Software Development

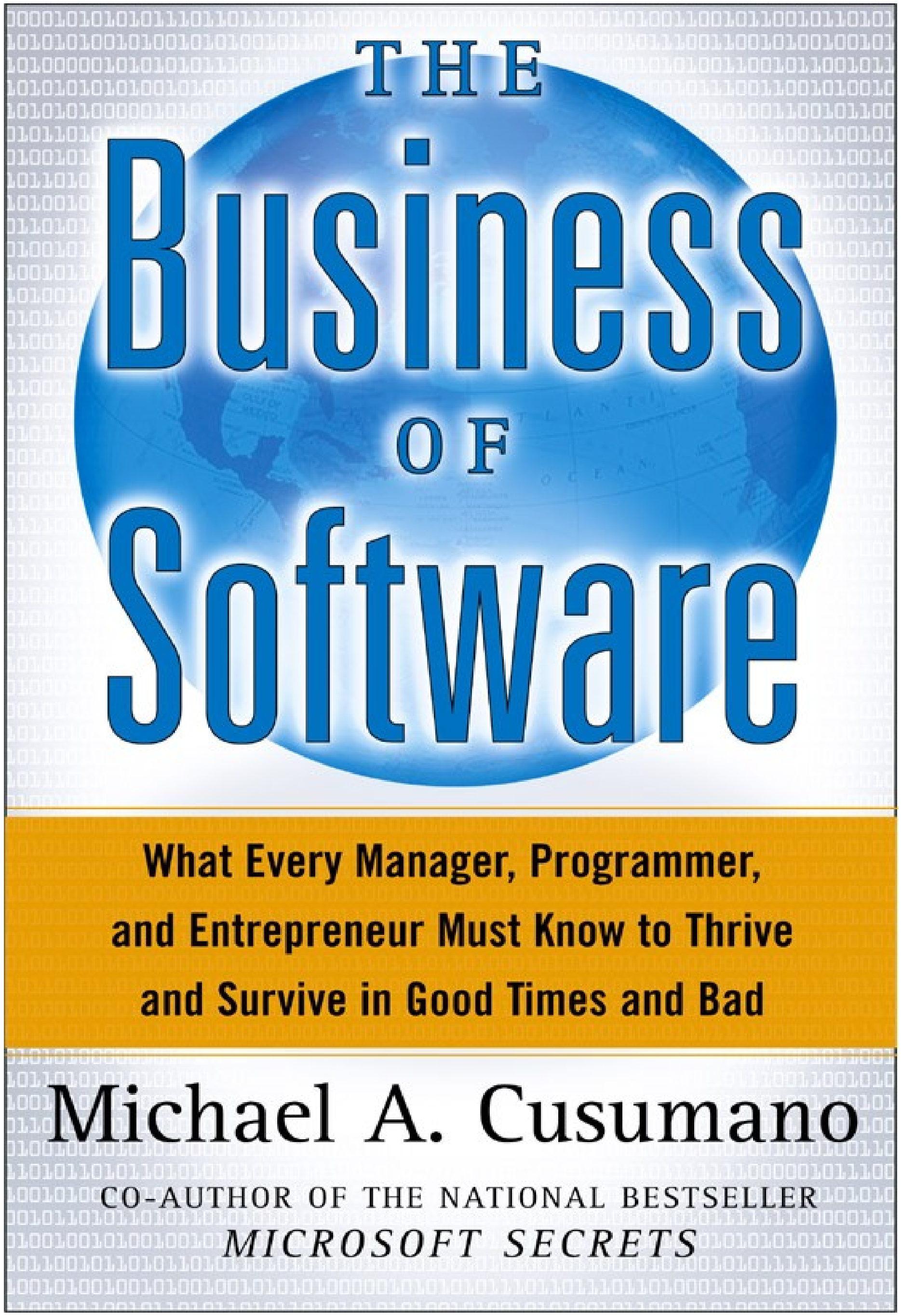
The State of the Practice and the Business

Michael A. Cusumano

MIT Sloan School of Management,

Cambridge, MA USA

cusumano@mit.edu



THE Business OF Software

What Every Manager, Programmer, and Entrepreneur Must Know to Thrive and Survive in Good Times and Bad

Michael A. Cusumano

CO-AUTHOR OF THE NATIONAL BESTSELLER *MICROSOFT SECRETS*

(My) International Perspective on the Software Business

EUROPE: Software as ***Science***

- –*Formal Methods, Object-Oriented Design*

JAPAN: Software as ***Production***

- –*Software Factories, Zero-Defects*

The USA: Software as a ***Business***

- –*Windows, Office, Netscape Navigator, $$$$*

INDIA

- Software as a *Service*
- *Service* as a *Business*

RUSSIA??

- What products?
- What process?
- How much service vs. products?

Agenda

- **Software Process:**

  Transition from Waterfall to Iterative

- **Software Business:**

  Transition from Products to Services

Problems in Software Development

- Similar problems recurring since the 1960s
- **1969 NATO Report on Software Engineering:**

  Documented problems in

  - requirements, design vs. coding separation
  - estimates, monitoring progress, communication
  - productivity (26:1), metrics, reliability (bugs)
  - hardware dependencies, reuse
  - maintenance costs
- **Sound familiar??**

Solutions to Problems

- **Many attempts at solutions**
  - IBM-style software engineering (1960s, 1970s)
  - Japanese “software factories” (1970s, 1980s – stable teams, standard process & tools, reuse)
  - SEI Capabilities Maturity Model (1980s to present)
  - “Iterative” methods

- **No one process perfect for all software projects**
  - Variations: business models, customer requirements, application domain, competition, pace of change, etc.

- **How balance quality, flexibility, cost & speed?**

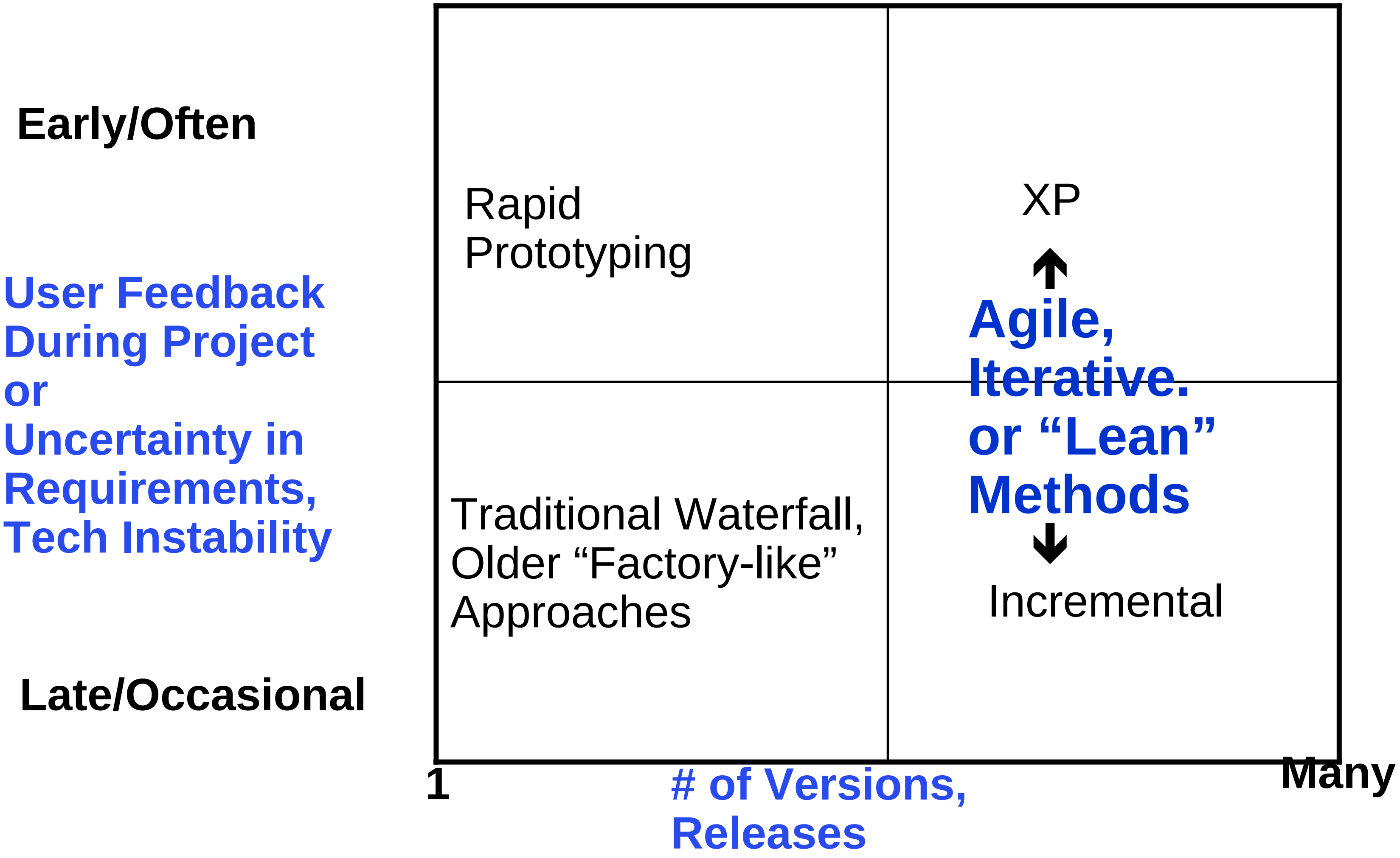
Different Process Philosophies

- **Waterfall-style (sequential, “Stage-gate”)**

  versus

- **Iterative-style (flexible, evolutionary)**
  - Spiral
  - Rapid Prototyping
  - Synch-and-Stabilize
  - HP’s Evo Process (short cycles of mini-waterfalls)
  - Extreme Programming (XP)
  - Many other variations at companies

In Reality:
A SPECTRUM of Approaches



Early/Often

User Feedback
During Project
or
Uncertainty in
Requirements,
Tech Instability

Late/Occasional

Rapid
Prototyping

XP

Agile,
Iterative.
or “Lean”
Methods

Incremental

Traditional Waterfall,
Older “Factory-like”
Approaches

1

# of Versions,
Releases

Many

Adapted from Bill Crandall (HP)

International Comparisons

(2003 IEEE Software article)

