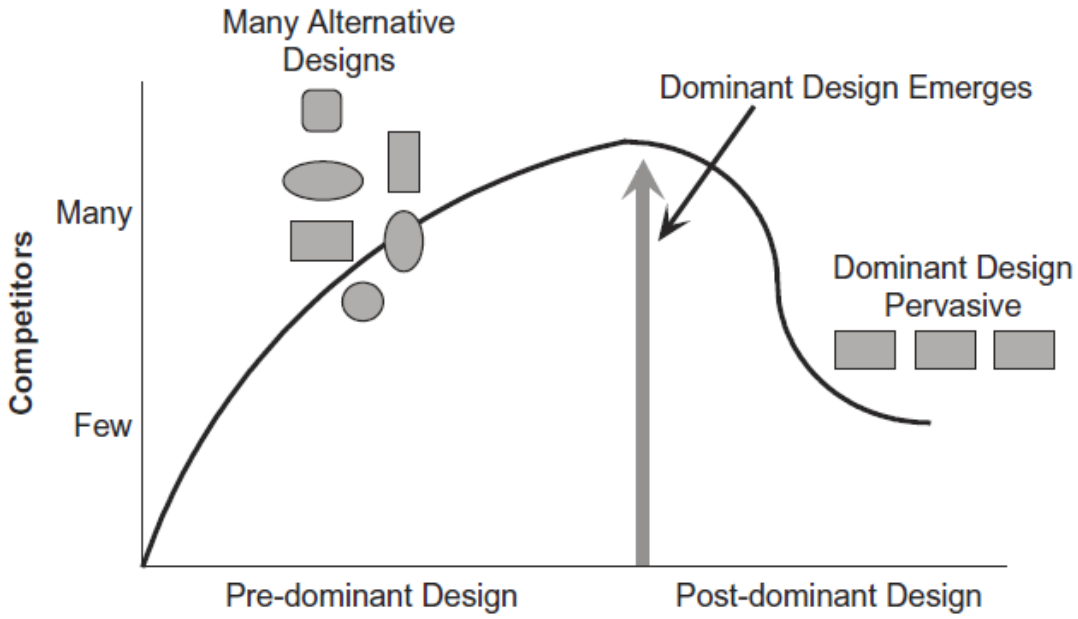


FIGURE 2.2

Pre- and post-dominant design phases in a software platform.

