



# CS207 #2, 2 Oct 2009

Gio Wiederhold

Gates B12



# Syllabus:

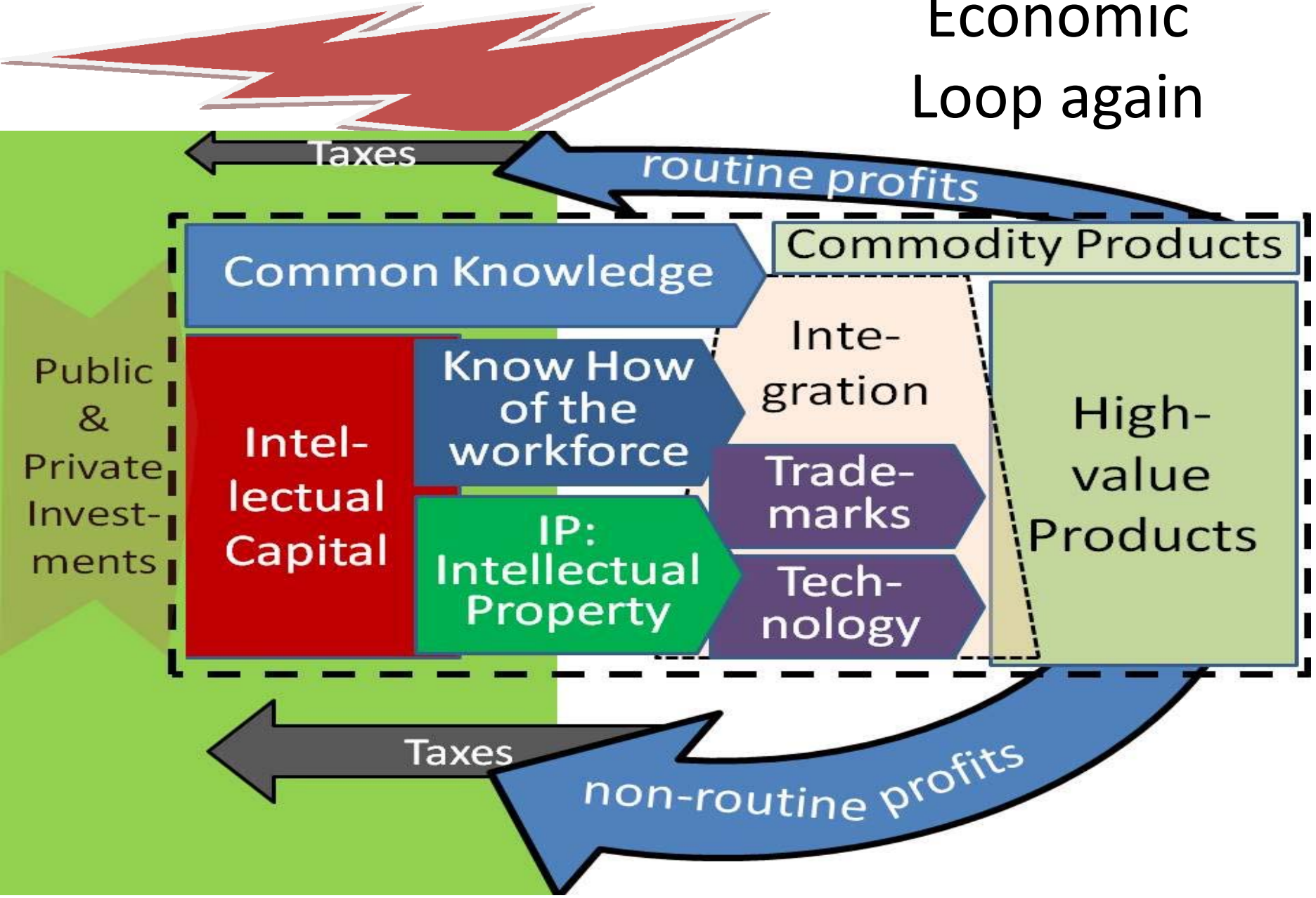
1. Why should software be valued?
2. Open source software. Scope. Theory and reality
3. Principles of valuation. Cost versus value.
4. Market value of software companies.
5. Intellectual capital and property (IP).
6. The role of patents, copyrights, and trade secrets.
7. Life and lag of software innovation.
8. Sales expectations and discounting.
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: Knowing what software is worth