- **Survey:** Completed in 2002-2003, with Alan MacCormack (HBS), Chris Kemerer (Pittsburgh), and Bill Crandall (HP)
- **Objective:** Determine usage of Synch-and-Stabilize versus Waterfall-ish techniques, with performance comparisons
  - 118 projects plus 30 from HP-Agilent for pilot survey
- **Participants**
  - **India:** Motorola MEI, Infosys, Tata, Patni
  - **Japan:** Hitachi, NEC, IBM Japan, NTT Data, SRA, Matsushita, Omron, Fuji Xerox, Olympus
  - **US:** IBM, HP, Agilent, Microsoft, Siebel, AT&T, Fidelity, Merrill Lynch, Lockheed Martin, TRW, Micron Tech
  - **Europe:** Siemens, Nokia, Business Objects

“Conventional” Good Practices

		India	***Japan***	***USA***	***Europe etc***		***Total***
Number of Projects		*24*	*27*	*31*	*22*		*104*
Architectural Specs %		83%	70%	55%	73%		69%
Functional Specs %		96%	93%	74%	82%		86%
Detailed Design %		100%	85%	32%	68%		69%
Code Generators -- Yes		63%	41%	52%	55%		52%
Design Reviews -- Yes		100%	100%	77%	77%		88%
Code Reviews -- Yes		96%	74%	71%	82%		80%

“Newer” Iterative Practices

		India	***Japan***	***USA***	***Europe etc***		***Total***
No. of Projects		*24*	*27*	*31*	*22*		*104*
Subcycles -- Yes		79%	44%	55%	86%		64%
Beta tests -- Yes		67%	67%	77%	82%		73%
Pair Testing -- Yes		54%	44%	35%	32%		41%
Pair Programmer -- Yes		58%	22%	36%	27%		35%
Daily Builds at project start		17%	22%	36%	9%		22%
In the middle		13%	26%	29%	27%		24%
At the end		29%	37%	36%	41%		36%
Regression test each build		92%	96%	71%	77%		84%

“Crude” Output Comparisons

		India	**Japan**	**USA**	**Europe etc.**		***TOTAL***
Projects		24	27	31	22		***104***
LOC/ Month	median	209	469 cf. *389* in 1990	270 cf. *245* in 1990	436		***374***
Bugs/ 1000 LOC	median	.263	**.020** cf. .20 in 1990	.400 cf. .80 in 1990	.225		***.150***

Some Global Observations

- Most projects (64%) not pure waterfall; 36% were!
- Mix of "conventional" and "iterative" common -- use of functional specs, design & code reviews, but with subcycles, regression tests on frequent builds
- Customer-reported defects improved -- over past decade in US and Japan; LOC "productivity" may have improved a little, but unclear
- Japanese projects still report best quality -- but what does this mean? Preoccupation with "zero defects"?
- Indian projects look strong in process and quality -- but not as strong as CMM Level 5 suggests??

Hewlett Packard Pilot Study

(2003 IEEE article)

- **Managers -- When to use iterative vs. waterfall?**
- Survey: 35 responses, 29 projects with complete data
- Median – 170K LOC, with 70K new code; 9-person team, 14 month projects
- 59% applications, 38% systems, 28% embedded
- **74% of variation in defects explained by early prototypes, design reviews, integration/regression testing on builds**
  - Median project -- 40% of functionality complete when first prototype released and 35.6 defects per million (.04/1000) LOC, reported by customers in 12 months after release, and 18 LOC per person day (360/month)

Multivariate Regression Analysis

Some striking results:

- Releasing prototype earlier with 20% of functionality **= 27% reduction in defect rate (compared to median project)**
- Integration/regression testing at each code check-in **= 36% reduction in defect rate (cf. the median)**
- Design reviews **= 55% reduction in defect rate**
- Releasing prototype with 20% of functionality **= 35% rise in LOC output/programmer**
- Daily builds **= 93% rise in LOC output/programmer**

Observations from HP Survey

- **Best “nominal” quality** from traditional “waterfall” (fewer cycles & late changes = less bugs, of course!!)
- **Best balance** of quality, flexibility, cost & speed from combining conventional & iterative practices

- ***BUT: Differences in quality between waterfall & iterative disappear with a bundle of techniques:***
  - Short development subcycles (subprojects/milestones)
  - Early prototypes to get customer feedback
  - Frequent builds to incorporate feedback, changes
  - Frequent design/code reviews (check quality continuously)
  - **Regression tests on each build (check for errors, late changes, integration problems)**

Waterfall vs. Iterative

- **Waterfall** still common; question is when to use this approach or for what parts of a project?
- **Iterative** now more common; question is how to control degree or timing of changes?
- **Process strategy** should differ based on many factors (requirements, experience, etc.)
- **Product or service** not determining factor; both standardized products and custom systems usually require multiple iterations to get the design right

Main References

- **The Business of Software** by M. Cusumano (Free Press/Simon & Schuster, 2004)

- Michael Cusumano, Alan MacCormack, Chris Kemerer, and Bill Crandall, **"Software Development Worldwide: The State of the Practice",** IEEE Software, November-December 2003. (International Comparisons)

- Alan MacCormack, Chris Kemerer, Michael Cusumano, and Bill Crandall, **"Trade-offs between Productivity and Quality in Selecting Software Development Practices",** IEEE Software, September-October 2003. (HP Survey)

The Software Business:

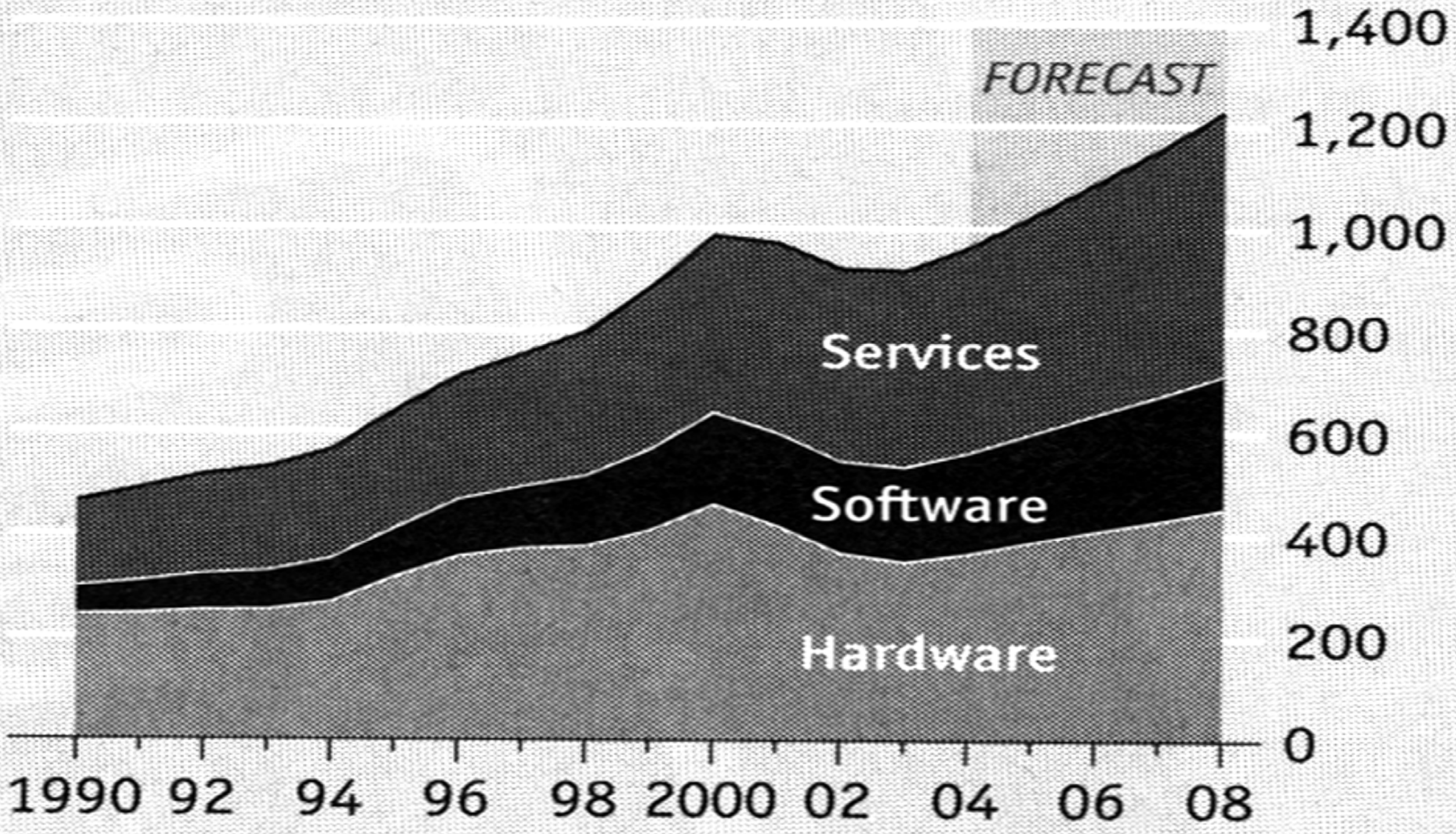
Products AND Services, Big AND Global

- $700+ billion in worldwide revenues
- About 1/3 products, 2/3 services
- 35,000 firms worldwide with ≥ 5 employees
- North America – 50%
- Europe – 30%
- Asia – 15%
- Top software producers: IBM, Microsoft, EDS, Accenture, Oracle, HP, NTT, SAP

