What Are Web Services Worth ?

Gio Wiederhold

Stanford University and MITRE Corp.

May 2005

www-db.stanford.edu/people/gio.html

A new type of Business

Services on the *Semantic Web**

- Inspiration for academics
- A hot topic for funders
- A motivation for technologists
- An opportunity for system integrators
- A puzzle for individual customers
- A dead-end for accountants
- A mystery for business economists
- Of undetermined value to investors

* Primarily a technology issue?

Software needed for Services

- Software Components
- Many interfaces

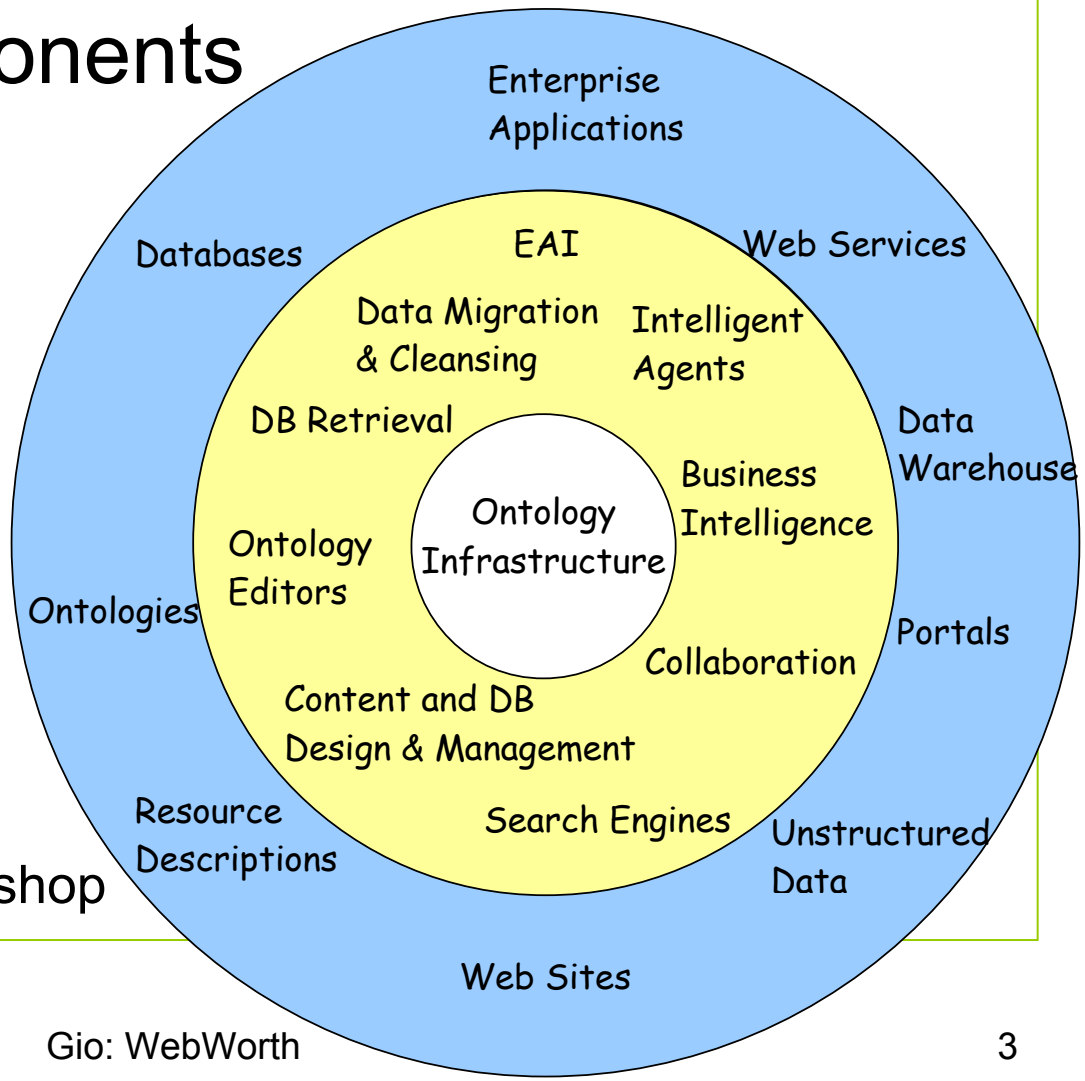
from

Avron Barr & Shirly Tessler:

Shared Semantics: A Briefing on the Emerging Market for Distributed Ontology Technology;

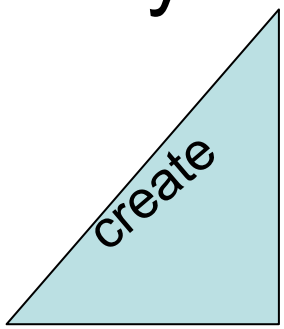
Aldo Ventures, Aptos CA, August 2002.

Based on ESWC 2001 workshop



Product Worth: **cost or income?**

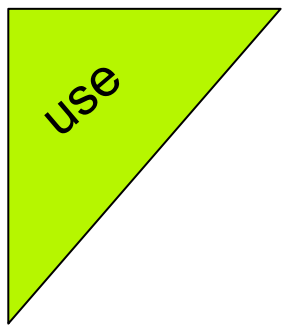
- Software producers traditionally care about
  - Cost of writing software
  - Time to complete products
  - Capabilities



life

- When the value is a concern, others play
  - Business people
  - Economists
  - Lawyers
  - Promoters

inconsistent



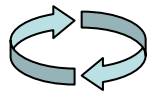
Why a concern for SW folk

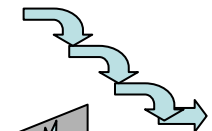
- Making decisions about creative tradeoffs
  - Elegance versus functionality
  - Rapid generation versus maintainability
  - Careful specification versus flexibility
- *Dealing with customers*

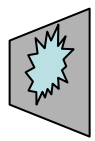
  Dijkstra model: *for self-satisfaction*

  Engineering model: *formal process driven*

  Startup model: *see if it sticks to the wall*







- Gain respect: *know what you are doing*

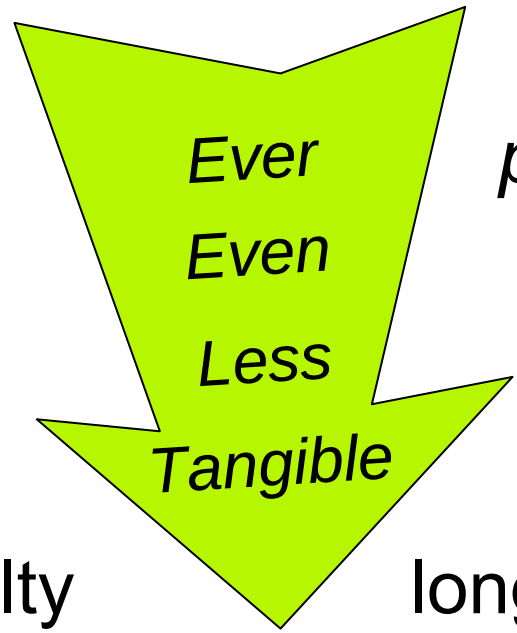
Why am I valuing SW now?

- Much software is being exported as part of **offshoring** (offshore outsourcing)
- It is typically property – i.e., *protected*
- If it is not valued correctly – i.e., *too low*
  1. Loss of income to the creators *in the USA*
  2. And loss of taxes *to the US* treasury
  3. Excessive profits *kept external to the USA*
  4. Increased motivation for external investment

SW is an Intangible

- Product of knowledge *by*
- Cost of original >> cost of copies

1. Books	*authors*
2. Software	*programmers*
3. Inventions	*engineers*
4. Trademarks	advertisers
5. Knowhow	managers
6. Customer Loyalty	long-term quality