3.1 Core Concepts

Technology maturity curve

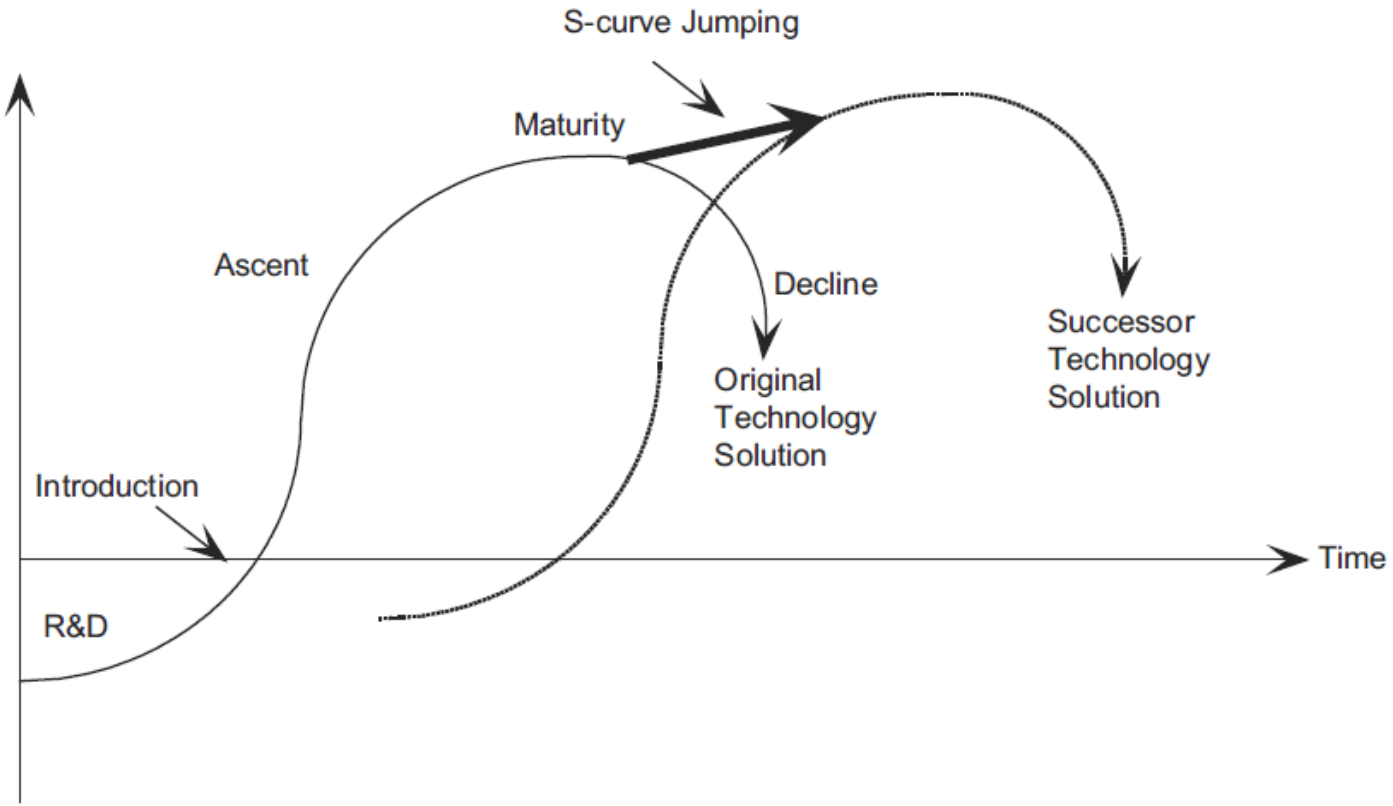


FIGURE 2.3

S-curves in the technology lifecycle.