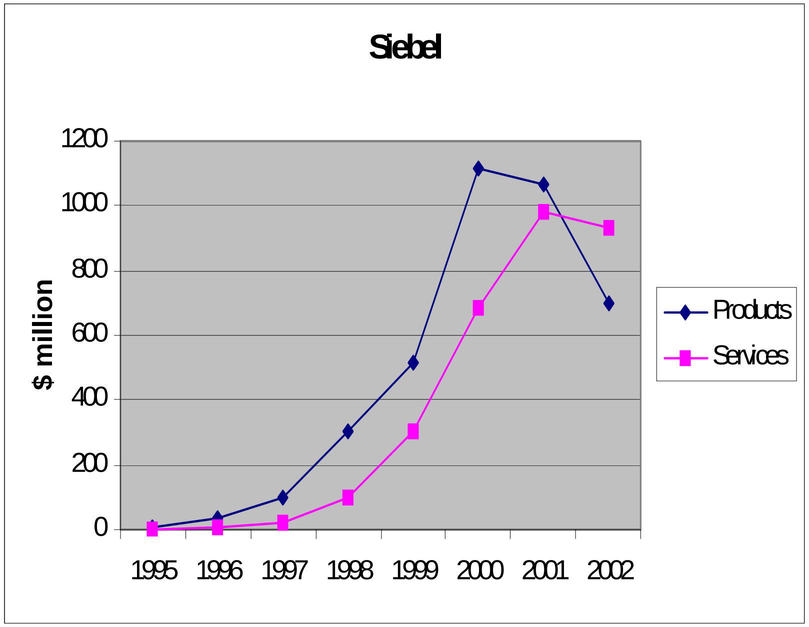
1

The kink in the middle

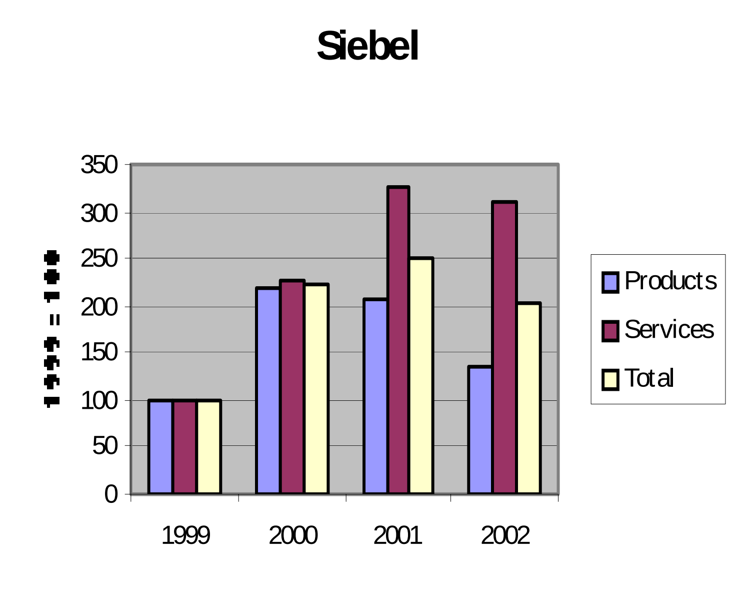
Total IT spending, $bn



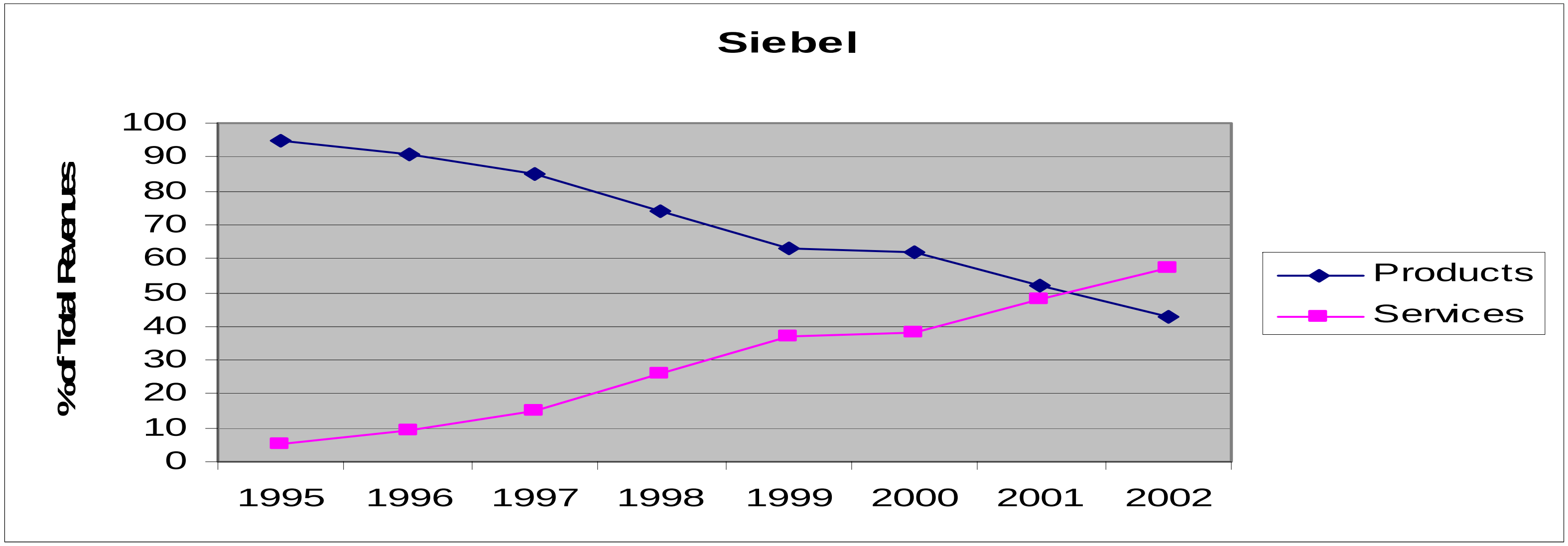
Source: IDC



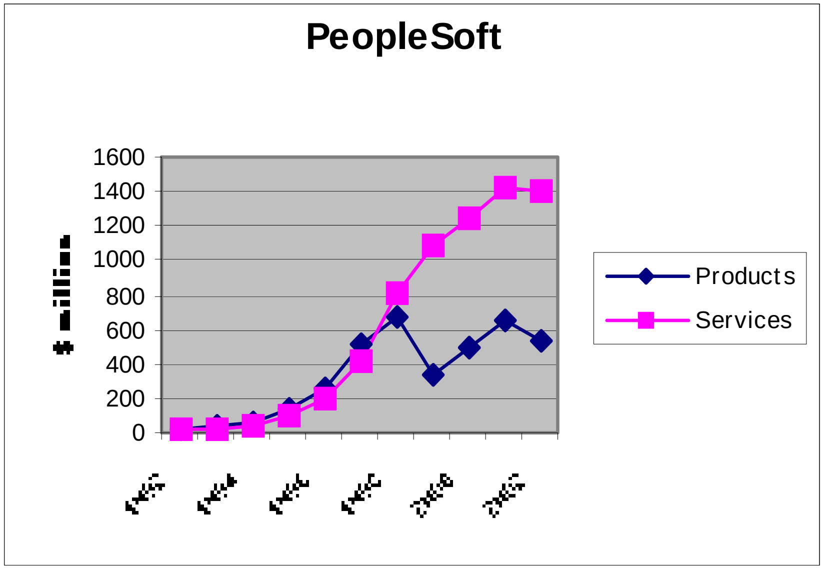
Siebel

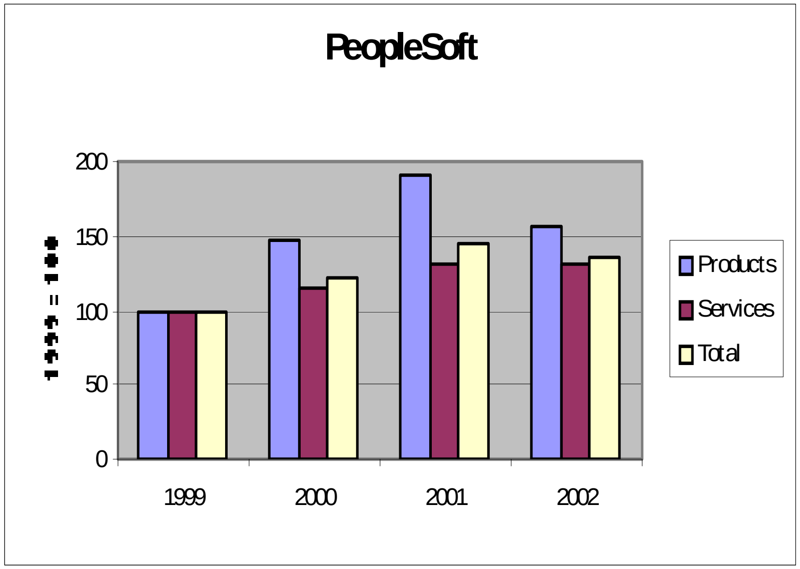


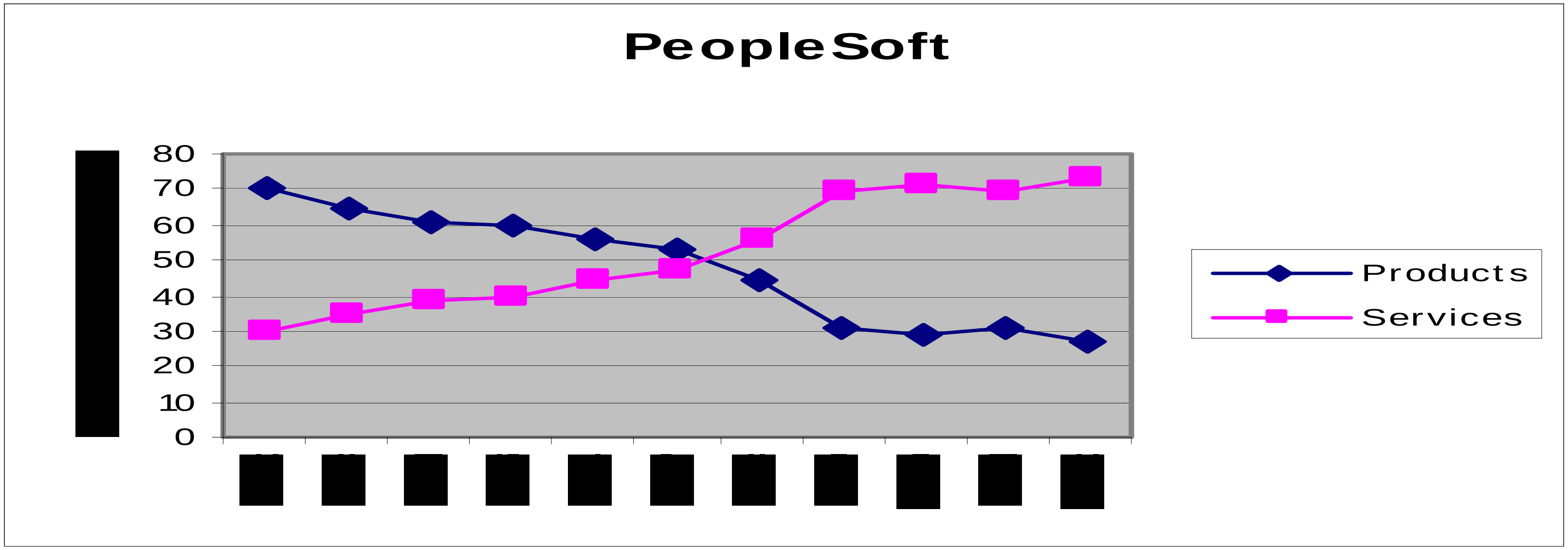
Siebel

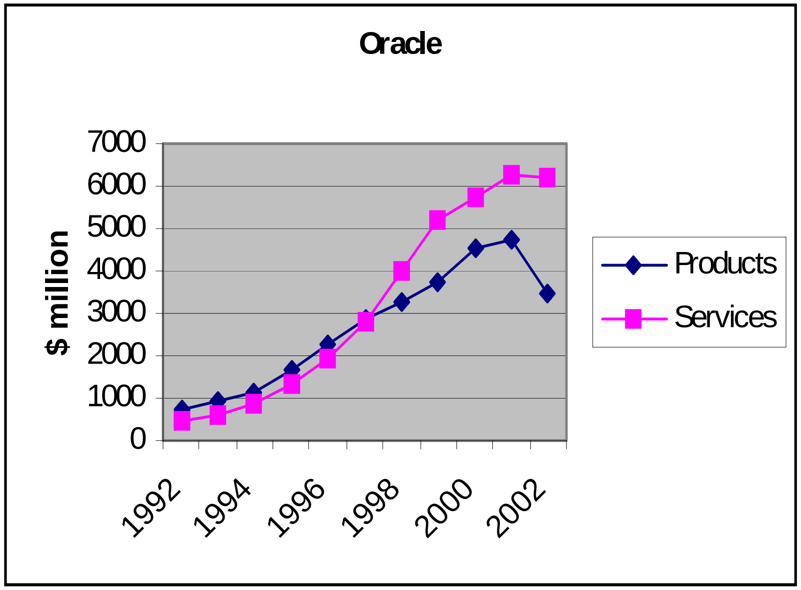


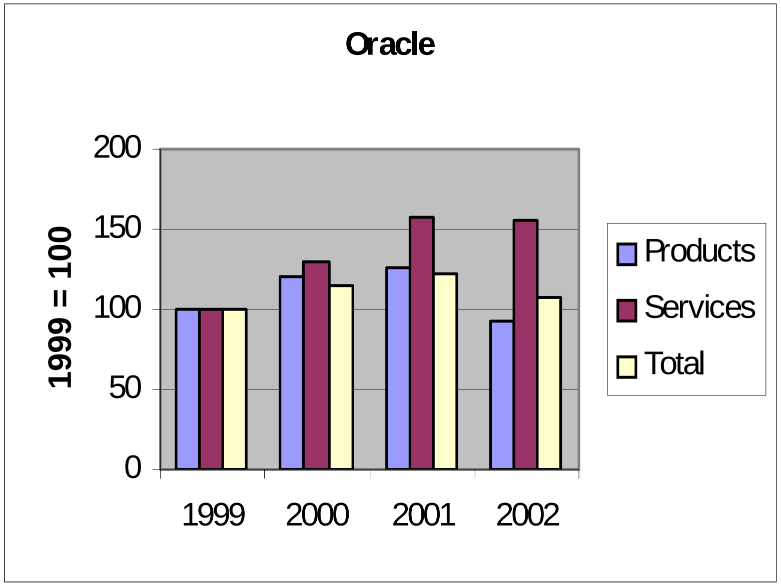
