CS207 #3, 11 Oct 2013

Gio Wiederhold

http://infolab.stanford.edu/people/gio.html

Hewlett 102

Don’t forget to initial the Signup sheet

Syllabus:

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

Review definitions: Intangibles

- Software is an intangible good

  If it is owned it is considered Intangible Property

In a business there are 3 parts that have value.

(Contributes to potential income)

1. **Tangible goods**: buildings, computers, money
2. The **know-how** of management & employees
3. **Intellectual property**: Software, patents, etc.

   2. + 3. make up the **Intellectual Capital** of a company.

Software is slithery !

Continuously updated

1. Corrective maintenance

   bugfixing reduces for good SW

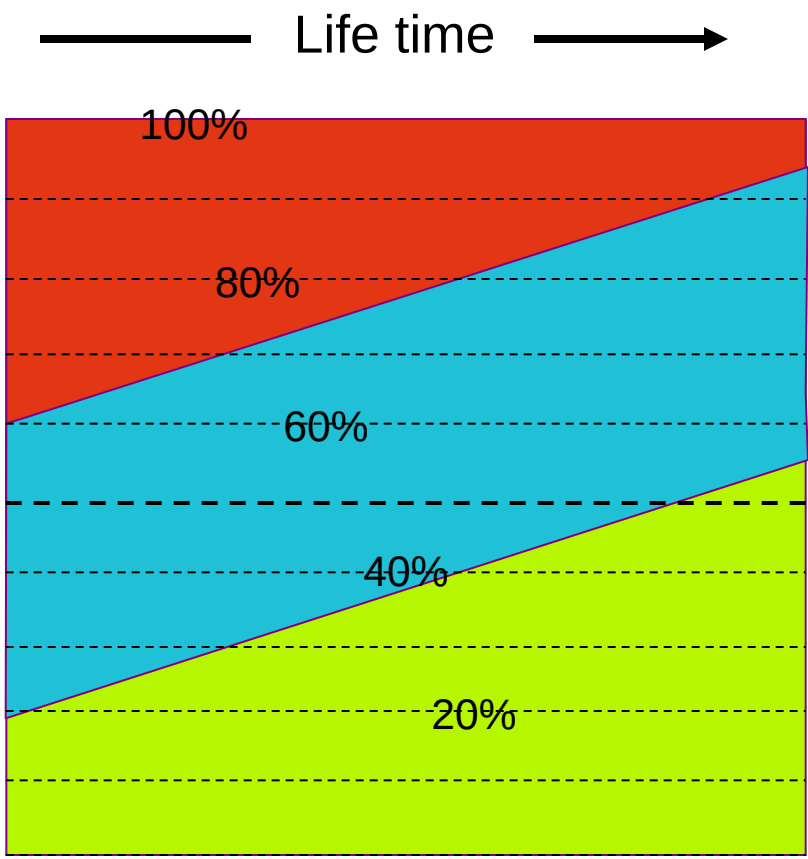
2. Adaptive maintenance

   externally mandated

3. Perfective maintenance

   satisfy customers' growing expectations

[IEEE definitions]



Ratios differ in various settings

Ongoing IP sources

- Corrective maintenance
  - Feedback through error reporting mechanisms
    - Taking care of bugs and missed cases, conditions
    - Complete inadequate tables and dimensions
- Adaptive maintenance
  - Staff to monitor externally imposed changes
    - Compliance with new standards
    - Technological advances
    - Keeping with viruses, spam etc.

    Effort depends on number & volatility of external interfaces
- Perfective maintenance
  - Feedback through sales & marketing staff
    - Minor features that cannot be charged for

Current value

Prior investment has created what you have now

"a bunch of software"

- That's what's to be valued
  - Based on reasonable expectations
    - future maintenance will be needed to earn income
    - future maintenance represents future investments

      More "software code"
  - not promises of new innovations ← new IP

Later we look at other valuation/business models

Technical Parameters needed

design, code, ...

IP is to be valued as of some specific date

1. **Life** of the IP in the product from that time on

   The interval from completion until little of the original *stuff* is left

2. **Diminution of the IP** over the Life

   A bit like a depreciation schedule, but based on content replacement, until little IP is left. 10% is a reasonable limit.

3. **Lag period***, interval from transfer to start of IP diminution
   - also called "Gestation Period

   Effective Lag = the average time before an investment earns revenue

4. **Relative allocation,** if there are multiple contributors to income.