3.1 Core Concepts

- **Process innovation replaces product innovation**

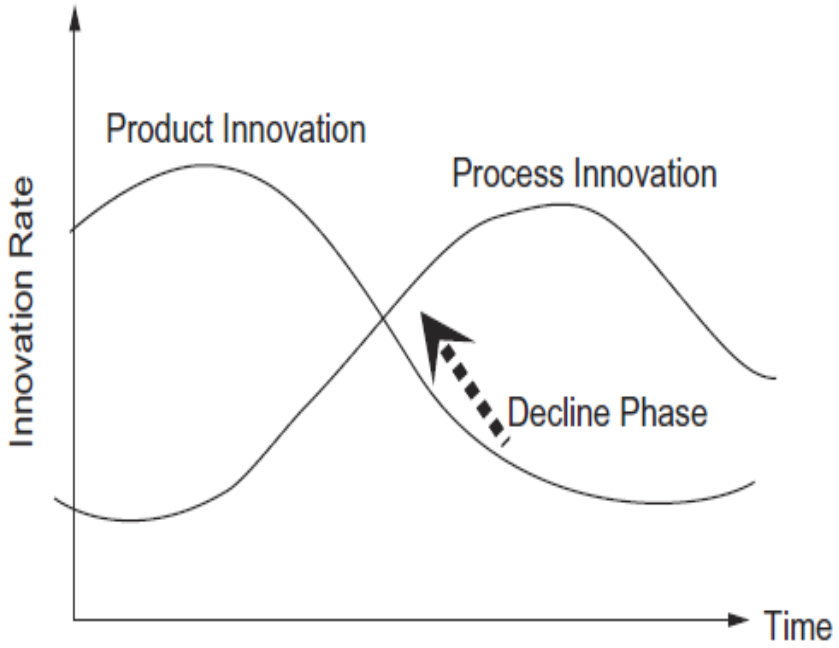
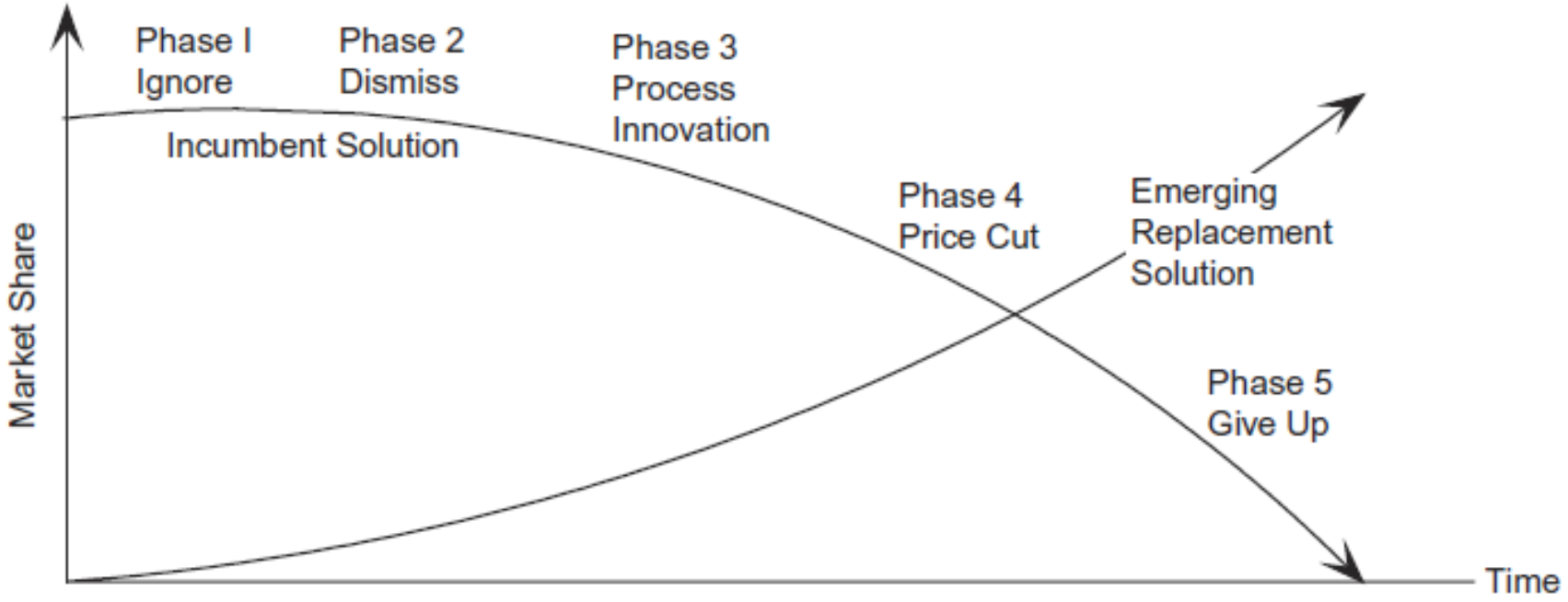