- Allows rational design decisions, as
    - Allocating development efforts
    - Programming investment for long-lived SW
    - Understand limit to Software Life
  - 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

# Economic Loop again





# Value

Profit margins are the excess left after  
*CoGS* [Cost of Goods Sold] are deducted

## Conclusion from last week

- If goods are sold based on their manufacturing cost, there is no accounting for the value added due to their uniqueness.
- If anyone can compete profit margins will be modest.
- Uniqueness has value because it raises profit margins
- Uniqueness in software is not a tangible



# Quick definitions: Intangibles

In a business there are 3 parts that have value

(Contribute to potential income)

1. **Tangible goods:** buildings, computers, working capital
  2. The **know-how** of management & employees
  3. **Intellectual property:** Software, designs, methods, etc.
- 2. + 3. make up the **Intangible Capital** of a company.
  - Software is an intangible good

If it is ***owned*** it is considered Intangible **P**roperty

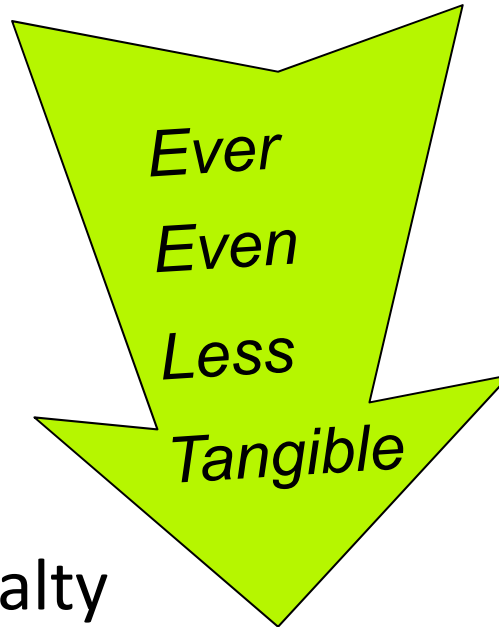


# Intangibles

- Product of knowledge

Cost of original >> cost of copies

1. Books
2. Software
3. Inventions
4. Trademarks
5. Knowhow
6. Customer Loyalty



*by*

*authors*

*programmers*

*engineers*

advertisers

managers

long-term quality



# Ownership

Claimed via

3. Patents

2. Copyright

1. Trade secret

More on those issues another day

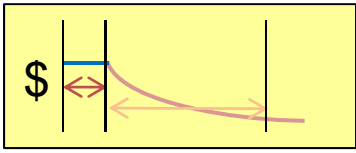


# Approaches to assess IP

- Technical alternatives

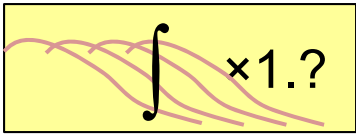
1. Income Prediction

Based on expected sales, life, lag



2. R&D spill-over

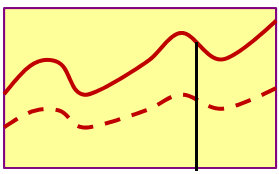
Based on life and effectiveness of R&D



- Broader alternative approaches

3. Market capitalization (Market Cap)

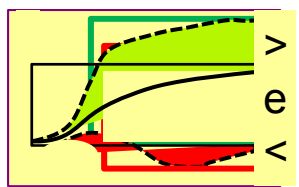
Covers everything the shareholders value



4. Real opportunity options (included in market cap?)

Increase value of IP for uncertain income

can expand market if > expected, cancel if < expected



# Fraction of intangibles

- **Principle**

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

*Implicitly estimated by share holders by the market cap*

- **Example: Market Cap value of a company (SAP, 2005)**

➤ Largely intangible – like many modern enterprises

- |   |        |      |
|---|--------|------|
| 1. Market cap = share price × no. of shares                             | €31.5B | 100% |
| 2. Bookvalue = sum of all tangible assets<br>Equipment, buildings, cash | € 6.3B | 20%  |
| 3. <b>Intangible</b> value per stock market                             | €25.2B | 80%  |

**Intangible/tangible = 4 x**

➤ How much of it is software at SAP ?