Price

remember IP = f(income)

- <u>But --- Price stays ≈ fixed over time</u>

like hardware Moore's Law

- <u>Effect</u>

The income per unit of code reduces by 1 / size →

Total income

Total income = price × volume (year of life)

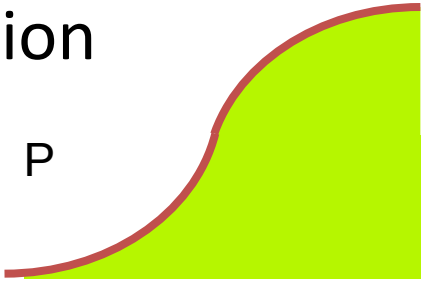
- Hence must estimate volume, lifetime

Best predictors are Previous comparables

- Erlang curve fitting (m=6 to 20, 12 is typical)

and apply common sense limit = Penetration

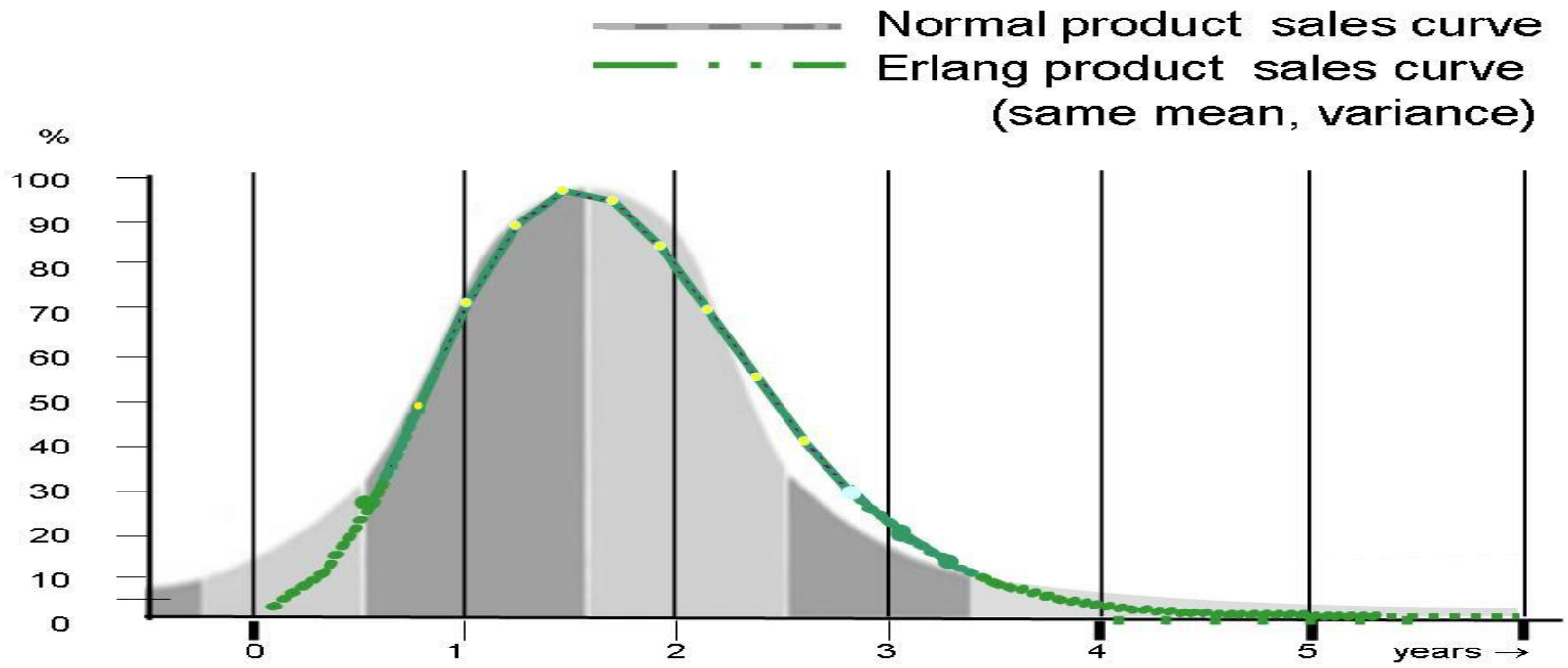
- estimate total possible sales F × #customers
- above F= 50% monopolistic aberration



Sales models

1. Normal curve: simple, no defined start point
2. Erlang: realistic, more complex

both have same parameters: mean and variance



Fraction of income for SW

Income in a software company is used for

- Cost of capital *typical*
  - Dividends and interest ≈ 5%
- Routine operations -- not requiring IP
  - Distribution, administration, management ≈ 45%
- IP Generating Expenses (IGE)
  - Research and development, i.e., **SW** **≈ 25%**
  - Advertising and marketing Joint distr.&creator ≈ 25%

These numbers are available in annual reports or 10Ks

Example

Software product

- Sells for $500/copy
- Market size 200 000
- Market penetration 25%

➢ Expected sales 50 000 units

➢ Expected income $500 x 50 000 = $25M

What is the result?