Siebel

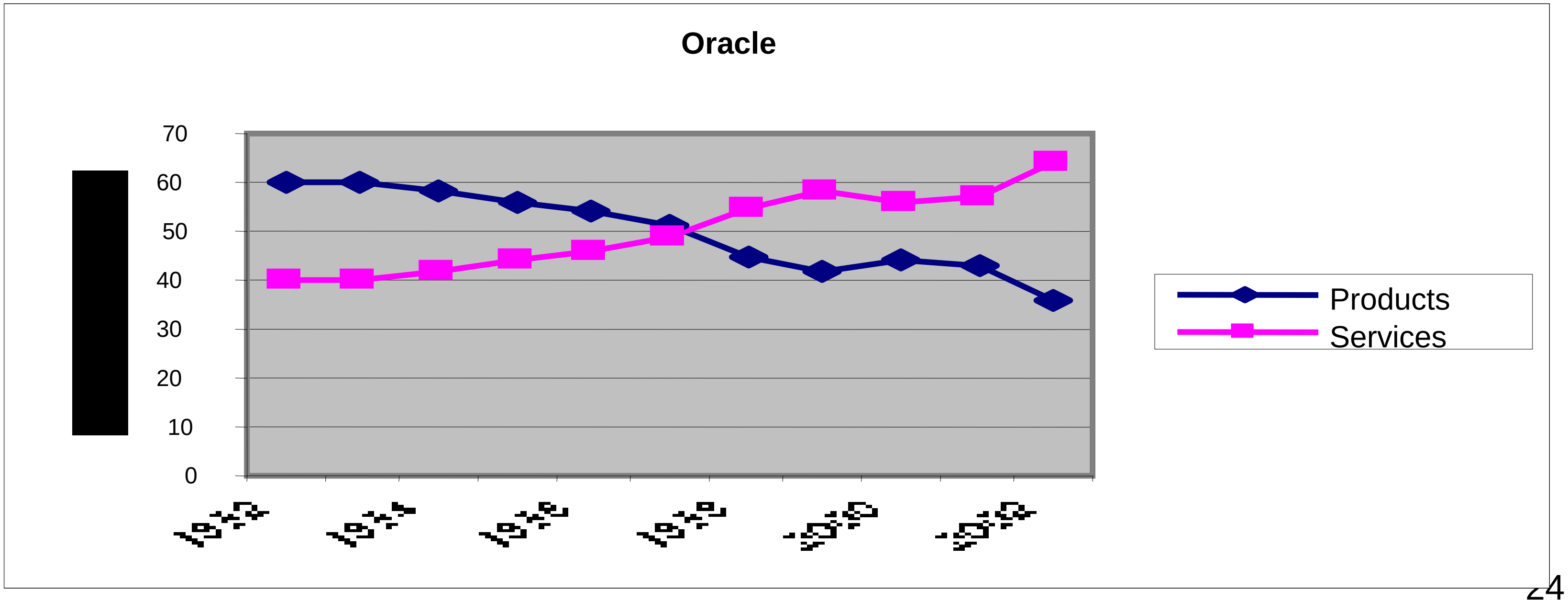

PeopleSoft
1600
1400
1200
1000
800
600
400
200
0
Products
Services


PeopleSoft
200
150
100
50
0
1999
2000
2001
2002
Products
Services
Total

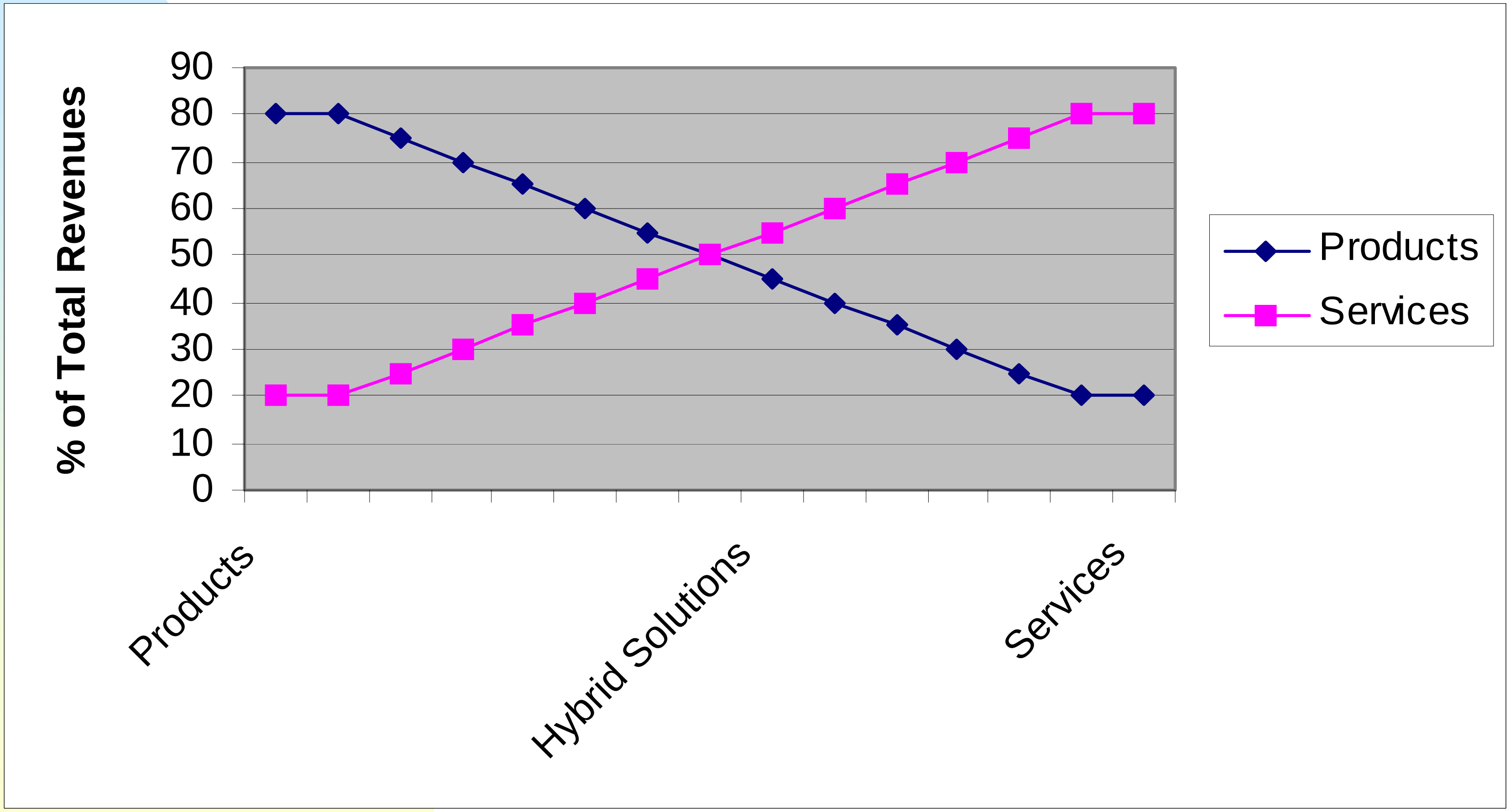

PeopleSoft
80
70
60
50
40
30
20
10
0
Products
Services


Oracle
$ million
7000
6000
5000
4000
3000
2000
1000
0
1992
1994
1996
1998
2000
2002
Products
Services