# Basis for SW value as of today

- Sum of future income
  - Sales = price x copy count
  - Maintenance fees if service subscription
- Minus sum of future costs
  - Cost of goods sold
  - Cost of marketing
  - Cost of doing business
  - Cost of maintenance
- Discounted to today
  - To account for value of money and risk

Independent of cost



# Discounting

- Standard economic accounting principle

*Getting \$1 next year is less valuable than getting \$1 today.*

1. *If no risk of getting it later, discount by available interest rate*
  - *Say 4%, 1-year off is \$0.96, 5-year is \$0.855, 15 year only \$0.542*
  - *Formally, use Federal bonds rates for that period*
2. *If there is a risk - likely in business – use risk experience*
  - *Say 15%+4%: 1-year is \$0.81, 5-year is \$0.349, 15 year only \$0.042*
  - *Tables per industry are available (at a price), based on past experience*

Discounting has a large effect on income estimates



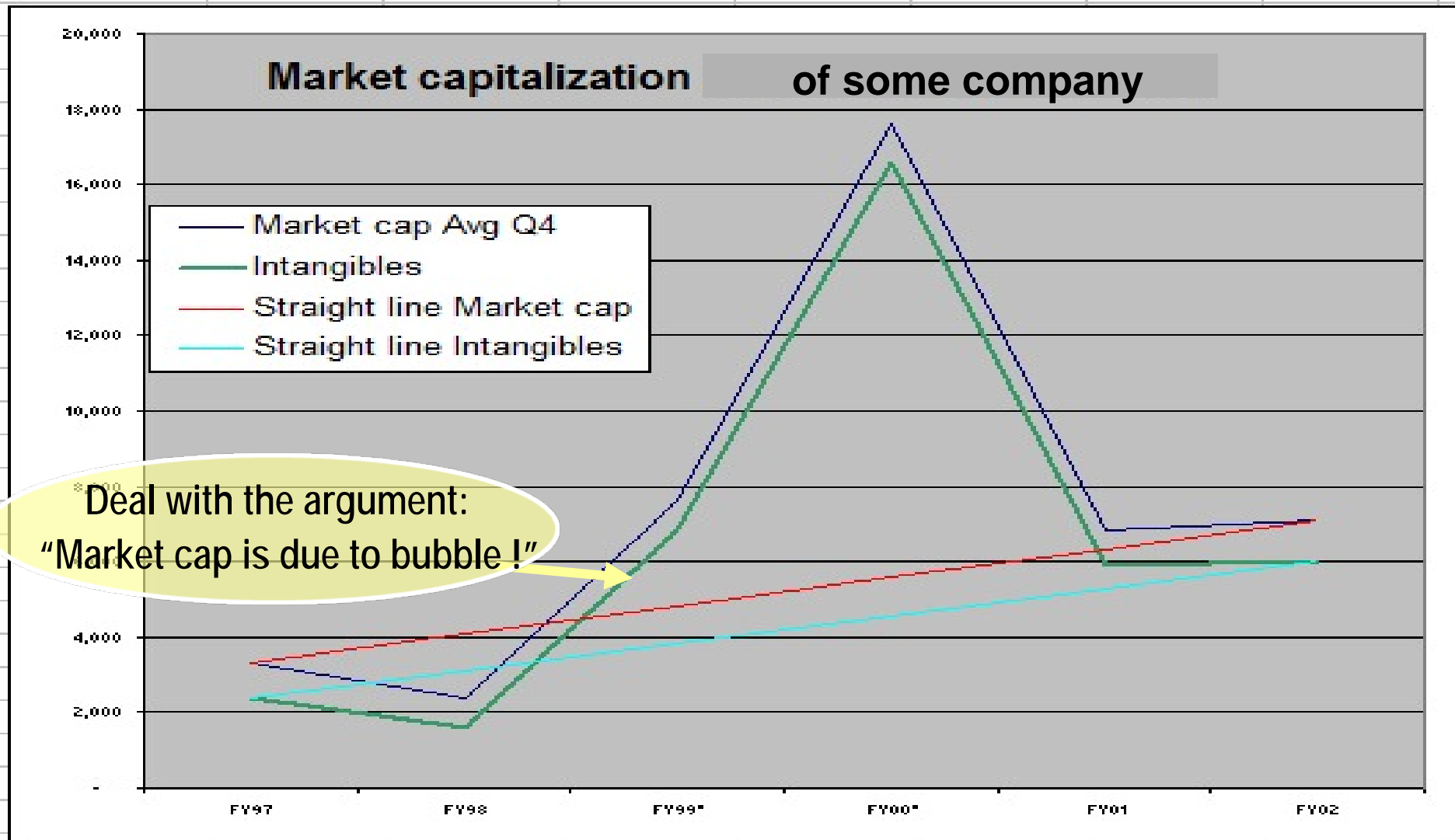
## Market cap : only a hint

### Issues

- Stockholders don't know what is really going on
  - Wisdom of the masses?
  - Are fed limited information
  - Indirect indicators are delayed: sales by principals
- Market cap is unreliable due to high variability
  - Market bubbles mislead
  - Option values are hard to judge
- In a multi-product company
  - Allocate income to each product line

