CS207 #6, 2 Nov. 2012

Gio Wiederhold
Hewlett 103
Homepage at
https://cs.stanford.edu/wiki/cs207/Main/HomePage
Syllabus:

1. Why should software be valued?
2. Open source software. Scope. Theory and reality
3. Intellectual capital and property (IP).
5. Market value of software companies.
6. Sales expectations and discounting.
7. Alternate business models.
8. Allocation
9. Life and lag of software innovation.
10. The role of patents, copyrights, and trade secrets.
11. Licensing.
12. Separation of use rights from the property itself.
13. Risks when outsourcing and offshoring development.
14. Effects of using tax havens to house IP.
People & IP

Last week, 26 Oct 2012, Vishal Sikka CTO of SAP

• Focus on personnel
  ➢ Largest cost item in a hightech company
  ➢ Represents the creative part of intellectual capital

• Software is the primary Intellectual property
  ➢ Generated by people
  ➢ Consumed by people internally
  ➢ Embedded in
    ■ products sold
    ■ services rendered
Review of layers

Intellectual Resources

Public & Private

Intellectual Capital
Rights owned by the business

Intellectual Assets
Available for transfer

Intellectual Property
Legally protectable

- Patents
- Copyrights
- Trade secrets
- Trade marks
- Contracts covering intellectual capital
Software users & IP

Companies that

1. develop & sell software → *
   - Basis of IP: income from sales and services

2. purchase & license software for internal use
   - Do not generate IP with software

3. develop software internally for their own use
   - Basis of IP: relative SW expense × all income

4. combinations
Allocation

1. When there are multiple products

2. When there are other contributors to income
   a. Substantial hardware
   b. Financial consultants in financial firms
   c. Experts in call centers
   d. Brand name

   Not all of the income can be allocated to software

   Pareto Optimum
Pareto Optimality
(not Pareto Efficiency: 80/20 rule)

The point were any change lowers the total benefit/cost

1. Spending more on software will have less benefit than spending on other stuff
   a. People
   b. Hardware
   c. Advertising

2. Spending more on people ... lowers the total benefit/cost

Conclusion:
• If a company is managed optimally, we can allocate IP contribution by multi-year spending patterns
Management

• Responsible for optimal operation
• Invest to obtain greatest benefit
• Lags differ
  ➢ People 3 months to a year
  ➢ Computers 1 months
  ➢ Communication 3 months
  ➢ Cloud services
  ➢ Acquisitions 3 months to 3 years, or never
  ➢ Outsourcing contractors 3 to 6 months
• For the Pareto-optimality allocation of income one simply uses cost.

➤ But recall: Do NOT use cost as a surrogate for value, value of intangibles come from income.

1) Some spending supports routine income
   □ computers, facilities, cloud services, outsourcing …

2) Other spending supports non-routine income
   ❖ creative people: engineers, programmers, marketeers
   ❖ those should be valued, taking leverage and lag into account

Approach: only consider 2) the `Residual’ ← economists’ term
Staff Growth: Linear
Effort total = \( \frac{1}{2} E \times T \)
A simple metric: lag vs completion=
Centroid of prior expenditure
here @ 33% (without discounting)

“Gestation period”
Marketing

• Business model must allocate spending optimally
  ➢ Technology, as needed, long life and lag
  ➢ Marketing, necessary, less lag, slower growth
    □ For large 10 IT companies the average value allocated to their brand name is 22% (BW survey).

• Interdependence viral
  ➢ Consistent
  ➢ Relevant
  ➢ Linked by a common name and label
  ➢ Honest
Various Lags

- **Development Lag**
- **Centroid of Total Development Cost**
- **Research, Design, Implementation**
- **Development Done**
- **Centroid of Pre-sales Marketing Costs**
- **Marketing**
- **Marketing Lag**
- **Post-sales Marketing, Part of Sales Cost**
- **Manufacturing & Distribution Delay**
- **Distribution to Sales**
- **Release to Production (RTP)**
- **Sales Lag**
- **Sales**
- **Centroid of Total Revenue**
- **Costs**

- **Development**
- **Costs**
- **Time**
- **Revenues**
Expense leverage
A valuation based on cost