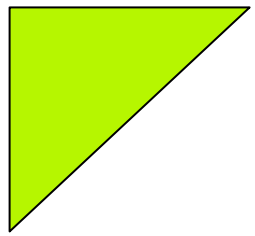


Valuation of intangibles

- **Principle**

  The sum of all future income discounted to today (NPV)

  Implicitly estimated by share holders



- Example: Value of a company *(SAP)*
  - Largely intangible – like many modern enterprises

1.	Market value = share price × no. of shares	€31.5B	100%
2.	Bookvalue – sum of all tangible assets Equipment, buildings, cash	€ 6.3B	20%
3.	**Intangible** value per stock market	€25.2B	80%

Intangible/tangible = 4

  - How much of it is software ?

Value of software today ...

- You have a great Web Service today.
- It is based on wonderful software.
- You are getting great customers.

Next Year, in 5 years, in 10 years

- Will your Web service be the same?
- Will your software be the same?
- Will your customers be happy?

Software is slithery !

Continuously updated

1. Corrective maintenance

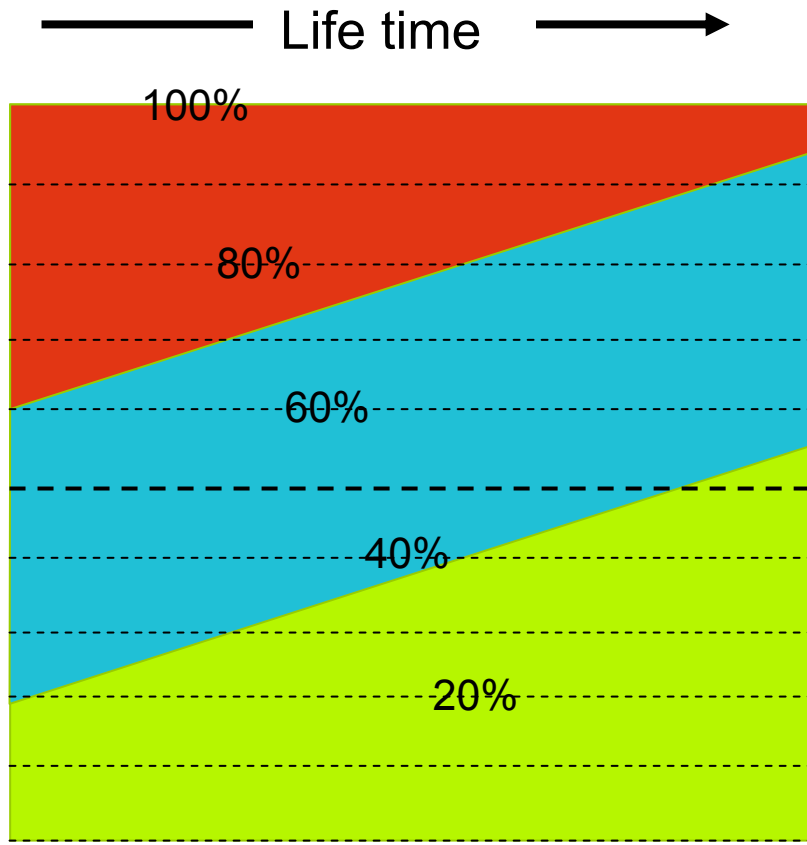
   bugfixing, reduces for good SW

2. Adaptive maintenance

   externally mandated, steady

3. Perfective maintenance

   satisfy customers' growing expectations, grows



Ratios differ in various settings, unknown for web services

IP sources

- Corrective maintenance
  - Feedback through error reporting mechanisms
    - Taking care of missed cases
- Adaptive maintenance
  - Staff to monitor externally imposed changes
    - Compliance with new communication standards
    - Compatibility with related services
- Perfective maintenance
  - Feedback through sales & marketing staff
    - Needed to keep and gain customers

Maintenance costs

- are substantial
  - About 20% of the initial cost per year
- cause the software to grow
  - Larger software is more costly to maintain
- prolong life of successful applications
  - 8 to 18 years, *but unknown for web services*
- dominate over the lifetime of a product
  - Maintenance cost is about 80%
- in time kill products
  - Innovation replaces them with more effective products and services