Spreadsheet

Allows easy alteration of assumptions

but best to keep them simple: `Occam's razor'

- Updated version every 1.5 years
  - implies modestly sized product
  - responsive to customers
  - finite life – stop when less than 10% of original IP left
- Linear software growth per Roux Rule
  - linear staff growth if perfectly modularized
  - supra-linear staff growth if not, but constrained
  - few deletions (memory is free, not embedded SW)
- Split IGE for R&D and marketing equally
  - Bayes' base assumption

Combining it all

factor	**today**	y1	y2	y3	y4	y5	y6	y7	y8	y9
Version	1.0	2.0		3.0	4.0		5.0	6.0		7.0
unit price	$500	500	500	500	500	500	500	500	500	500
Rel.size	1.00	1.67	2.33	3.00	3.67	4.33	5.00	5.67	6.33	7.00
New grth	0.00	0.67	1.33	2.00	2.67	3.33	4.00	4.67	5.33	6.00
replaced	0.00	0.05	0.08	0.12	0.15	0.18	0.22	0.25	0.28	0.32
old left	1.00	0.95	0.92	0.88	0.85	0.82	0.78	0.75	0.72	0.68
Fraction	100%	57%	39%	29%	23%	19%	16%	13%	11%	10%
Annual $K	0	1911	7569	11306	11395	8644	2646	1370	1241	503
Rev, $K	0	956	3785	5652	5698	4322	2646	1370	621	252
SWIP25%	0	239	946	1413	1424	1081	661	343	155	63
Due old	0	136	371	416	320	204	104	45	18	6
Disct 15%	1.00	0.87	0.76	0.66	0.57	0.50	0.43	0.38	0.33	0.28
Contribute	0	118	281	274	189	101	45	17	6	2
Total	1032	≈ $ 1 million								

Result of Example

Software product

- Sells for $500/copy
- Market size 200 000
- Market penetration 25%

➢ Expected sales 50 000 units

➢ Expected income $500 x 50 000 = $25M

Earnings (Profit before taxes) is just $ 1M

after your salary etc ...

Only technical contribution

Value of product – not of company

- basis for setting a product price
- or license fee
  - The value of the company includes future potential
    - Human capital
  - The value of its trademark
    - derived from technical qualities
    - strengthened by marketing , including advertising, accounted for in SD&A after product availability

Guidance obtained

- We applied an overall Erlang sales curve
  - new versions keep market going but customers do not replace earlier versions
- The assumption are sufficiently simple that alternatives can be intelligently discussed
  1. keep development costs low
  2. design so that SW maintenance is low
  3. charge a higher price
  4. minimize sales cost, without reducing market size
  5. broaden the market
  6. or →

Business models

0. New versions do not replace earlier versions

Alternative business models

1. New versions encourage replacement
2. Provide related services
3. Charge for maintenance

   Lower initial cost, slower income stream

4. Make product Open source to broaden market

   Charge only for services

Alternate business model

Consider maintenance and its income

"Service model"

- More assumptions – now include cost @50% of value

1. Original cost $500 000 (used to estimate 2.)
   a. Maintenance cost 15%/year of aggregate original cost
   b. Maintenance fee 15%/year of original price, 1 year delay.
   c. 85% annual retention of customers.
2. Maintenance Lag = Δ (*t* cost , *t* income) = 1 year
3. Stop maintenance when cost > income

Additional Effect of service model

factor	**today**	y1	y2	y3	y4	y5	y6	y7	y8	y9
Version	1.0	2.0		3.0	4.0		5.0	6.0		7.0
Org.cost $K	500	Assume designed for maintenance								
Maint.cost	**0**	**75**	**86**	**99**	**114**	**131**	**151**	**174**	**200**	**230**
Spending	75	86	99	114	131	151	174	200	230	*Σ1523*
SW@Cust.	0	812	4475	9456	13735	15997	16243	15176	13520	11744
Maint.Fees	0	0	122	671	1419	2060	2060	2400	2437	2277
Total income	0	956	3906	6324	7116	6382	5045	3806	2897	2380
Contribute @25	0	239	977	1581	1779	1569	1261	952	724	570
Unspent SW	-75	153	877	1467	1648	1445	1088	752	495	306
Unspent Disc.	-75	133	663	965	942	718	470	162	87	40
Total	4 348	≈ $ 4.3 million >> sales only but $1 523M for maintenance								

Cost of maintenance = 1523/(500+1523) = 75% of total

typical

Labb

More years of service model ?