**Over time, many factors should even out**

# For that hint: Adjust market cap



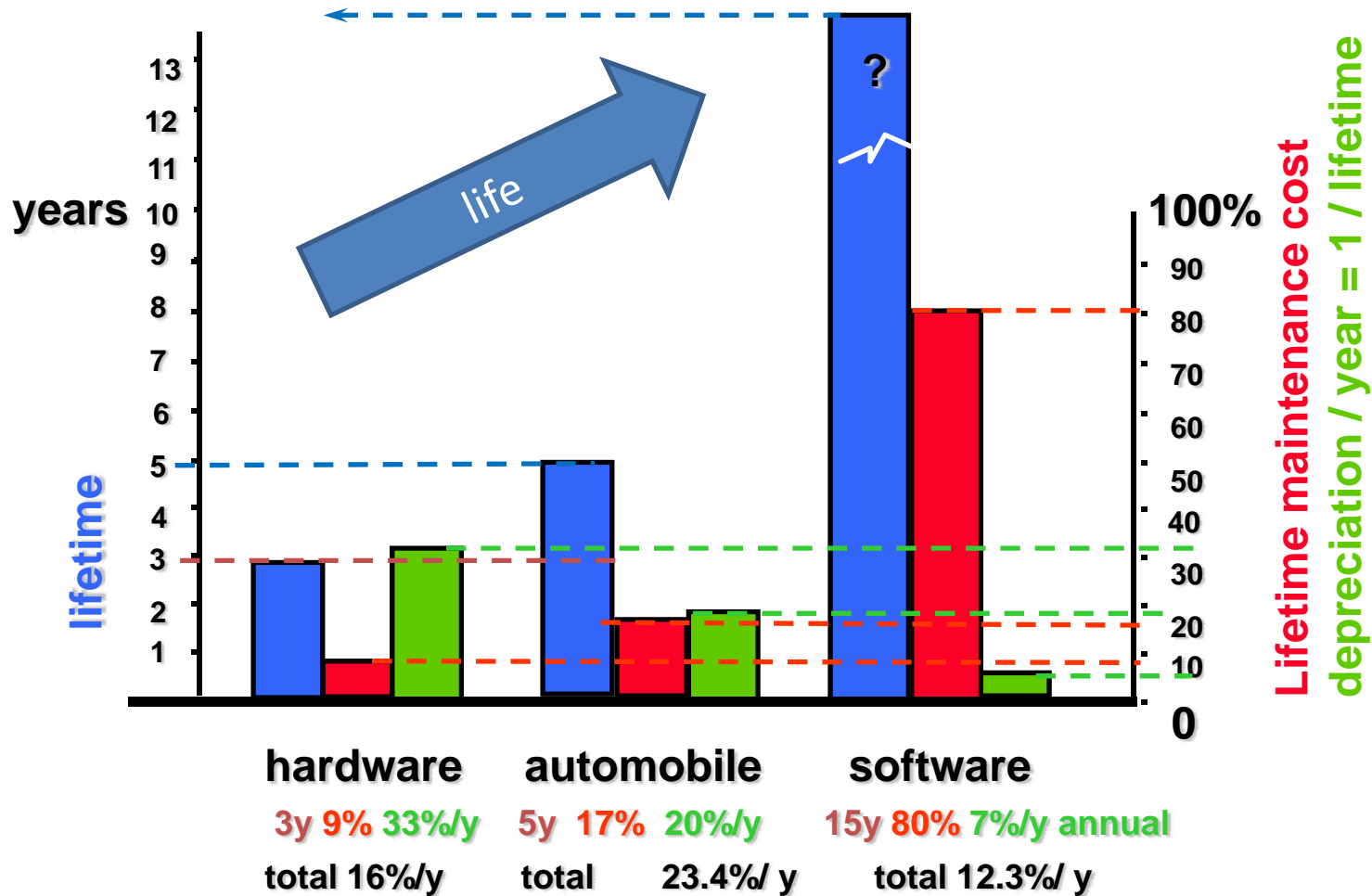


## A better, direct & new approach

- Value the software specifically by income over its lifetime
- But software is not stable over time: *Slithery*
  - Getting long-term income requires maintenance
  - Maintenance enables long-term income
- Much more so than other intangibles
  - Books, music,
- Similar to some intangibles that contribute to life
  - Customer loyalty, trademarks



# Maintenance is beneficial







# Software is slithery !

Continuously updated

1. Corrective maintenance

*bugfixing reduces for good SW*

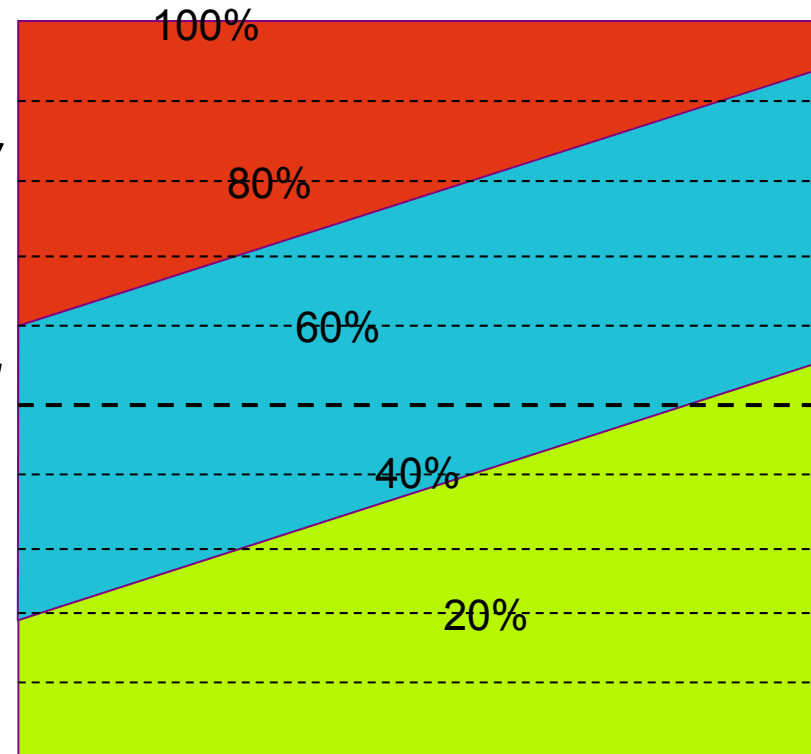
2. Adaptive maintenance

*externally mandated*

3. Perfective maintenance

*satisfy customers' growing expectations*

————— Life time —————>






[IEEE definitions]

**Ratios differ in various settings**



# IP sources

- **Corrective maintenance** 
  - Feedback through error reporting mechanisms
    - Inadequate protection from virus etc.
    - Taking care of missed cases
    - Complete inadequate tables and dimensions
- **Adaptive maintenance** 
  - Staff to monitor externally imposed changes
    - Compliance with new standards
    - Technological advances
- **Perfective maintenance** 
  - Feedback through sales & marketing staff
    - Minor features that cannot be charged for



# 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\***, interval from transfer to start of IP diminution

= the time before an investment earns revenue

- also called "Gestation Period"