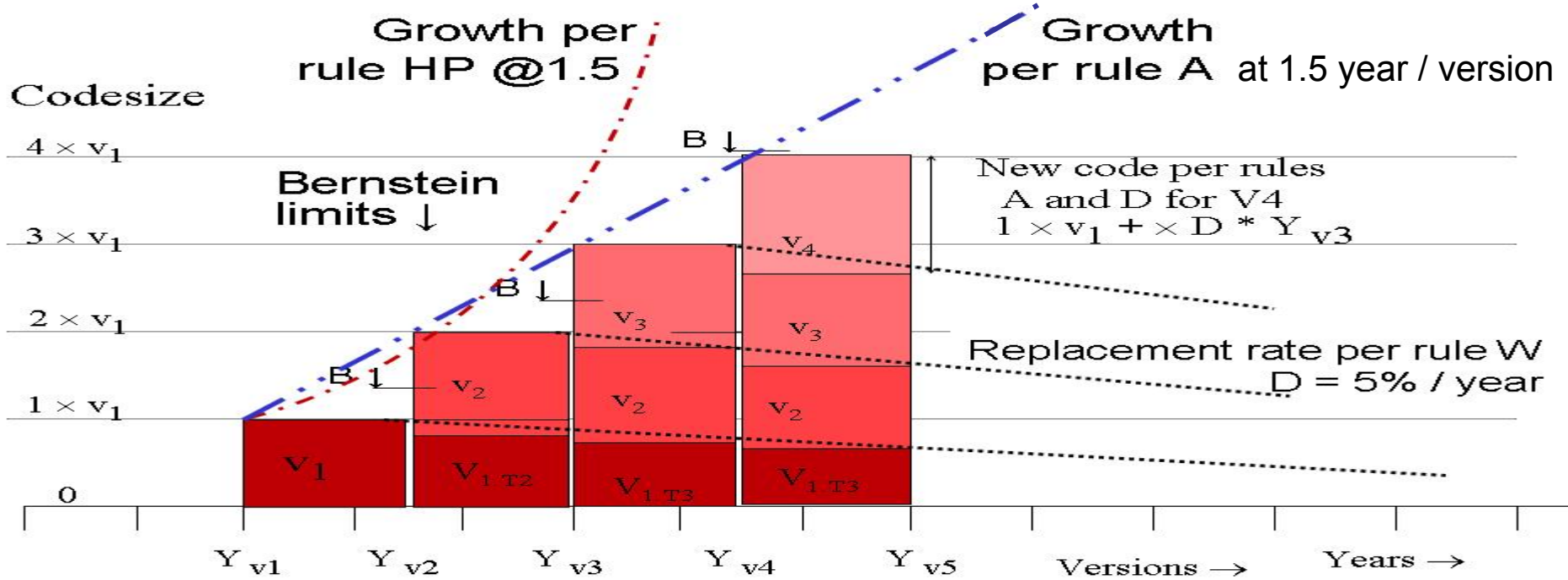
To value: quantify it all

Rules: S_{n+1} = 2 to 1.5 × S_n per year [HennesseyP:90]

$V_{n+1} \leq 1.30\% \times V_n$ [Bernstein:03]

$V_{n+1} = V_n + V_1$ [Roux:97]

Deletion of prior code = 5% per year [W:04]