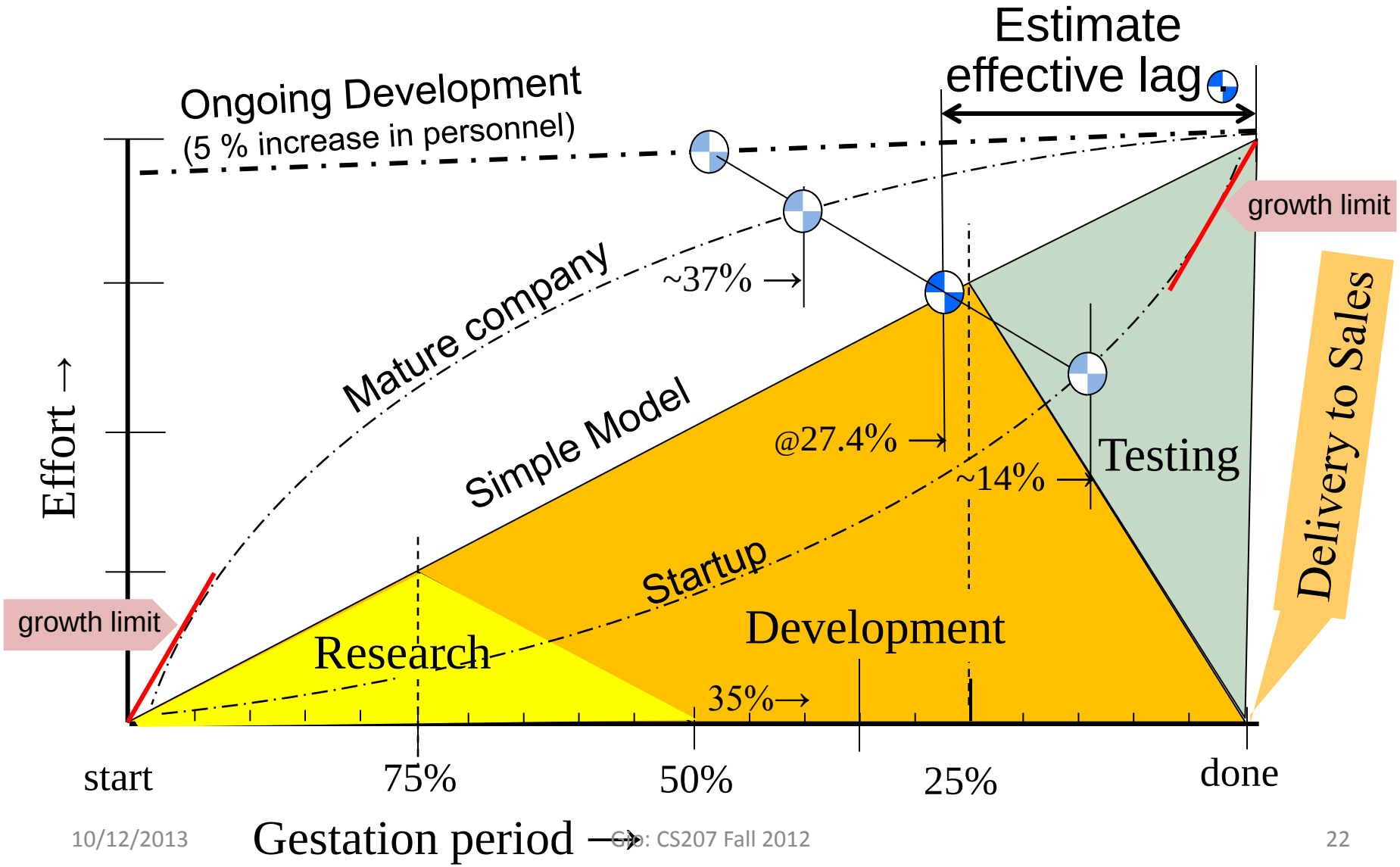
factor	**Cont.**	y10	y11	y12	y13	y14	y15	y17	y17	y18
Version	7.0	8.0		9.0	10.0		11.0	12.0		13.0
Curr.cost €K	1530									
Maint.cost	229	264	303	349	401	461	530	610	702	540
Spending	264	303	349	401	461	530	610	702	540	
SW@Cust.€M	11.7	10.1	8.6	7.3	6.2	5.3	4.5	3.8	3.2	1.9
Maint.Fees	2038	1761	1511	1289	1098	933	794	674	573	487
Total income	2280	1855	1543	1300	1101	934	794	674	573	487
Contribute @25	570	464	386	325	275	234	198	169	143	122
Unspent SW	306	160	40	-76	-186	-297	-411	-533	-397	-145
Unspent Total	2015	1551	1194	898	639	404	184	-27	32.6	221
Unspent Disc.	-75	133	663	965	942	718	470	162	87	40
Total SW	4 158	Less, out year losses because €5 687M spent on maintenance								

Good time to quit

But still have income to v12

Quit: reduce expense & income 1/3 each year

Lag delays benefits of R&D investments



Service model factors

- Same proportion was used for SW contribution: 25%
  - Maintenance income has lower sales cost, perhaps more should be made available for software improvements
- Discount total only after maintenance cost
  - Income comes at time of spending
- Maintenance fees still generate substantial income
  - Organize business sector to collect those in out years
  - Use excess SW income for replacement or new products
- Continue longer, but stop in time!
  - When maintenance costs more than income

More years of service model ?