FIGURE 2.4

Process innovation replaces product innovation as a technology solution matures.

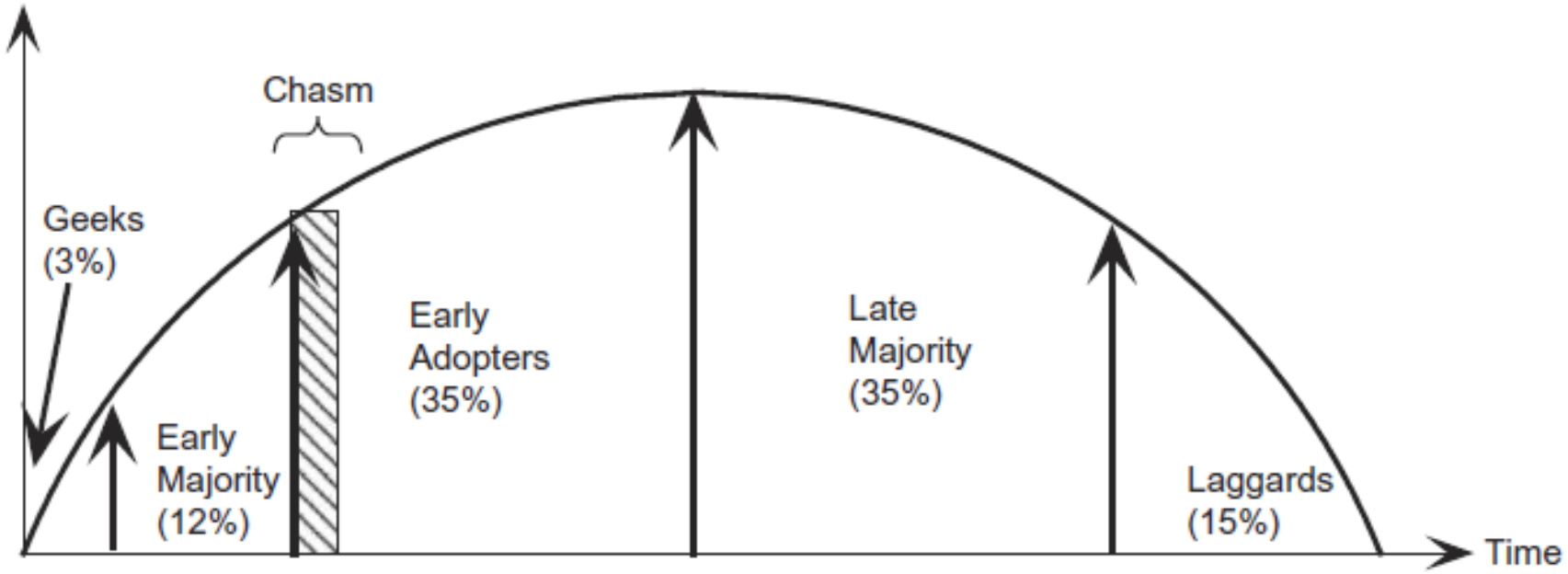
3.1 Core Concepts

- **Incumbent tech-solution reacts to potential replacement technology**



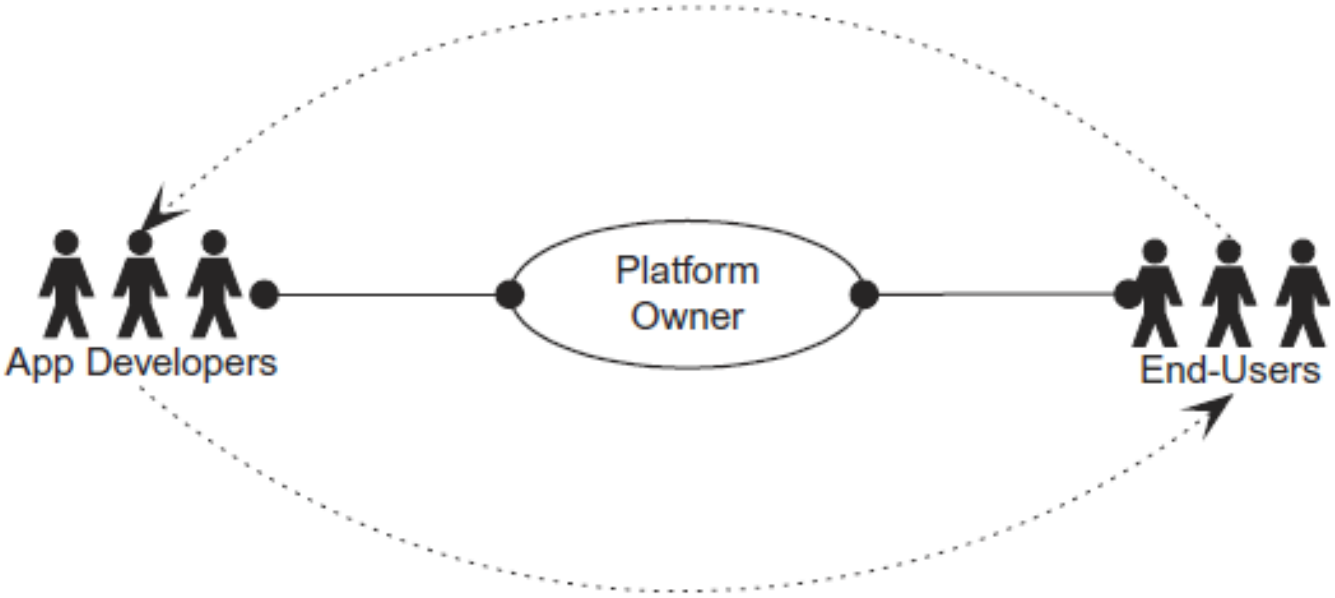
3.1 Core Concepts

- **The technology diffusion curve on the end-user side**



3.1 Core Concepts

- **Multi-sidedness**



3.1 Core Concepts

- **Multi-sidedness**

Table 2.1 Examples

Multisided Market Makers Outside the Software Industry

Table 2.2 Examples of Multisided Market Makers Outside the Software Industry

Two-Sided Market-Maker	First Side	Second Side
Credit cards	Cardholders	Merchants
HMOs	Patients	Physicians
Operating systems	End-users	Application developers
Travel sites	Travelers	Airlines
e-healthcare	Physicians	Medical services firms
Agoras in ancient Greece, shopping malls	Buyers	Merchants
Business schools	Future managers	Employers
Cable television networks	Subscribers	Content providers and studios
Professional services firms	Professional specialists	Clients
Flea markets	Bargain shoppers	Sellers