1. Collect the expenses $e_i$ over the total lag period $p$
2. Adjust past costs by a capitalization rate $d$, $a_i = (1+d)^{p-i}$
3. For year $i = 1 \rightarrow p$ estimate the R&D retained $r_i = 1 - 1/p$
4. Aggregate retained to the end date, $R = \sum r_i \times e_i \times a_i$
5. From experience, publications obtain an expected leverage $m$; $m$ can range from 1 to 20 ...
6. Expected value of IP $V = m \times R$ \(m\approx2\) in the first model we used

But the estimation of $m$ is verrrrrrrrrrrrry iffy

Technological advances are rarely stable

But could be used for
a) venture investing $m = 6 \rightarrow 20+$
b) advertising -- much untrustworthy data
c) stable maintenance component only
Development done → General availability

Sales

Costs

Research, Implementation & Testing

Marketing

Centroid of revenue

Costs

development lag includes testing

sales lag

marketing lag

Centroid of pre-sales marketing costs

RTP

GA

2-Nov-12

Gio: CS207 Fall 2012

14
Quantified model efforts during SW development

100% level is based on effort at time of completion

Total Development

@ 0.33

25% Testing
@ 0.08

Release to sales

Simple Model

62.5 Implementation
@ 0.35

12.5% Research
@ 0.33

100% level is based on effort at time of completion

Effort

Total development period

100% - 25%

25%

75%

100% - 25%
A startup is unlikely to ramp up linearly

Use exponential growth, $\exp 0.025$

Assume

1. 12.5% research
   Given that idea is clear, only towards for implementation
2. 25.0% testing
   Minimal and risky
3. 67.5% left for implementation

- Overlap research and implementation until testing starts
- Overlap implementation and testing until RPS

Results

Overall centroid @ 0.27 before RPS -- later

Research from 1.00 to 0.33, centroid @ 0.65 before RPS
Implementation from 0.67 to 0.00, centroid @ 0.29 before RPS
Testing from 0.17 to 0.00, centroid @ 0.08 before

Hiring rate at RPS 21%, at the limit for effectiveness

Ignore different staff salaries
Graph of start-up development

- Implementation starts when 67% time remains
- Research ends when 33% time remains
- Testing starts when 17% time remains
- 12.5% Research @0.65
- 62.5% Implementation @0.29
- 25% Testing @0.05
- 21% effort growth

Release to sales

Start: 0
Done: 1

Gio: CS207 Fall 2012
Start up with 15% research and 50% testing effort.

Research ends and Testing starts when ~37% time remains.

Implementation starts when ~69% time remains.

~ 15% Research

~ 35% Implementation

50% Testing

Release to sales
Development in mature company with 12.5% research and 25% testing effort, 62.5% implementation.

Available resources

38% effort growth at start

Res., Imp, & Test @0.42

Research ends when 65% time remains

Implementation starts when 85% time remains

Testing starts when 40% time remains

5% company staff growth

Release to sales

Relative Effort

Values based on finite integration, exp = 0.05

Gio: CS207 Fall 2012
Product revision, Mature development with 50% testing effort

![Diagram showing effort growth and resource availability over time]

- Effort growth at start: 35%
- Available resources:
  - Effort: 50%
  - Time: 8% at 0.22
  - Overall: 50% R & I at 0.58
  - 50% testing (T) at 0.38

Research ends and Testing starts when 67% time remains.

Date: 2-Nov-12

Gio: CS207 Fall 2012
Summary: Maturity effect

- Start: Nov-12
- Total development period: 0.75
- Effort:
  - Research: 50%
  - Implementation: 25%
  - Mature growth: 25%
  - Testing: 0%

Maturity effect:
- Research: @0.42
- Simple Model: @0.33
- Startup: @0.26
- Release to sales
Effective lag = Development period × Centroid fraction

Lag differs less than development period
New considerations

1. Have staff already
   a. Early versions rapid growth, but observe ~20% limit
   b. Later, best grow slower

2. Can overlap version development
   a. Don’t let valuable staff be idle
   b. Missing features should already be understood
   c. Rapid analysis of problems to allow next version fixes
   d. Any research should be done before major staff effort