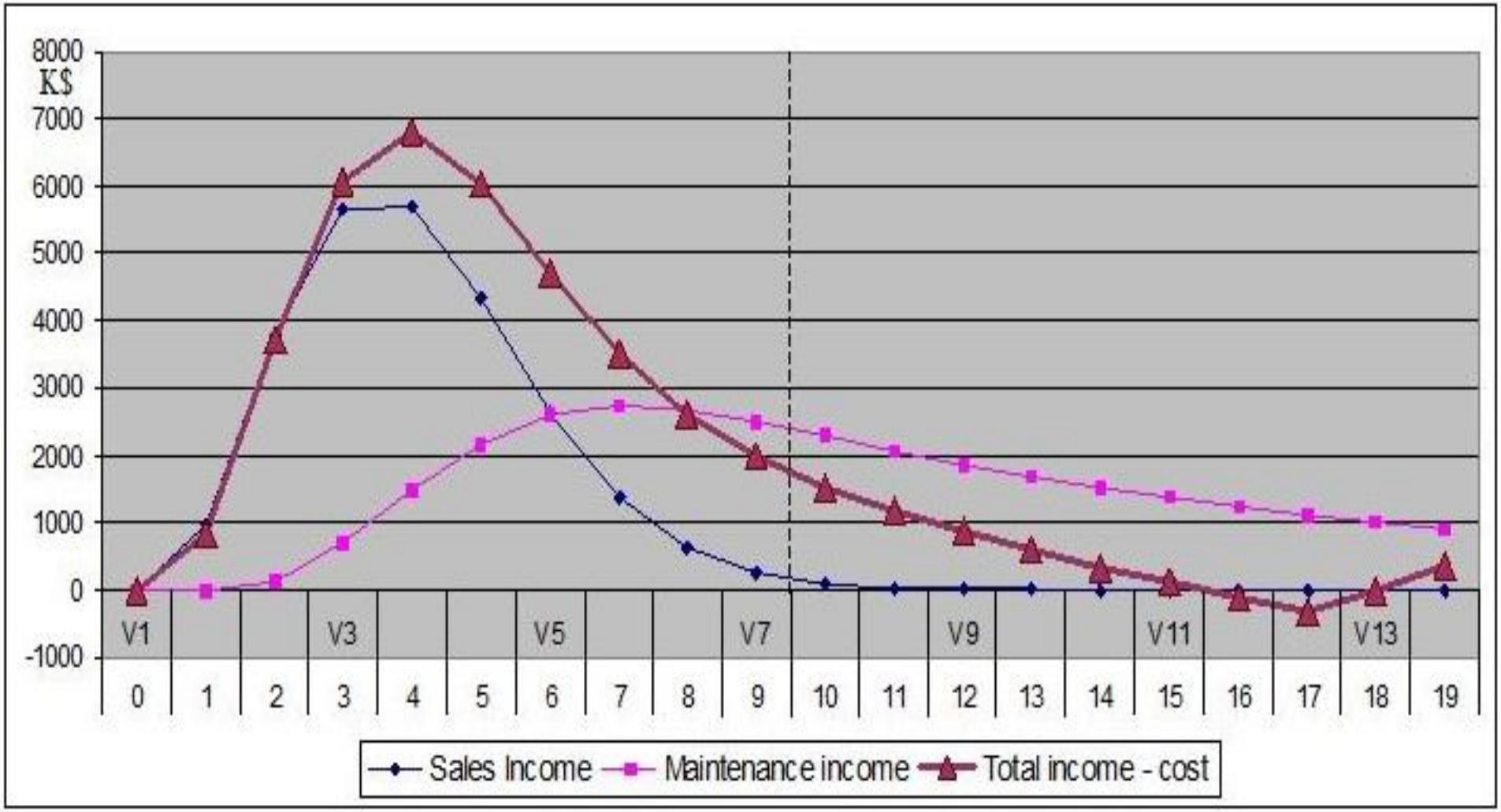
factor	**Cont.**	y10	y11	y12	y13	y14	y15	y17	y17	y18
Version	7.0	8.0		9.0	10.0		11.0	12.0		13.0
Curr.cost $K	1530									
Maint.cost	229	264	303	349	401	461	530	610	702	540
Spending	264	303	349	401	461	530	610	702	540	
SW@Cust.$M	11.7	10.1	8.6	7.3	6.2	5.3	4.5	3.8	3.2	1.9
Maint.Fees	2038	1761	1511	1289	1098	933	794	674	573	487
Total income	2280	1855	1543	1300	1101	934	794	674	573	487
Contribute @25	570	464	386	325	275	234	198	169	143	122
Unspent SW	306	160	40	-76	-186	-297	-411	-533	-397	-145
Unspent Total	2015	1551	1194	898	639	404	184	-27	32.6	221
Unspent Disc.	-75	133	663	965	942	718	470	162	87	40
Total SW	4 158	Less, out year losses because $5 687M spent on maintenance								