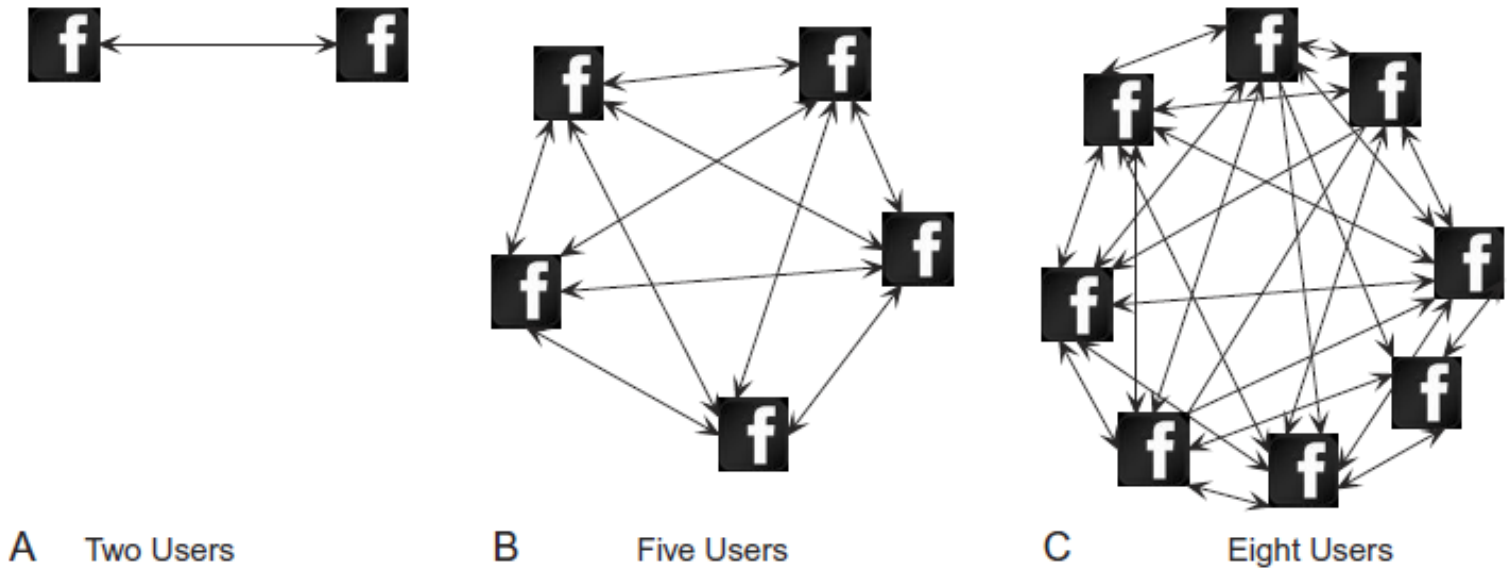
3.1 Core Concepts

Network Effects

- Network effects refer to the degree to which every additional user of a platform or app makes it more valuable to every other existing user.
- Economists call these network externalities (Katz and Shapiro, 1994; Saloner and Shepard, 1995) or Metcalfe's law.
- **Question**: What is Metcalfe's Law ?
- **Metcalfe's law** states the effect of a telecommunications network is proportional to the square of the number of connected users of the system (n^2).