3. Adequate testing to keep reputation
2nd version technical lag

Staff becomes available when prior version enter testing

Rapid, 33% increase in personnel per version interval

25% Testing for version \( n \)
 Starts at .057

All 100% Starts at .057

Research & Implementation

Release version \( n-1 \)

Research for version \( n \) release

Implementation for version \( n \) release

Release version \( n \)

Staff becomes available when prior version enter testing

Rapid, 33% increase in personnel per version interval

25% Testing for version \( n \)
 Starts at .057

All 100% Starts at .057

Research & Implementation

Release version \( n-1 \)

Research for version \( n \) release

Implementation for version \( n \) release

Release version \( n \)

Staff becomes available when prior version enter testing

Rapid, 33% increase in personnel per version interval

25% Testing for version \( n \)
 Starts at .057

All 100% Starts at .057

Research & Implementation

Release version \( n-1 \)

Research for version \( n \) release

Implementation for version \( n \) release

Release version \( n \)

Staff becomes available when prior version enter testing

Rapid, 33% increase in personnel per version interval

25% Testing for version \( n \)
 Starts at .057

All 100% Starts at .057

Research & Implementation

Release version \( n-1 \)

Research for version \( n \) release

Implementation for version \( n \) release

Release version \( n \)

Staff becomes available when prior version enter testing

Rapid, 33% increase in personnel per version interval

25% Testing for version \( n \)
 Starts at .057

All 100% Starts at .057

Research & Implementation

Release version \( n-1 \)

Research for version \( n \) release

Implementation for version \( n \) release

Release version \( n \)
Mature ongoing technical lag

Staff becomes available when prior version enter testing

Research & Implementation for version $n$ release

25% Integration and Testing

Overall @ 0.63

@0.77

@0.21

Delivery of version $n-1$

Delivery of version $n$

Version $n$ development interval

Done
2\textsuperscript{nd} Version
substantial testing

\begin{itemize}
\item 33\% increase in personnel per version interval
\item 43\% Testing during version \textit{n} interval
\item R&I @1.00
\item All@0.61
\item T@0.33
\item Release version \textit{n-1}
\item Release version \textit{n}
\item Effort
\item 100\%
\item 25\%
\item 50\%
\item 75\%
\item 56\%
\end{itemize}
Version development, mature growth, much testing.
1. Determine type of development
   a. Startup
   b. Simple
   c. Mature
   d. Ongoing?

2. Determine interval of development

3. Does testing contribute to IP?

4. Determine growth of personnel effort
   Lag, because it precedes IP diminution, has a large effect on economic valuation
Effort total = 8.6 x original effort
Test ratio: 37%

Multi Version product effort and lag
First to market advantage

Original Multi Version efforts and lag

Competition (drawn to scale)
Growth Rate 20%/year average
Effort total = 5.4 units

Competition/Original multi version source
Effort ratio R/O = 0.63
Time ratio \(\frac{t(R)-s(R)}{t(O)-s(O)}\) = 0.41
Effective Lag ratio = 0.23

But at that point the original is 3.5 versions ahead of the competition!
Discussion

• A long-lived product is hard to displace if
  ➢ It is well maintained,
    ■ but that becomes costly
  ➢ Keeps up with all standards

• Internal replacement
  ➢ Should be easier
    ▪ But has not been in practice
Lag conclusion
(snake in the grass)

Lag is the effective development period
Has a large effect on early valuations
1. No income during that time
2. A chance for others to overtake you

We assumed a lag of 2 years in the examples.

Depends on product development strategy.

Separate paper on website
Say you want to delegate sales in Europe to some company EUsales that can do it easier over there

• How do you set the fees or royalties?

  1. You have computed a value of your SW of $1M
     ▪ But without discounting, it is actually $1.6M = \sum (due old, slide 5)
     ▪ You will also maintain the SW 1.36M = \sum (maintenance cost, slide 12)

     The total due is $3M

  2. You expect the European sales will be 40% of total, 20 000
     ▪ The reason for not discounting is that funds arrive at the same times.

• To earn the same you should charge \frac{1}{2} \times \frac{1}{1.6} \times 1.36M = $150/unit
  ▪ It does not matter how EUsales sells it and what it charges
  ▪ Complexities are required language, interface improvements