Observations

- Software cannot grow exponentially

  no Moore's Law

  Because

  1. Cost of maintaining software grows exponentially [Brooks:95]
  2. Can't afford to hire staff at exponential $*^2$
  3. Cannot have large fraction of changes in a version
  4. Cannot impose version changes on users > 1 / year
  5. Deleting code is risky and of little benefit

     except in game / embedded code

Price *remember IP =f(income)*

- Price stays ≈ fixed over time

now like hardware

€ $ £ ¥ € $ £ ¥ € $ £ ¥ € $ £ ¥ € $ £ ¥ €

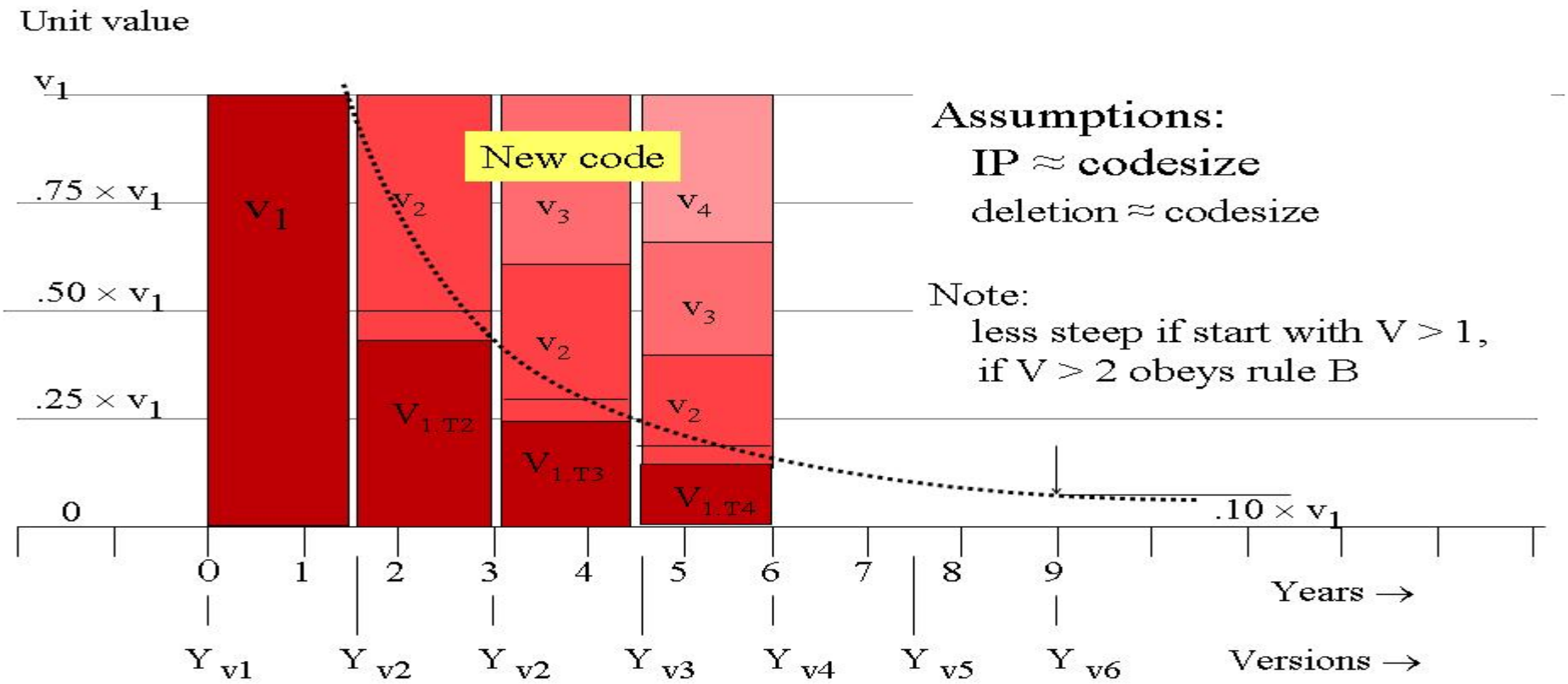
Because

1. Customers expect to pay same for same functionality
2. Keep new competitors out
3. Enterprise contracts are set at 15% of base price/an

- Effect

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

Growth diminishes IP



Total income

Total income = price × volume (year of life)