Oracle
1999 = 100
200
150
100
50
0
1999
2000
2001
2002
Products
Services
Total


Oracle
70
60
50
40
30
20
10
0
Products
Services

Three Business/Life Cycle Models

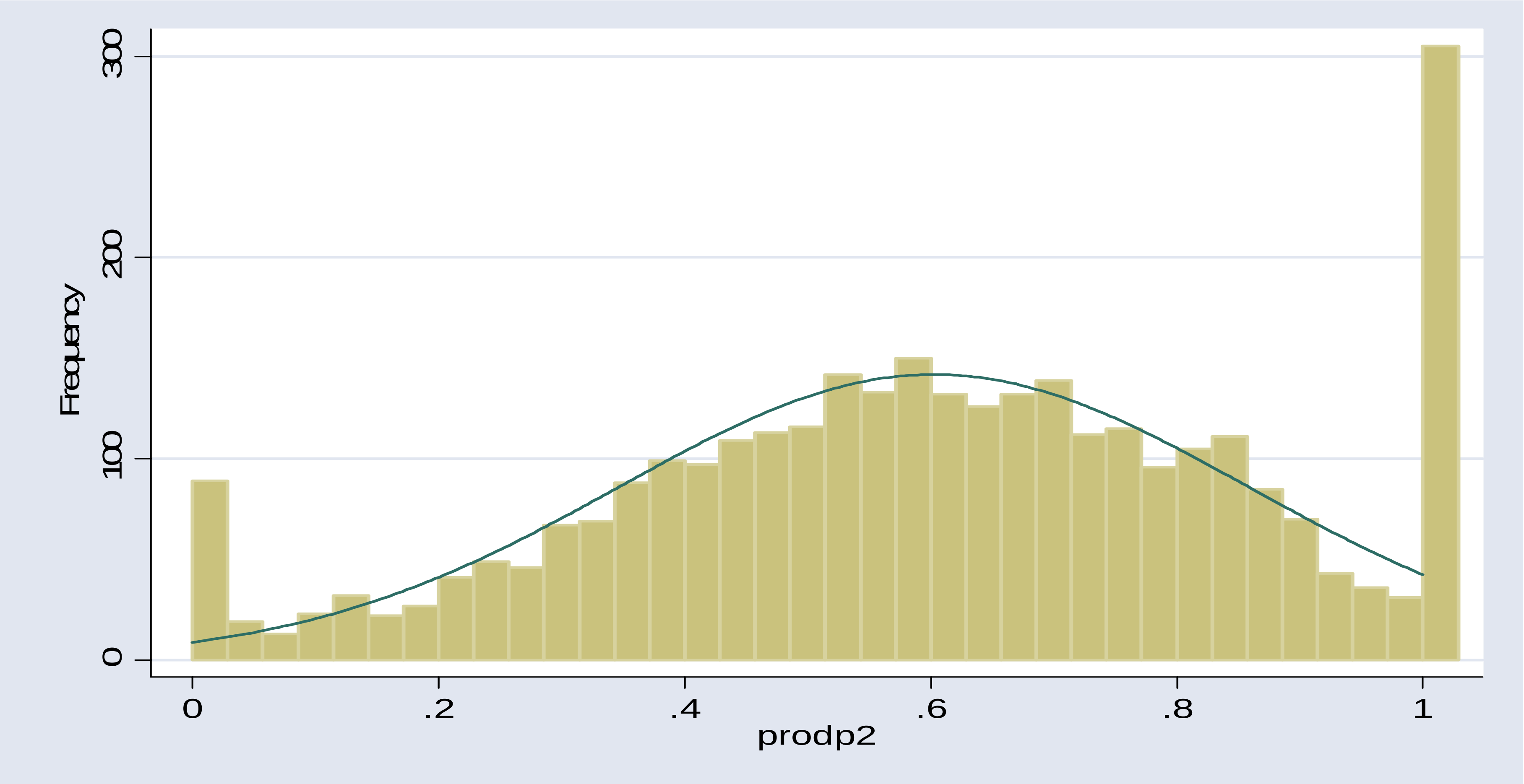


New Database Study

- Continuation of business model research done in The Business of Software book
- Now identified 463 public software "products firms" on US stock exchanges under SIC code 7372 – **PrePackaged Software** (NAICS #51121)
- Avg. 9, maximum 15 years of detailed financial information, from firms listed in 1995 or later.
- 3386 yearly observations (4198 with no-breakout of products vs. services)

Annual % Product Revenues by Firm

(374 Software PRODUCT Firms, 3386 yearly observations)



Notes: -- Excludes 89 packaged software firms with no sales breakout and unclear status.
-- 1 (100%) includes some product firms that did not break out revenue mix (MSFT, Adobe, SPSS, Visio, Symantec, and Fair Issac, and game software firms).

Data Analysis

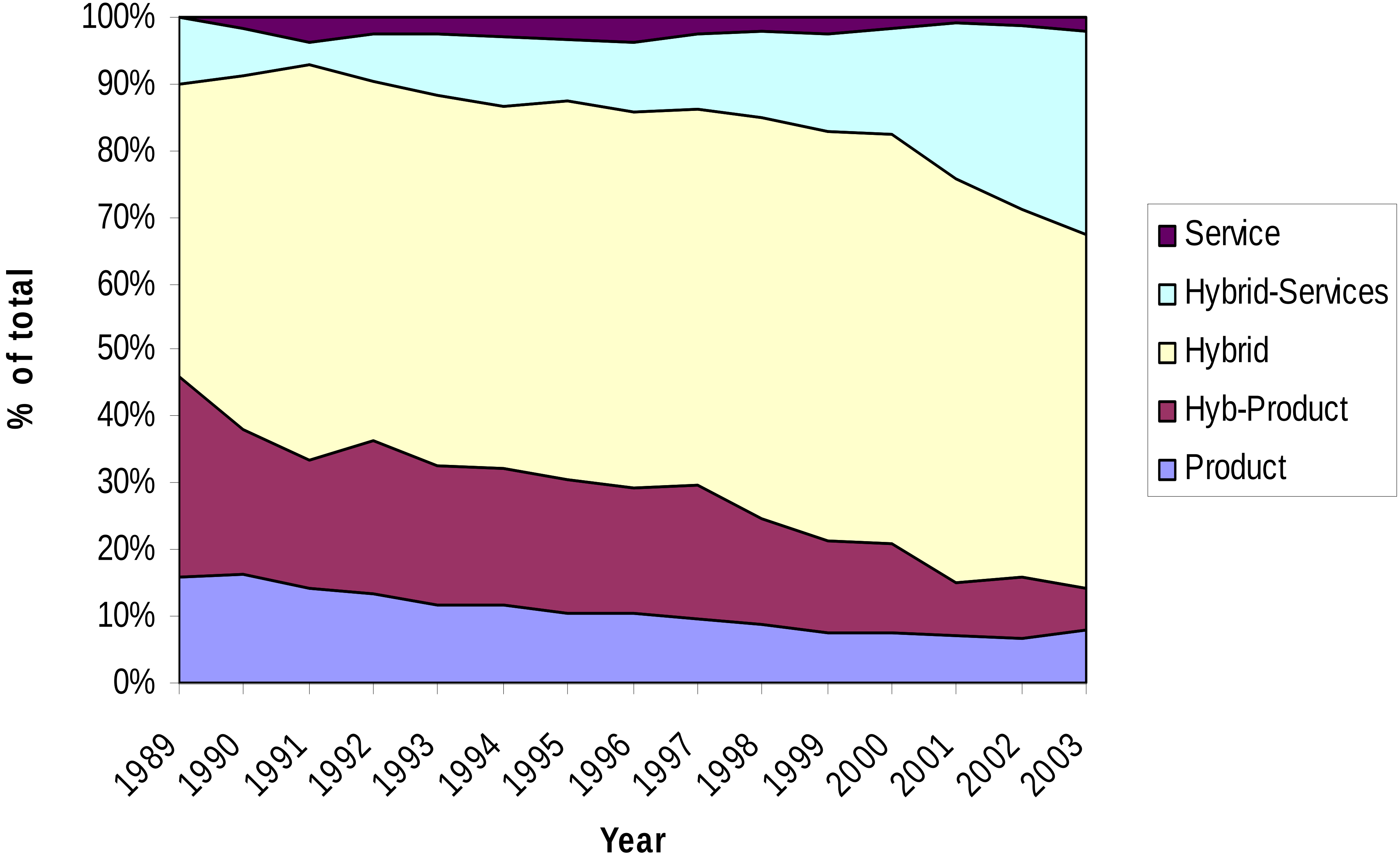
- Broke out “hybrids” using standard deviation. Distribution approximated normal. Used 1 standard deviation to calculate the middle group.
  - **HybridServices = product sales > 0 but < 35%**
  - **HybridBalanced = product sales ≥ 35% but ≤ 80%**
  - **HybridProducts = product sales > 80% but < 100%**

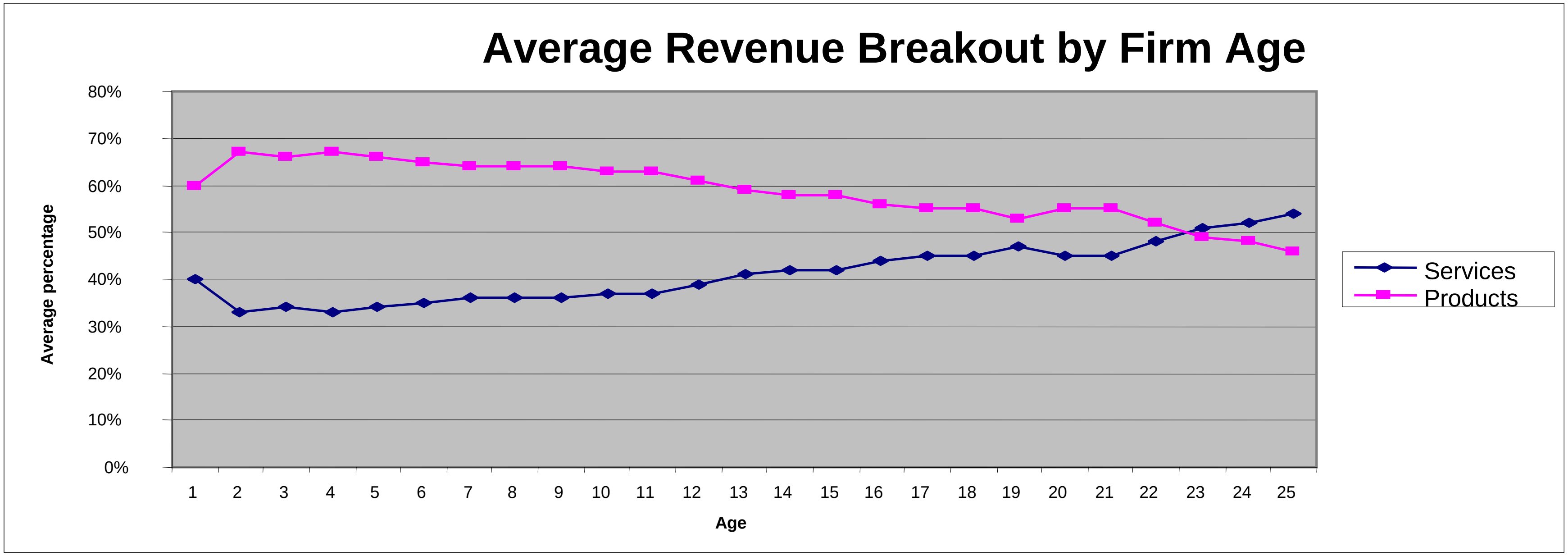
- **Total observations for the 5 groups:**