Good time to quit

But still have income to v12

Quit: reduce expense & income 1/3 each year

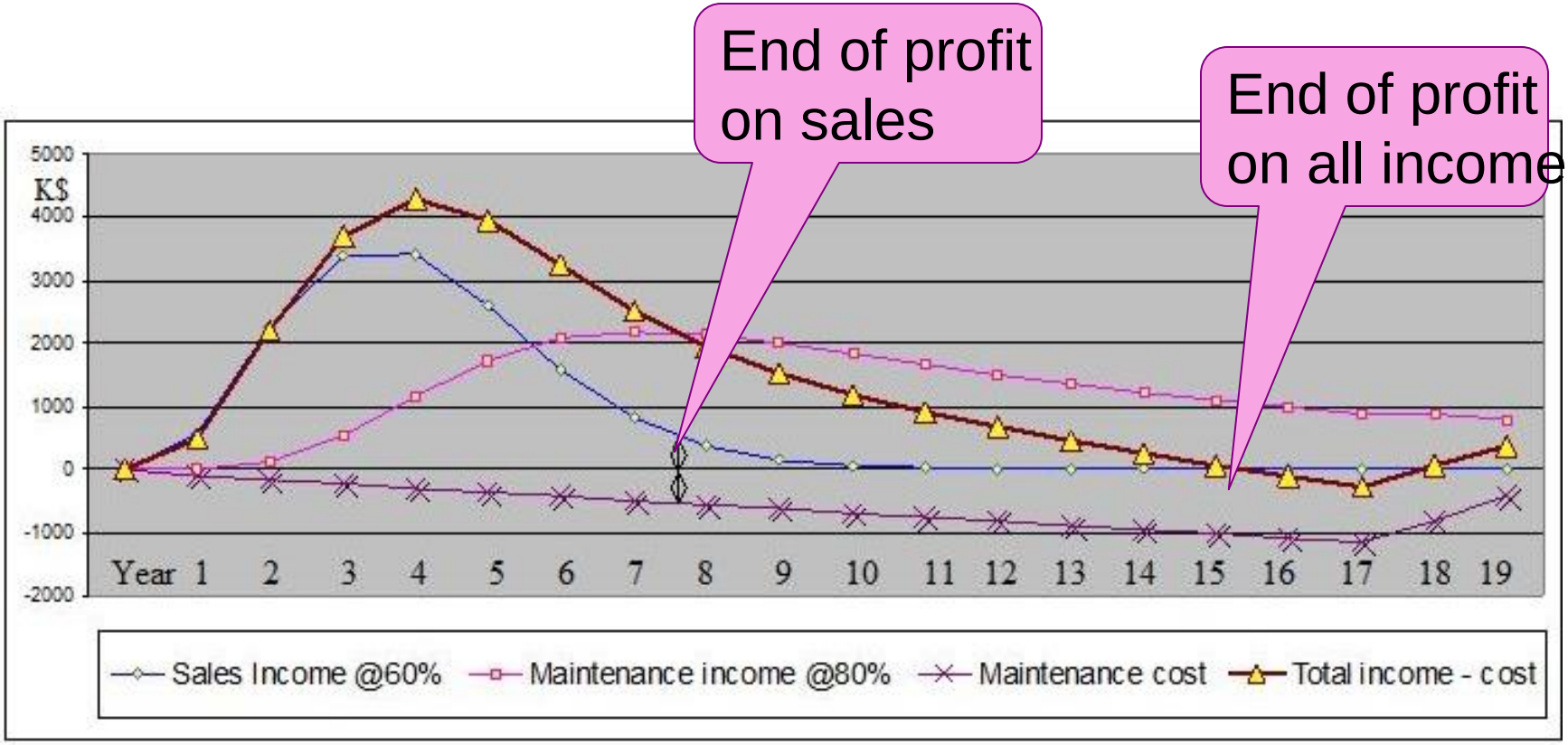
Total income vs technical cost



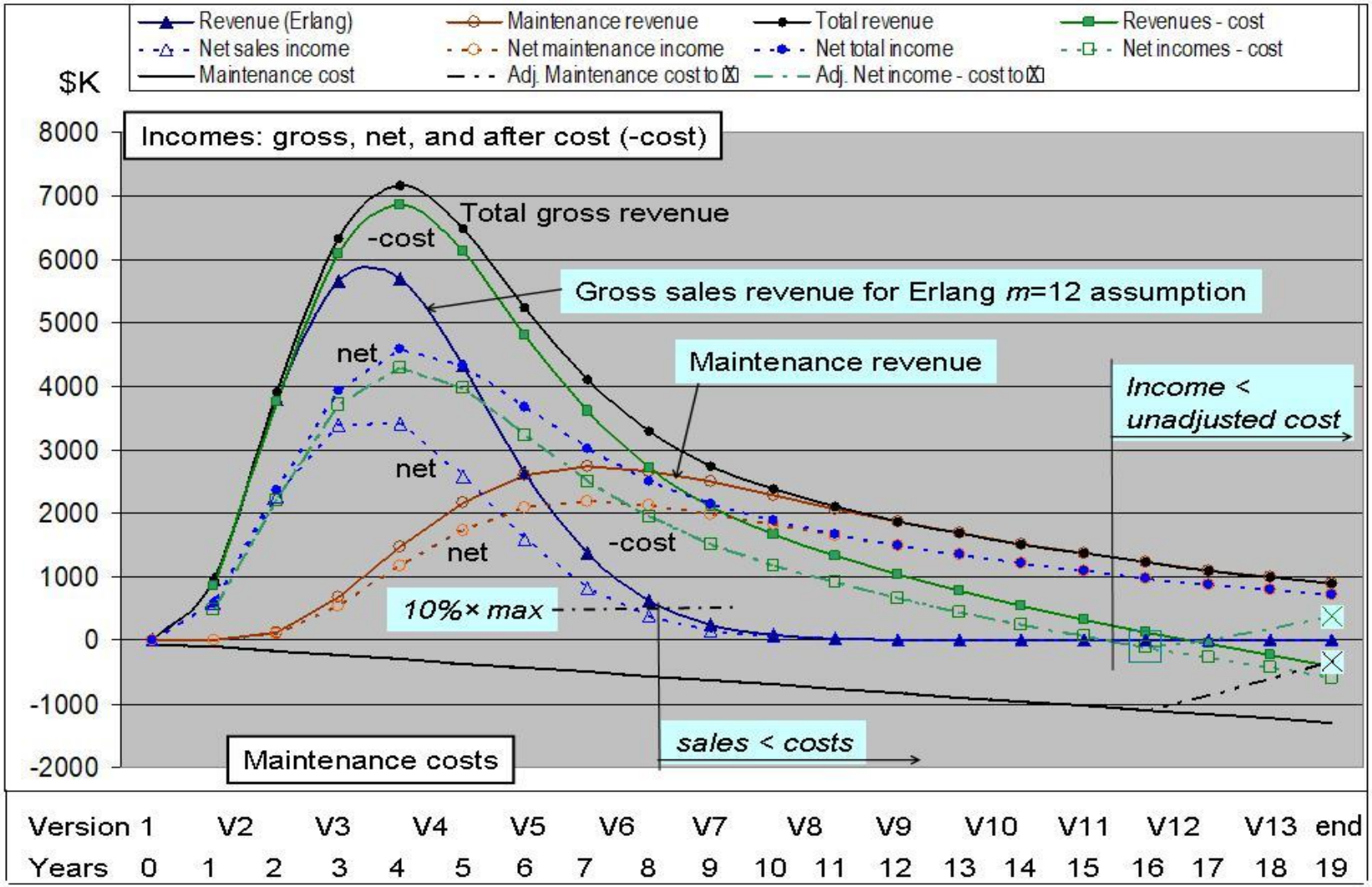
Service model factors

- Same proportion was used for SW contribution: 25%
  - Maintenance income has lower sales cost, perhaps more should be made available for software improvements
- Discount total only after maintenance cost
  - Income comes at time of spending
- Maintenance fees still generate substantial income
  - Organize business sector to collect those in out years
  - Use excess SW income for replacement or new products
- Continue longer, but stop in time!
  - When maintenance costs more than income

Net income, after sales cost



All Graphs


Revenue (Erlang)
Maintenance revenue
Total revenue
Revenues - cost
Net sales income
Net maintenance income
Net total income
Net incomes - cost
Maintenance cost
Adj. Maintenance cost to
Adj. Net income - cost to
$K
Incomes: gross, net, and after cost (-cost)
Total gross revenue
-cost
Gross sales revenue for Erlang m=12 assumption
net
Maintenance revenue
Income < unadjusted cost
net
net
-cost
10%× max
sales < costs
Maintenance costs
8000
7000
6000
5000
4000
3000
2000
1000
0
-1000
-2000
Version 1 V2 V3 V4 V5 V6 V7 V8 V9 V10 V11 V12 V13 end
Years 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19

Life of Software

We learned now why software has a finite life

Although SW can be indefinitely maintained

Eventually the maintenance costs exceed income

- A very well-selling product can have a long life
  1. Unique
  2. High quality
  3. Well maintained
- An easy to maintain product can have a long life
  1. Well designed
  2. Insulated from change by established standards

Conflict?