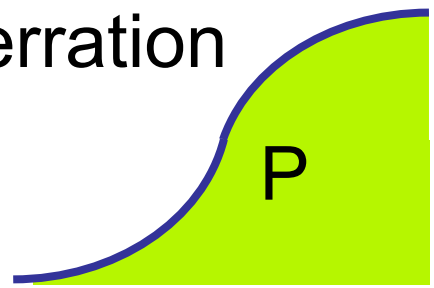
- Hence must estimate volume, lifetime

Best predictors are previous comparables

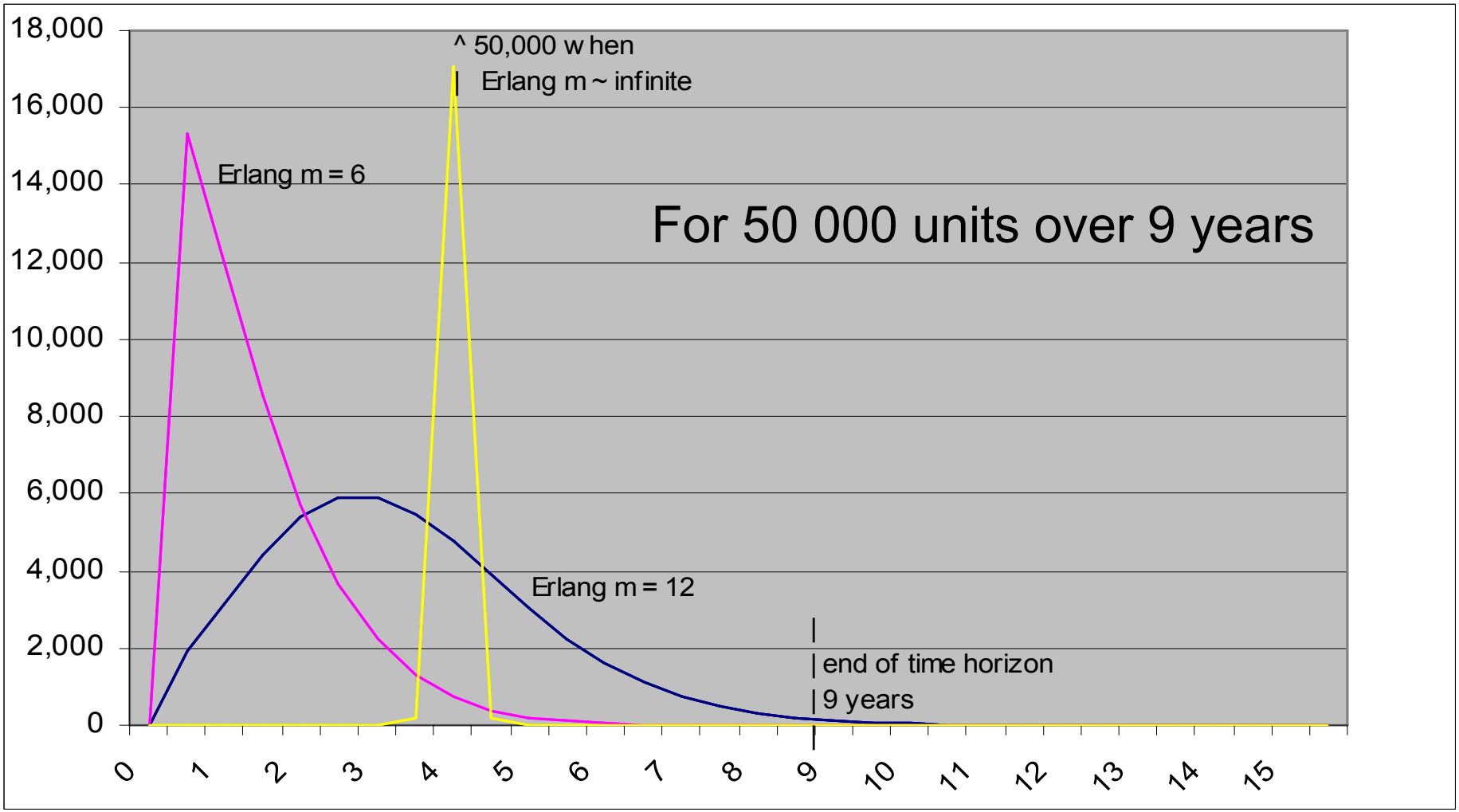
- ➢ Erlang curve fitting (m=6 to 20, 12 is typical)