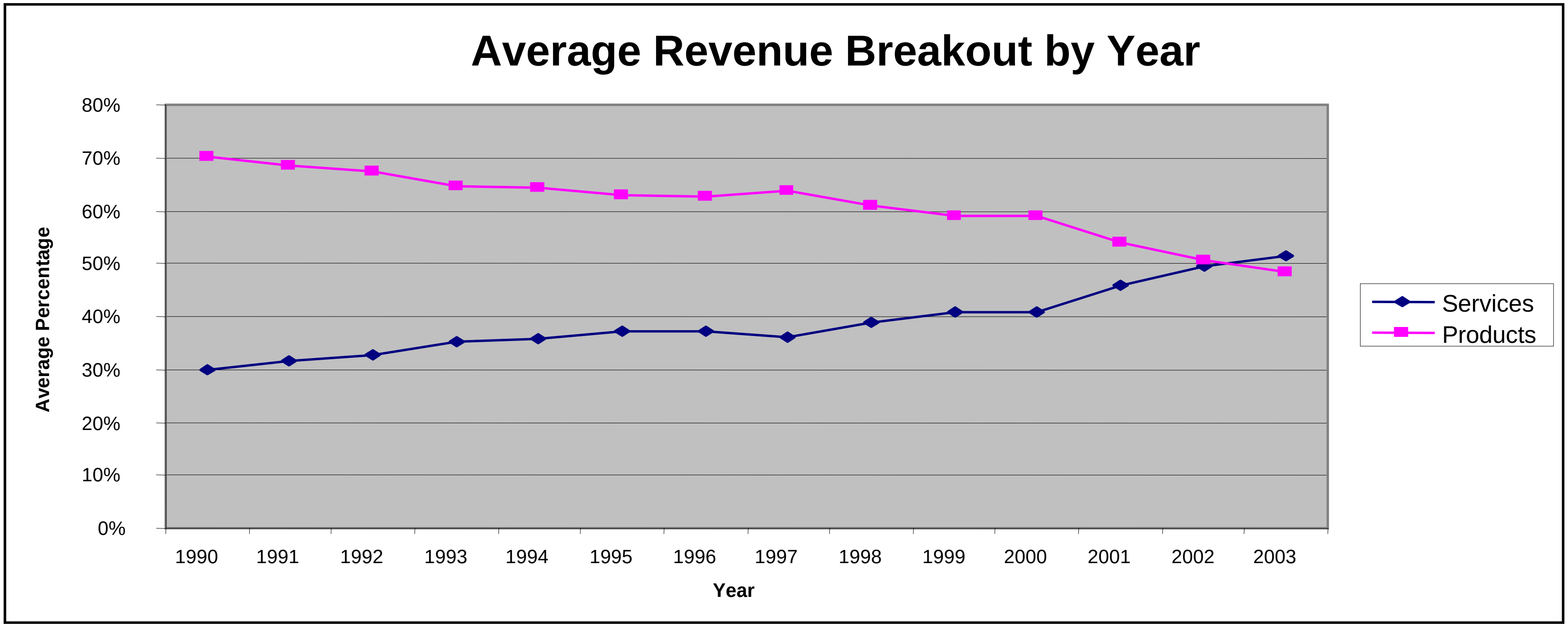
Services:	72
Product:	300
HybridS:	463
HybridB:	1805
HybridP:	504
Total:	3144

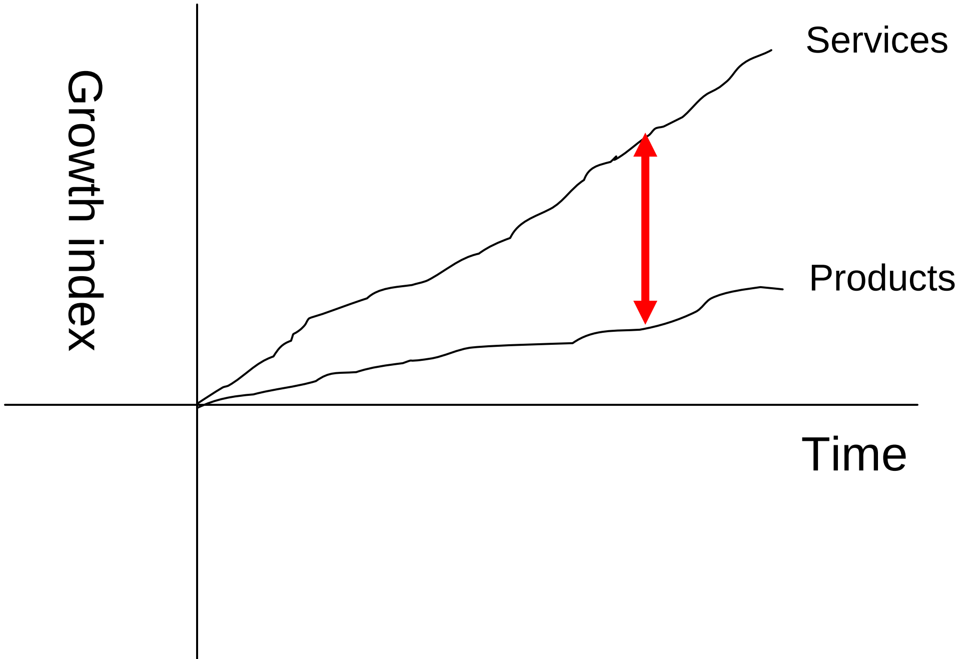
	100%	Hybrid	Hybrid	Hybrid	100%	
Year	Product	Product	Balance	Service	Service	Total
1995	25	48	135	22	8	238
1996	28	52	154	29	**10**	273
1997	**30**	**64**	178	36	8	**316**
1998	27	49	**187**	41	6	310
1999	23	41	186	44	8	302
2000	21	38	172	44	5	280
2001	18	21	157	61	**2**	259
2002	16	21	128	**64**	3	232
2003	**16**	**13**	**106**	62	4	**201**
Total	300	504	1805	463	72	3144

Percentage Breakout by Year

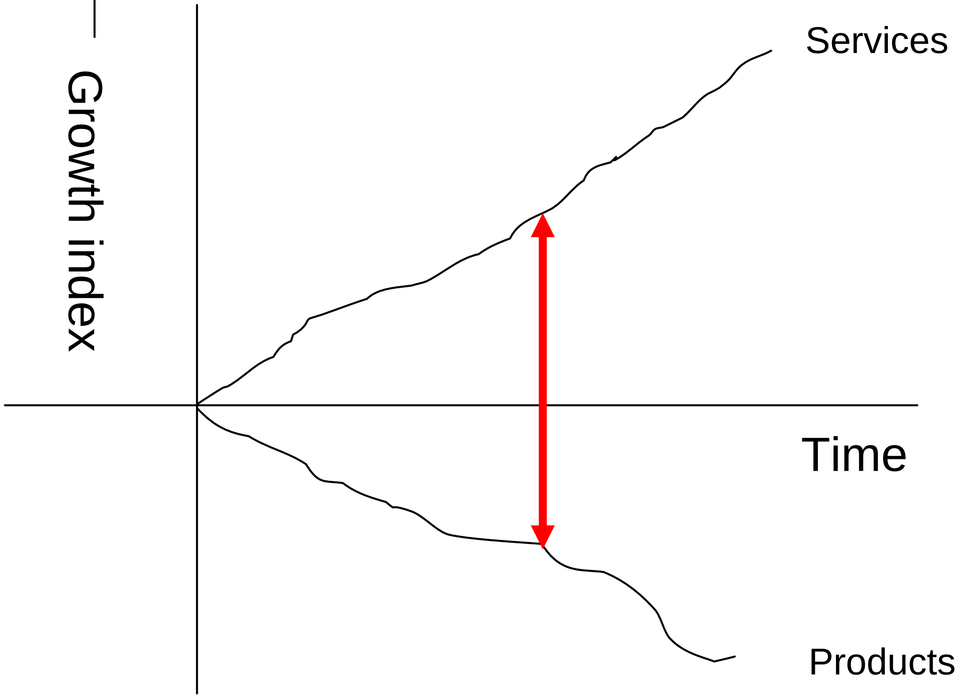



Average Revenue Breakout by Firm Age
Average percentage
80%
70%
60%
50%
40%
30%
20%
10%
0%
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
Age
Services
Products


Average Revenue Breakout by Year
Average Percentage
80%
70%
60%
50%
40%
30%
20%
10%
0%
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
Year
Services
Products