Freemium

Software is free

1. Charge for fancy version
2. Charge for upgrades (maintenance)
3. Charge for multi-user version
4. Charge for Internet sharing

no spreadsheet yet

Planning: Consistency in plans

When comparing business alternatives

- Give each choice the same chance

1. Temporal consistency
   - Computing versus communication
     - Local versus Cloud in 2012
       - *Skate to where the puck is going* [Gretsky]
2. Discount rate
3. Resource prices
   - Green alternatives
     - Benefits may depend on future price of oil –
       - if you assume future price = 3 x now, why not invest in oil instead

Example

Enterprise SW versus cloud

[Benioff:2009]

SIEBEL enterprise sales force management $

1. Price $1,500 per seat, at 200 users = 300,000
2. $54,000 for support (18%) /year, x 5 = 270,000
3. $1,200,000 consulting for installation = 1,200,000
4. $100,000 admin.personnel/year, x 6 = 600,000
5. $ 30,000 training / year, x 6 = 180,000

➢ 6 years' usage Total = 2,550,000

Note that the customer's total is >> than the price

Allocaction

- When there are multiple products
- When there are other contributors to income
  - Substantial hardware
  - Financial consultants in financial firms
  - Experts in call centers
  - Brand name

  Not all of the income can be allocated to the software
- Pareto Optimum

Pareto Optimality

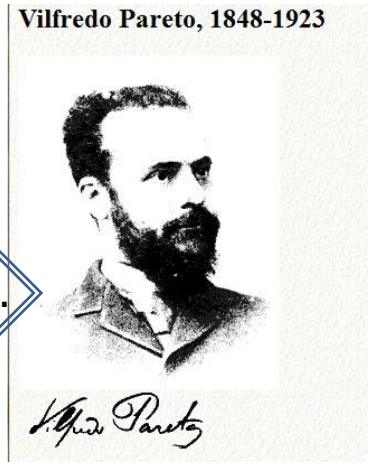
(not Pareto Efficiency : 80/20 rule)

The point were any change lowers the total benefit/cost

- Spending more on software will have less benefit than spending on other stuff
  - People
  - Hardware
  - Advertising
    - For large 10 IT companies the average value allocated to their brand name is 22% (BW survey).

1873

1870 startup: Rome Railway Co.



Conclusion:

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

Setting License fees

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 = Σ(due old, slide 5)
     - You will also maintain the SW 1.36M = Σ(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 *1./2.*= $150/unit
  - It does not matter how EUsales sells it and what it charges
  - Complexities are required language, interface improvements

Discussion

- Many choices now

a. Technical

b. Business

Interact with each other.

Review: Intangibles

- Software is an intangible good

  If it is owned it is considered Intangible Property

In a business there are 3 parts that have value.

(Contributes to potential income)

1. **Tangible goods**: buildings, computers, money
2. The **know-how** of management & employees
3. **Intellectual property**: Software, patents, etc.

   2. + 3. make up the **Intellectual Capital** of a company.

Recall: Discounting to NPV

Standard business procedure

- Net present Value (NPV) of

  getting funds 1 year later = F×(1 – discount %)

Standard values are available for many businesses

  based on risk (β) of business, typical 15%

Discounting strongly reduces effect of the far future

  NPV of $1.- in 9 years at 15% is $0.28

Also means that bad long-term assumptions have less effect

Making gadgets

Contains

a. Software
b. Firmware
c. Hardware components

Requires

1. Assembly
   induces delay
2. Inventory
3. Physical distribution

Versions
synchronized or
Tick-tock
Derived products

Changing model

- Versions
  - Synchronized
  - Tick-Tock

- Derived products

Quality

- Essential IP component
  - Product ► Trademark
- Disclaimers reduce trust in quality
- Cost avoided by disclaimers
  - more on that next week

Discussion

- Many parameters used to estimate IP
  - Uncertainty !
  - But better than not knowing what's going on.
- Many choices now

a. Technical options

b. Business options

Interact with each other.

Discussion: role of Government

Eugene Miya for NASA, NSF, DARPA

The Government giveth and payeth

- Rigid acquisition rules
  - No flexibilty as times and needs change
  - no ability to innovate, iterate to a desirable, but vague goal
  - lowest costbid preferred – huge incremental costs
- Single copy software for NASA, military
  - attempts to use Commercial-of-the-shelf
  - very high precision computation, possible, but different
  - reliability required, few in-flight fixes possible

5 October 2011 -2 years ago

“[Death is] life's change agent. It clears out the old to make way for the new.”