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





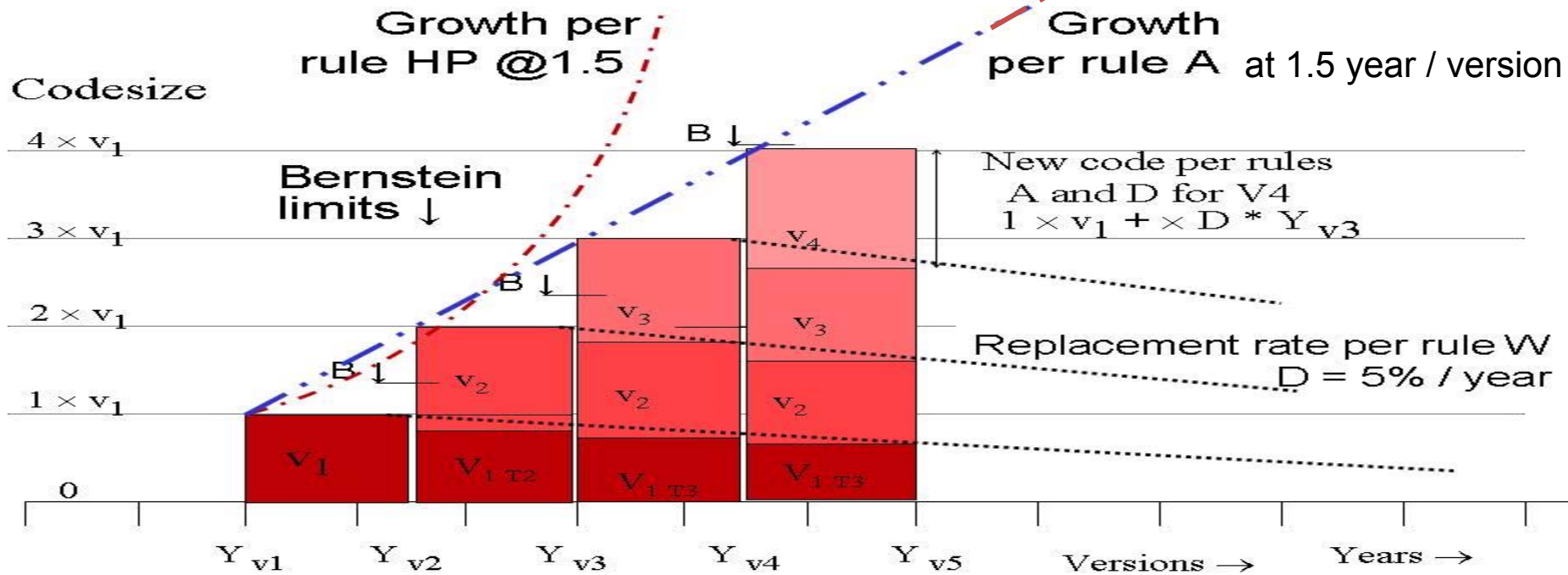
# Effect: SW Growth

Rules:  $s_{n+1} = 2 \text{ to } 1.5 \times S_n$  per year [HennesseyP:90]

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

$V_{n+1} = V_n + V_1$  [Roux:97] ([BeladyL72], [Tamai:92,02] indications) [Blum:98]

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  $< 1 / \text{year}$
5. Deleting code is risky and of little benefit

except in game / embedded code



# Price

*remember  $IP = f(\text{income})$*

- Price stays  $\approx$  fixed over time

like hardware Moore's Law

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
4. Shrink-wrapped versions can be skipped