A: Case of a firm where products and services revenues reinforce each other

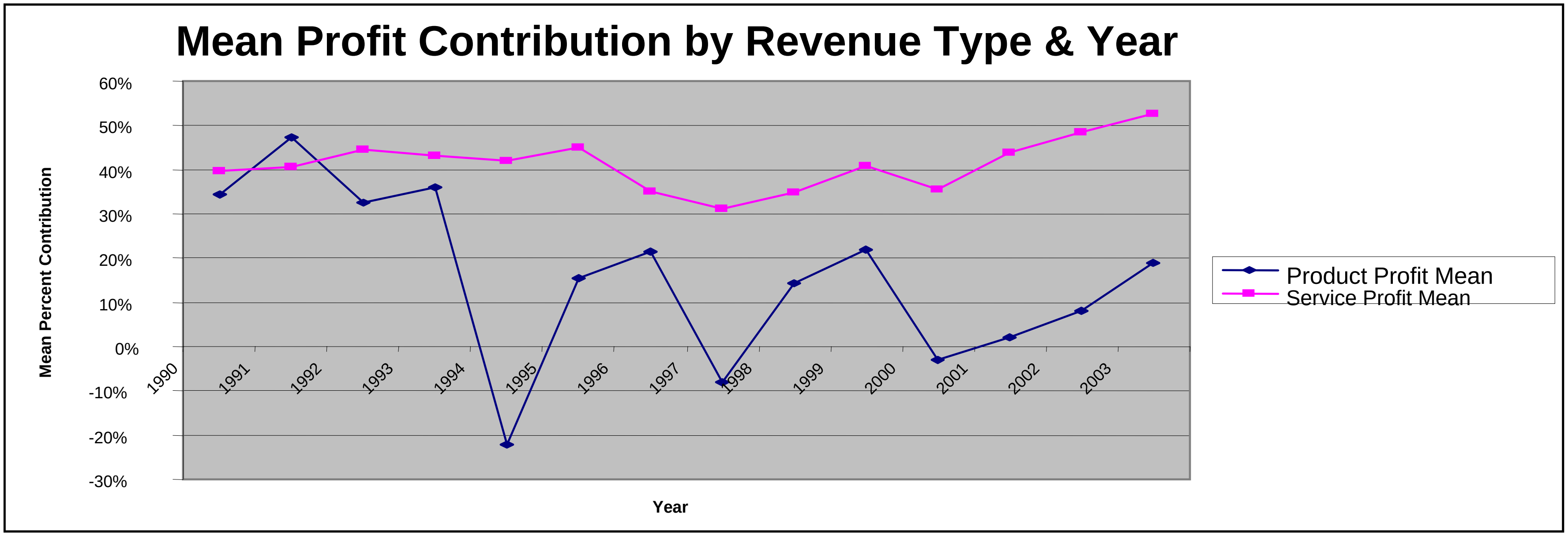


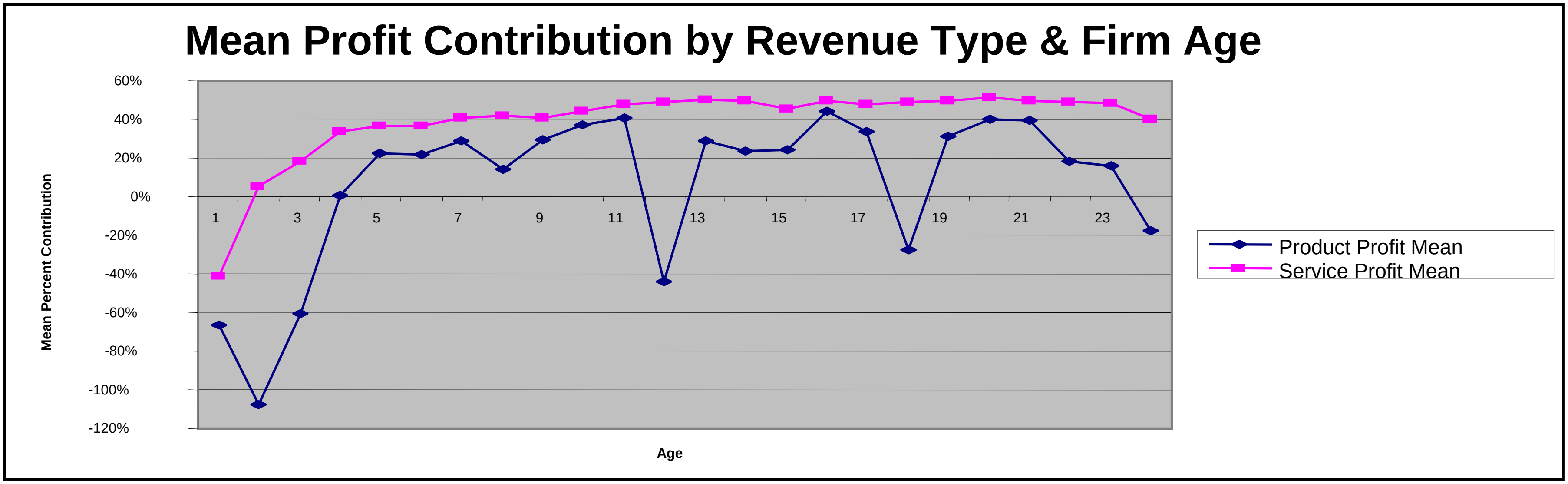
B: Case of a firm where products and services revenues <u>do not</u> reinforce each other

Revenue Mix and Performance

- **Service-maintenance revenues** generate higher and more stable profits than product revenues ***for all software product firms if we include the costs of R&D***

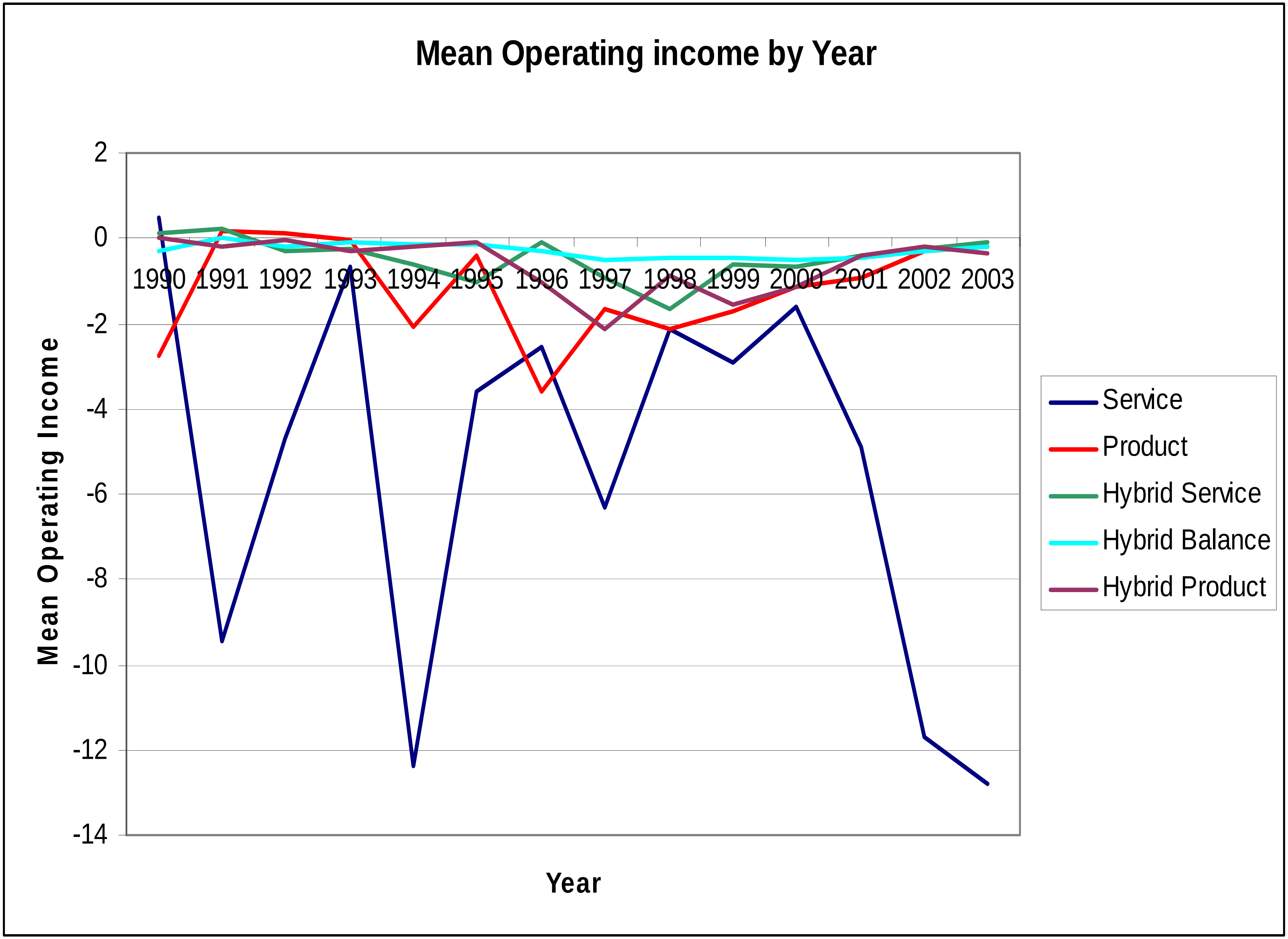
- **Hybrid solutions firms** generally have higher and more stable profits and higher market valuations than software product firms dominated either by product or service revenues ***if we exclude Microsoft***





Product profitability = (product sales – (product cost + R&D)) / product sales

Service profitability = (service & maintenance revenue – service & maintenance cost) / service & maintenance revenue


Mean Operating income by Year
2
0
-2
-4
-6
-8
-10
-12
-14
Mean Operating Income
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
Year
Service
Product
Hybrid Service
Hybrid Balance
Hybrid Product