3.1 Core Concepts

Network Effects



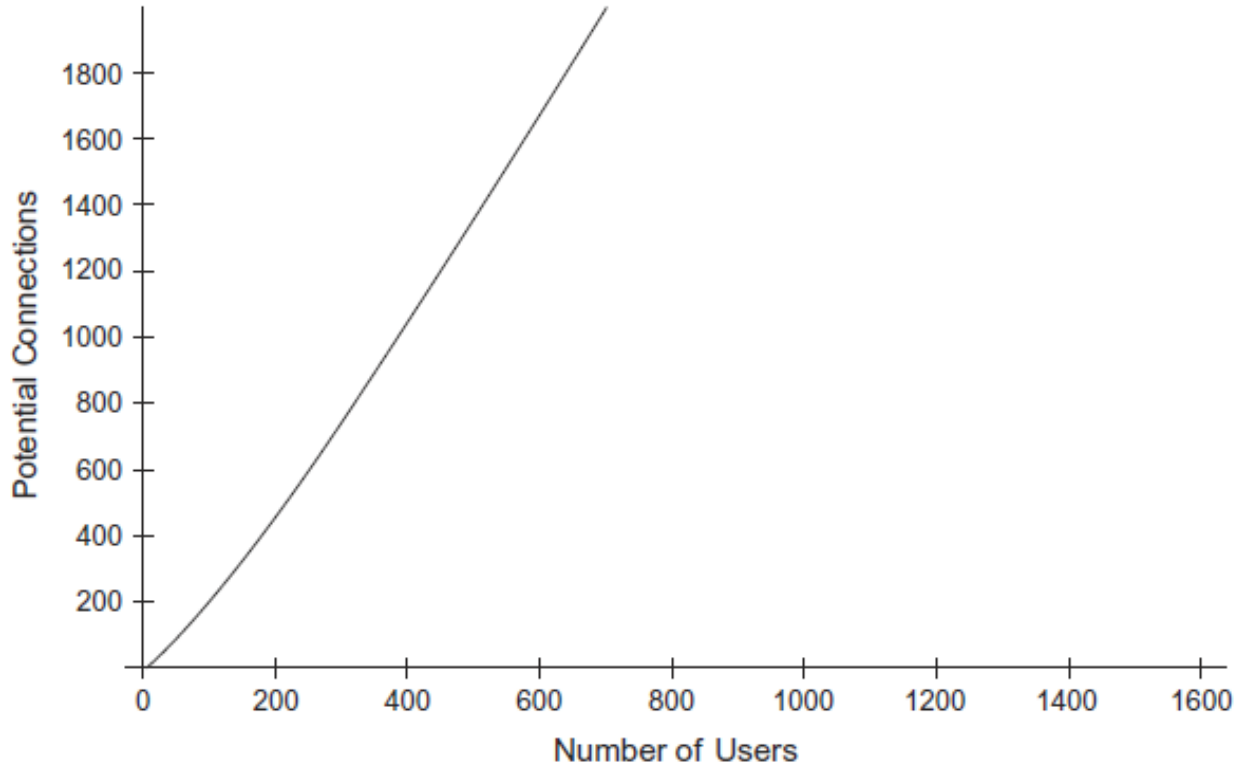
A Two Users

B Five Users

C Eight Users

3.1 Core Concepts

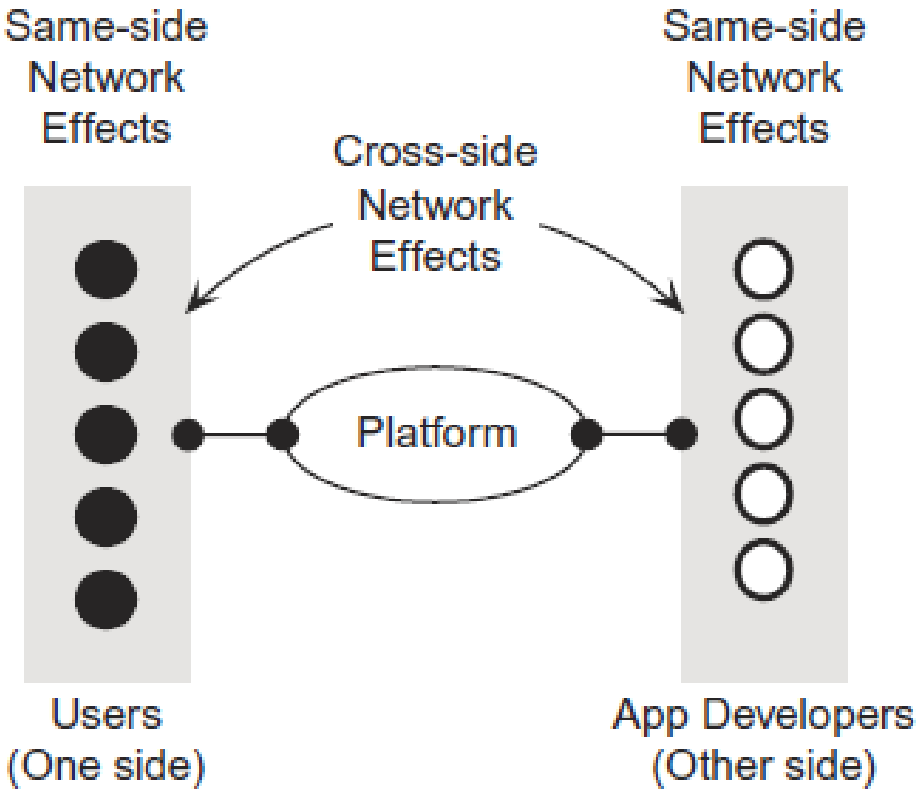
Network Effects



3.1 Core Concepts

Network Effects

- Direction
  - Positive direction
  - Negative direction
- Sidedness
  - Positive sidedness
  - Negative sidedness
- Cases



3.1 Core Concepts

■ Multi-homing

When a platform participant on either side participates in more than one platform ecosystem.