and apply common sense limit = Penetration

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



Sales curves



Web services companies

1. Develop software *
   - Generates IP, but no income yet, no value yet
2. Use that software to provide services
   - Basis of IP: income from service sales
3. Lease that software to other service providers
   - Basis of IP: income from software sales
4. Purchase & license software for their use
   - Generates income based on IP obtained
5. Have software for their own internal use
   - No income, just cost benefits, have IP value

. . . and combinations .

Fraction of income for SW*

Income in a software company is used for

- Cost of capital *typical*
  - Dividends and interest ≈ 10%
- Routine operations -- not requiring IP
  - Distribution, administration, management ≈ 40%
- IP Generating Expenses (IGE)
  - Research and development, i.e., SW ≈ 25%
  - Advertising and marketing ≈ 25%

These numbers are available in annual reports or 10Ks

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

  NPV of $1.- in 9 years at 20% is $0.19

Also means that bad long-term assumptions have less effect

Combining it all

modified from paper

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
Service Pr	$0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50
Rel.size	1.00	1.67	2.33	3.00	3.67	4.33	5.00	5.67	6.33	7.00
Service Volume K		1911	7569	11306	11395	8644	2646	1370	1241	503
Revenue $K		955	3785	5653	5698	4322	2646	1370	621	252
Part for SW 25%		239	946	1413	1425	1081	662	342	155	63
Maintenance @ 20%		200	333	467	600	733	867	750	450	225
Gross income $K		755	3451	5186	5097	3589	1779	620	170	27
Maint /Av.SW Ratio		0.84	0.35	0.33	0.42	0.68	1.31	2.19	2.90	3.58
Disct 20%	1.00	0.83	0.70	0.58	0.48	0.40	0.34	0.28	0.23	0.19
SW cost	1,000	ratio maintenance/total = 83%						Actual		5,625
Gross Inc.	12,641	after discounting						Actual		25,300
Total	1,700	≈ $ 1.7 million								

Result of Example

- Selling 50M service units @ $0.50 ≈ $ 1.7M

  not $ 25.0M

Once its in a spreadsheet, the effect of the many assumptions made can be checked.

When assumptions later prove unwarranted then management can make corrections.

To be wise, don't spend more than ≈ $500 000 to develop the services software.

Business assumptions here

1. Original cost $1 000 000 (used to estimate 2.)
2. Maintenance cost 20%/year of aggregate cost
3. Maintenance fee 15%/year of original price
4. Cost to income lag 1 year
5. Stop maintenance when cost > income
6. Moderately risky business: discount rate 20%

   Average for SW industry is 15%, for startups 60%

Profits can be fed to founders & investors or reinvested in new business ventures

Software is an ongoing effort

Analysis shows importance of maintenance

Even poor numbers can help convince others.

- To achieve success in a service business
  1. Management must value maintenance
  2. Related standards, services must be monitored
  3. Marketing and sales must provide feedback
  4. Education and training must recognize the value of maintenance and maintainability

Software is poorly understood

1. Academics don't teach it

   It's barely in textbooks: 3/850 pages [Pressman:01]

2. Accountants ignore it
   1. It's classified as R&D, and hence looks optional
   2. It should be considered as Cost-of-Goods-Sold
   3. Leads to ridiculous margins: 99.9% for web services,
      Is 91-97% for much of the software industry

3. Companies assign maintenance to novices

   Experienced programmers should maintain their work
   and learn from it.

Knowing what software is worth

- Allows rational design decisions, as
  - Limiting development efforts
  - Programming investment for maintenance
- Allows rational business decisions, as
  - Choice of business model
  - Where and when to invest
  - How to assign programming talent
- Improve focus of education in software
  - Consider quality, not just quantity in assignments
  - Effectiveness of curriculum