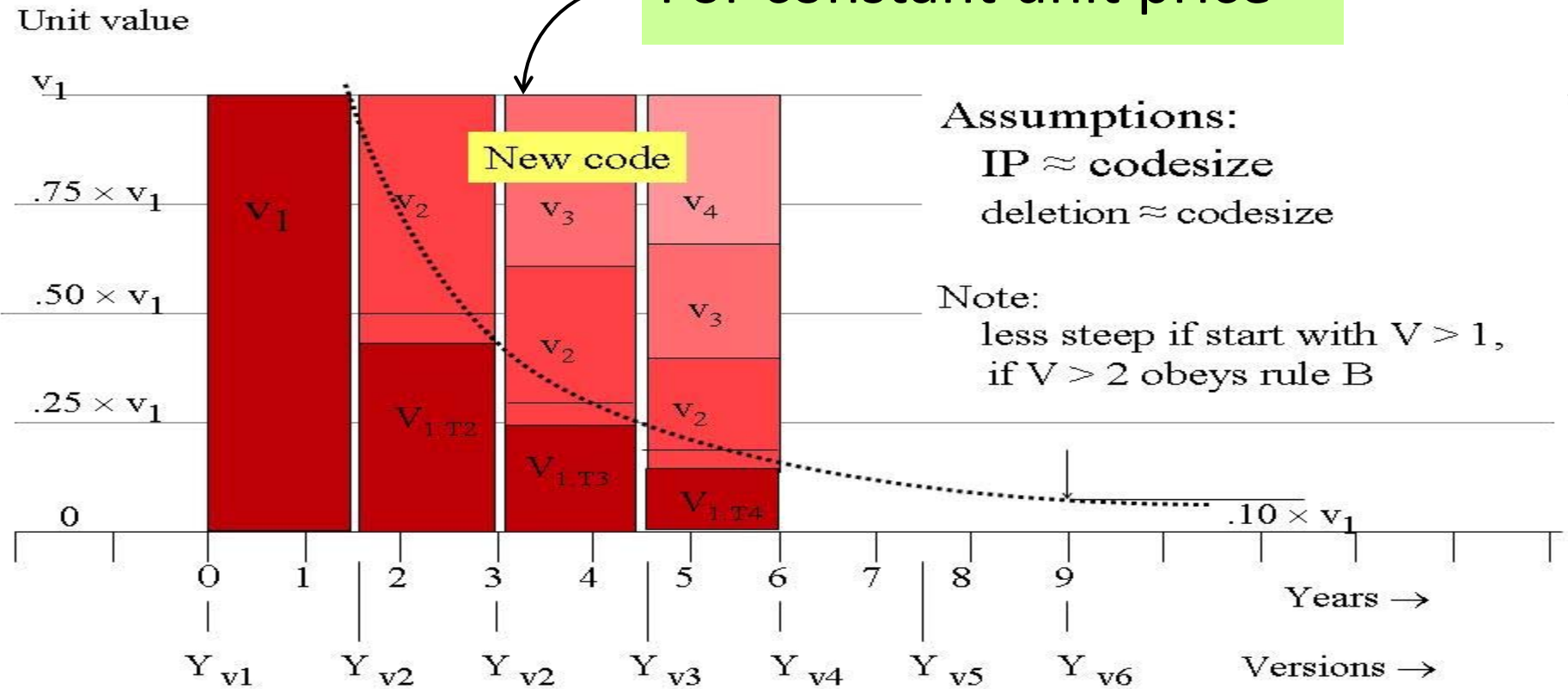
- Effect

**The income per unit of code reduces by  $1 / \text{size}$**

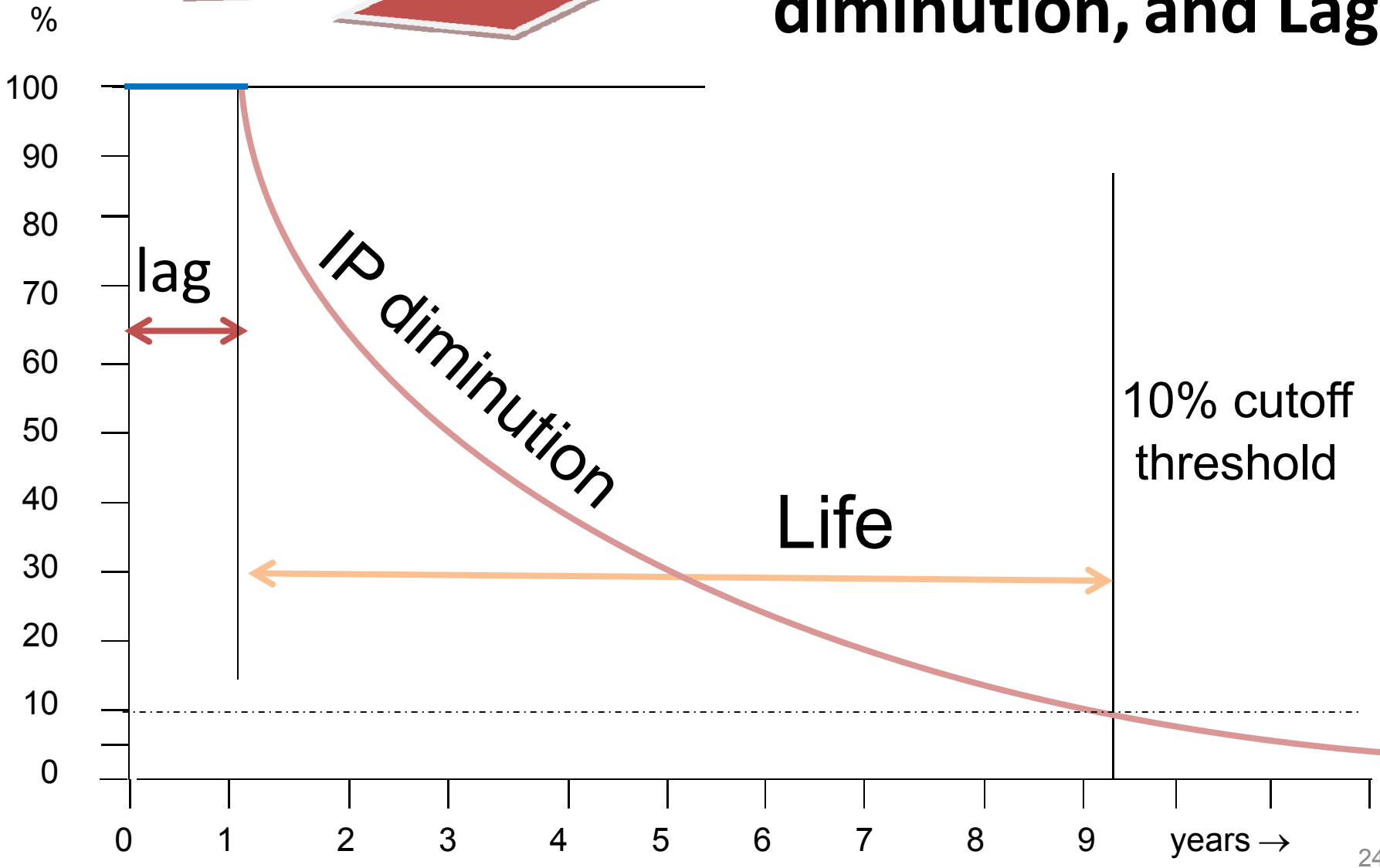
# Code Growth diminishes IP



For constant unit price



# IP Life, IP diminution, and Lag







# Lag

(snake in the grass)

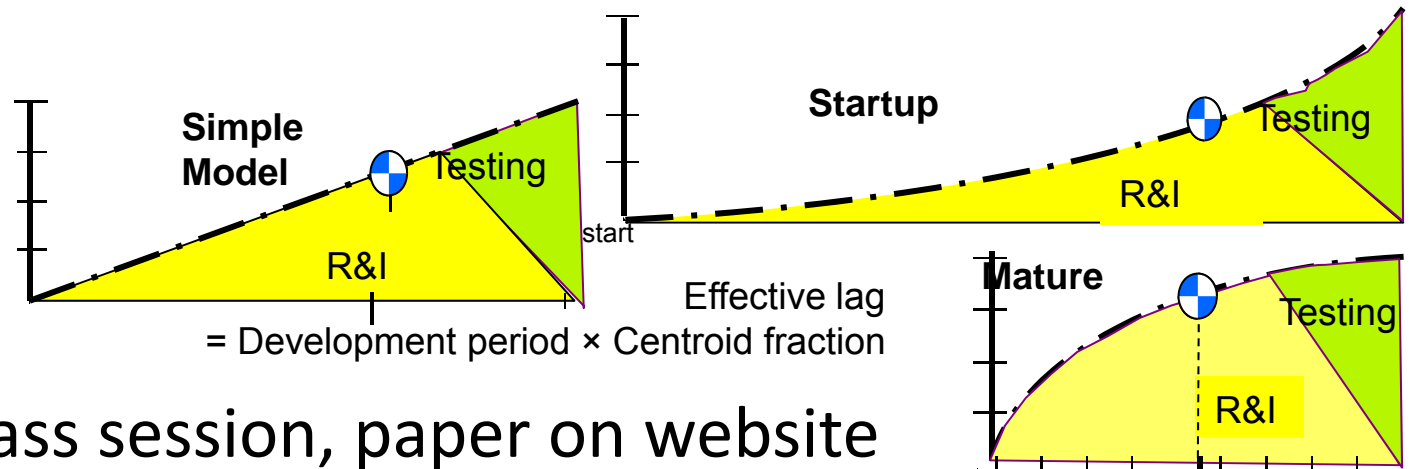
Lag is the effective development period

Has a large effect on early valuations

1. No income during that time
2. A chance for others to overtake you

*We assume a lag of 2 years in the examples.*

Depends  
on product  
development  
strategy



Separate class session, paper on website



## Next Class

- Still need to combine it with income
- Cost of maintenance