■ Tipping

This minimum number of adopters after which network effects are manifested is known as the platform's critical mass or tipping point.

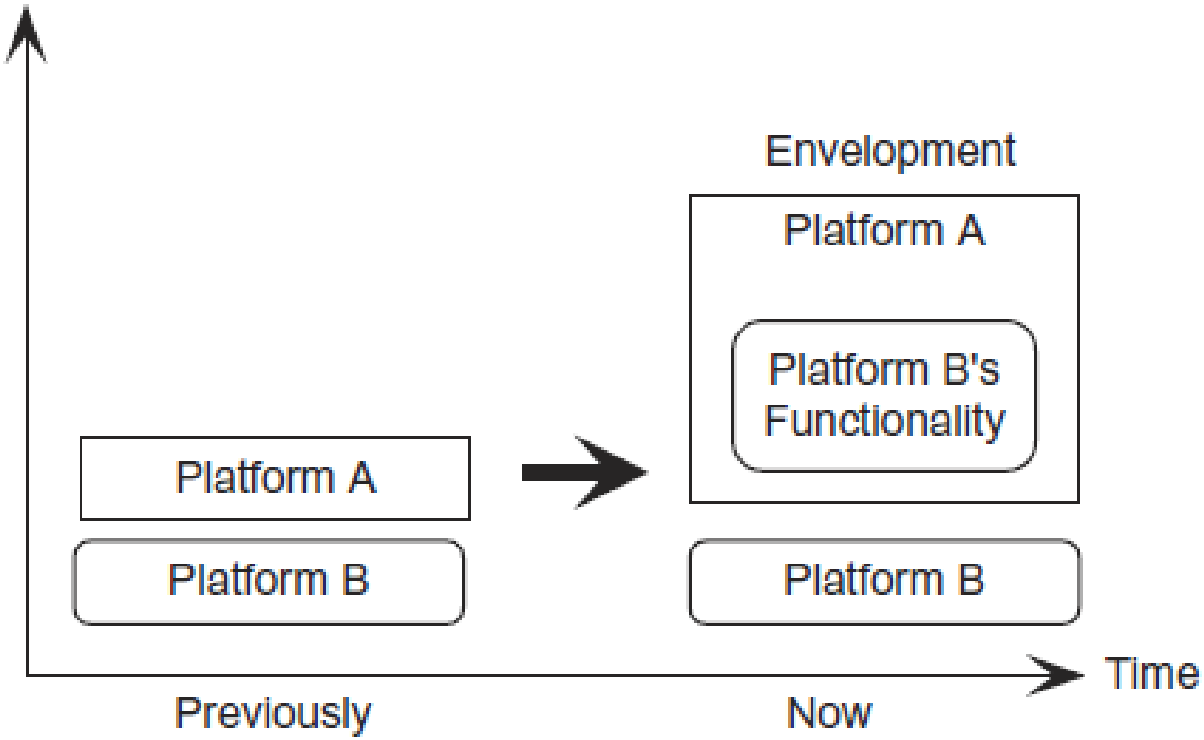
■ Lock-in

The ways in which a platform can make it more desirable for existing users to stay put and not jump ship to a rival platform broadly refers to lock-in. Lock-in can occur both in consumer-grade platforms (e.g., iOS, Android) and in enterprise-grade platforms (e.g., SAP).

3.1 Core Concepts

Envelopment

When a platform begins to offer the functionality of another platform in an adjacent market in addition to its existing bundle of functionality, it is said to have enveloped—or swallowed—the latter).



3.1 Core Concepts

Architecture

Architecture is the conceptual blueprint that describes the structure of a technology solution. Architecture describes the components of a complex system, what they do, and how they interact.

Governance

Governance impacts the evolutionary dynamics in platform ecosystems, and the competitive advantage generated by governance choices.

3.2 Guide Principles

- The **Red Queen** principle
- The chicken-or-egg problem
- The penguin problem
- The seesaw problem
- The Humpty Dumpty problem
- The mirroring principle
- Coevolution
- The Goldilocks rule

The Red Queen principle

Lewis Carroll's

“Through the Looking Glass”

The Red Queen said,

"*It takes all the running you can do, to keep in the same place*."

3.2.1 Key Concepts of Red Queen's principle

- **Co-evolution** is frequently seen in pairs of species that interact frequently or closely. Examples include herbivores and plants, pollinators and plants, predators and their prey, and parasites and their hosts
- The co-evolutionary '**arms race**' between a parasite and host is known as the Red Queen Hypothesis as both organisms must 'keep running in order to stay still'. Hosts better able to resist and tolerate parasitism have greater fitness. Parasites better able to feed, reproduce and find new hosts have greater fitness.

3.2.1 Examples of Red Queen's principle

Predators and their prey

Polar bears and seals
In this Arctic environment, the white coat of the polar bear makes it hard to spot against the snow but seal pups are also white so they can't be seen by the Polar bear!

3.2.1 Examples of Red Queen's principle

In co-evolution, a change in the traits of one species acts as a selection pressure on the other species.

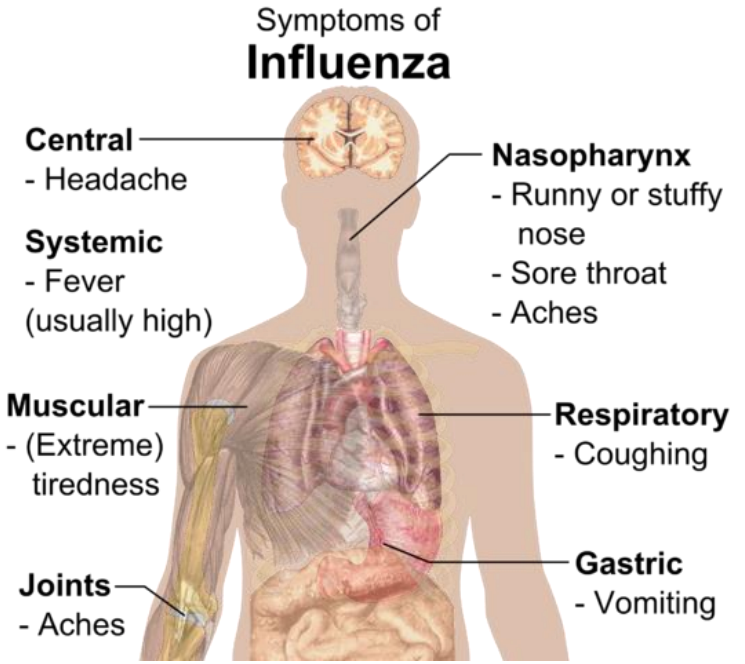
According to the Oxford Dictionary – the definition of a parasite is:

An organism which lives in or on another organism (its host) and benefits by deriving nutrients at the other's expense.

Parasites and their host

Influenza and its human hosts

Influenza is an example of an RNA virus and it has many hosts. As we race to develop flu vaccines on an annual basis, the virus is more than able to keep up the evolutionary arms race due to the fact it has a high rate of mutation.



In-Class Discussion

Please according to your preview, explain the following principles.

- *The chicken-or-egg problem*
- *The penguin problem*
- *The seesaw problem*
- *The Humpty Dumpty problem*
- *The mirroring principle*
- *Coevolution*
- *The Goldilocks rule*

Thinking & Considering

*Which one principle or more principles, is/are more **important** in the development of software platform?*

Please, give your explanation.

Conclusion

Key Terms

Ecosystem, Platform Lifecycle, S-Curve

Multi-sidedness, Network Effects, Multi-homing

Tipping

Lock-in

Envelopment

Architecture

Governance

The Red Queen

The chicken-or-egg problem

The penguin problem

The seesaw problem

The Humpty Dumpty problem

The mirroring principle

Coevolution

The Goldilocks rule

Assignments

Self Study

- How platform businesses differ from product and service businesses
- Why platforms need a mindset different from products and services
- How products and services can evolve into platforms
- Four lenses for spotting platform opportunities

This exercises will be discussed in the next lessons. You should not submit this exercise to the teacher.

Question

Since platform companies do not charge customer service fees,

How do they make money?

Preview

Chapter 1 and Chapter 2

from the textbook “Return on Software, how to maximizing the return on your software investment”, written by Steve Tockey.