Maintenance Contribution

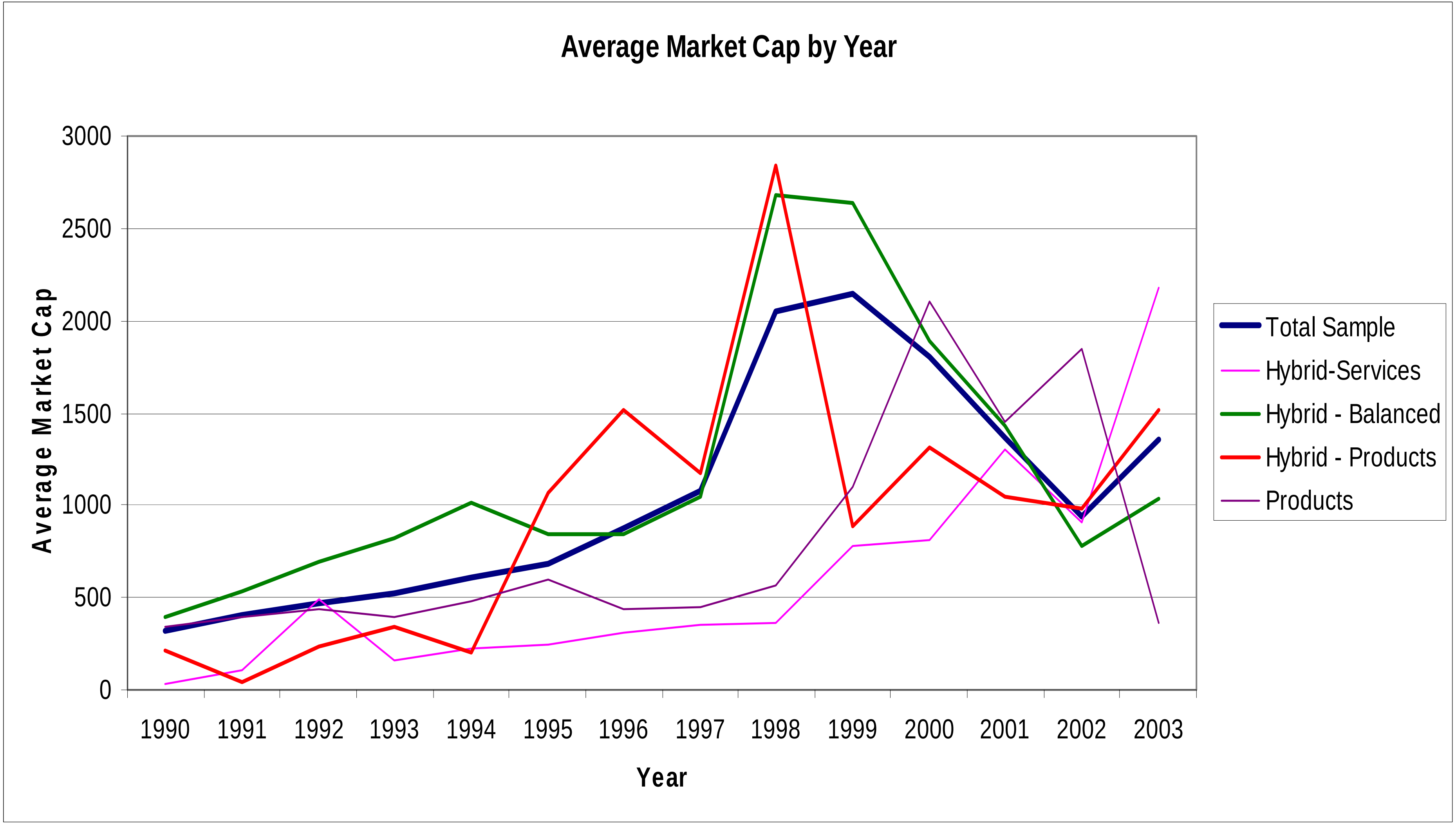
- Sample: 598 data points of firms per year that broke out maintenance from other service revenues

- Avg. 61% maintenance as % of total service revenues

- Adjusted avg. 55% if eliminate 75 data points of firms per year reporting 100% maintenance

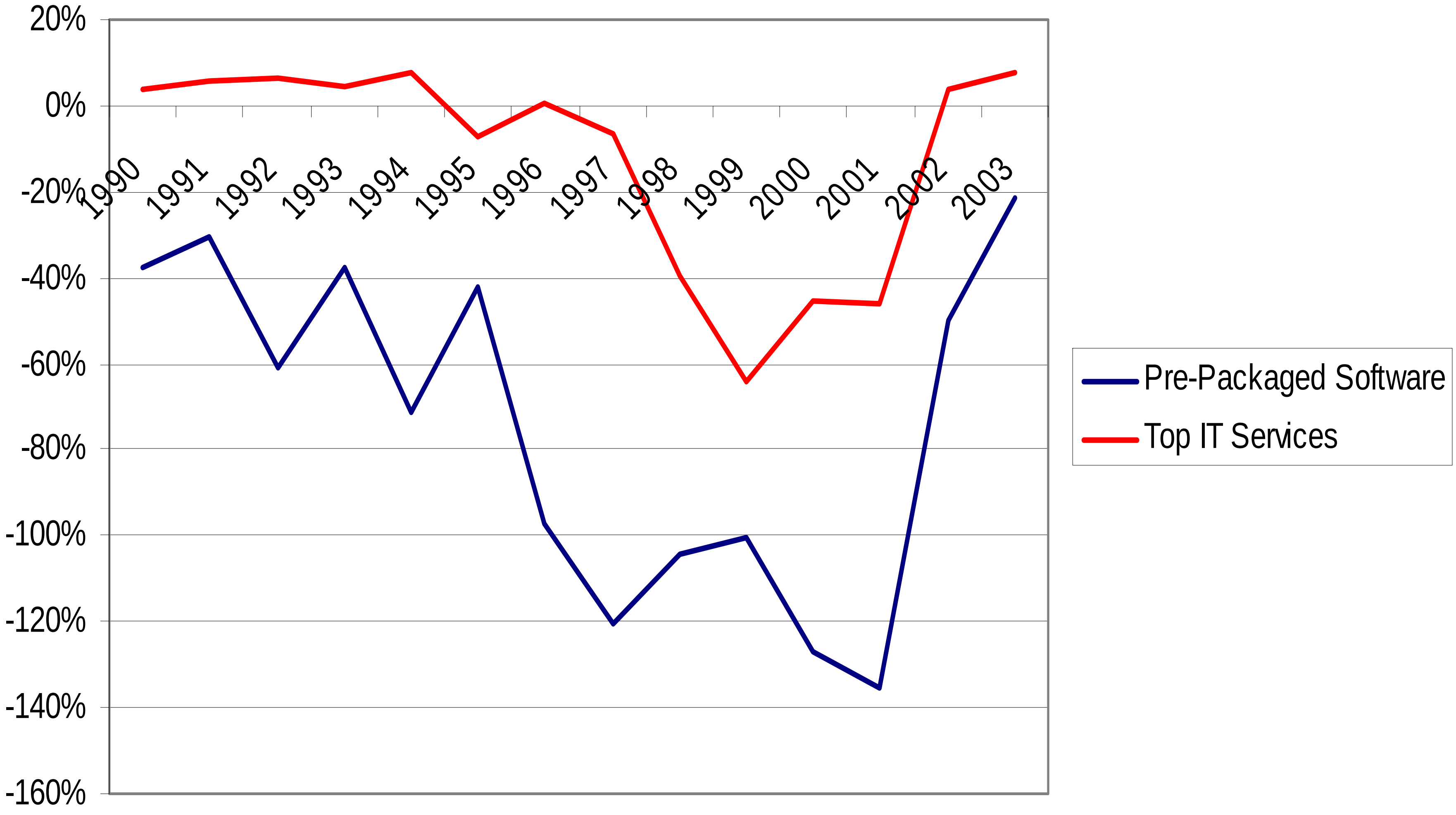
Regression results: 10% increase in maintenance as a % of service = 5.3% increase in service margins!!

Market Value by Business Model



NOTE: PRELIMINARY DATA (not all MKT CAP information entered), excludes Microsoft

Mean Operating Margin by Year



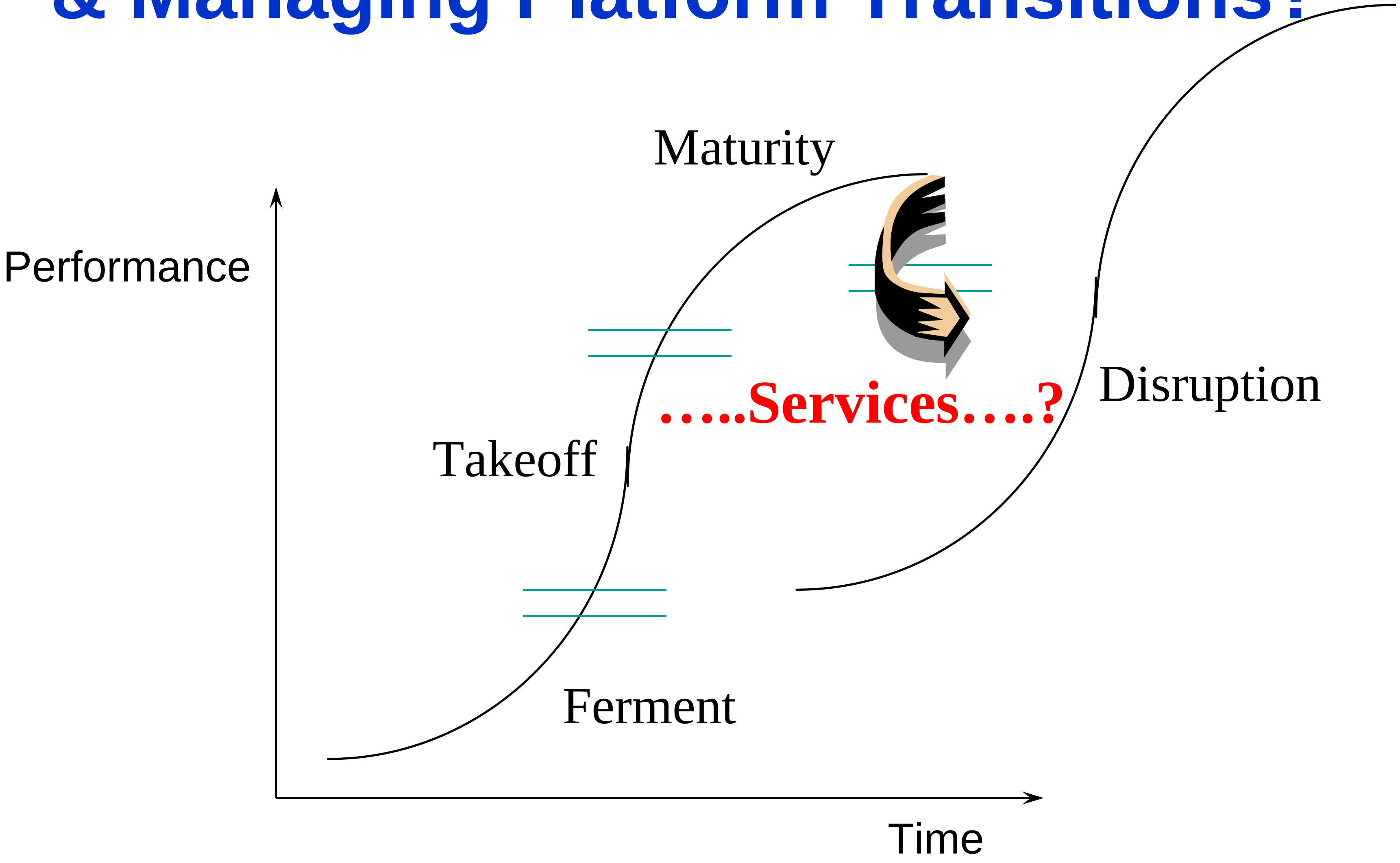
The Empirical Reality

- Most public software products firms ***become*** services or hybrid firms, ***like it or not***
- Service and maintenance revenues can rise dramatically as a % of total sales
  - **in bad economic times**
  - **over the industry/product lifecycle**
- May be a general trend, not limited to software

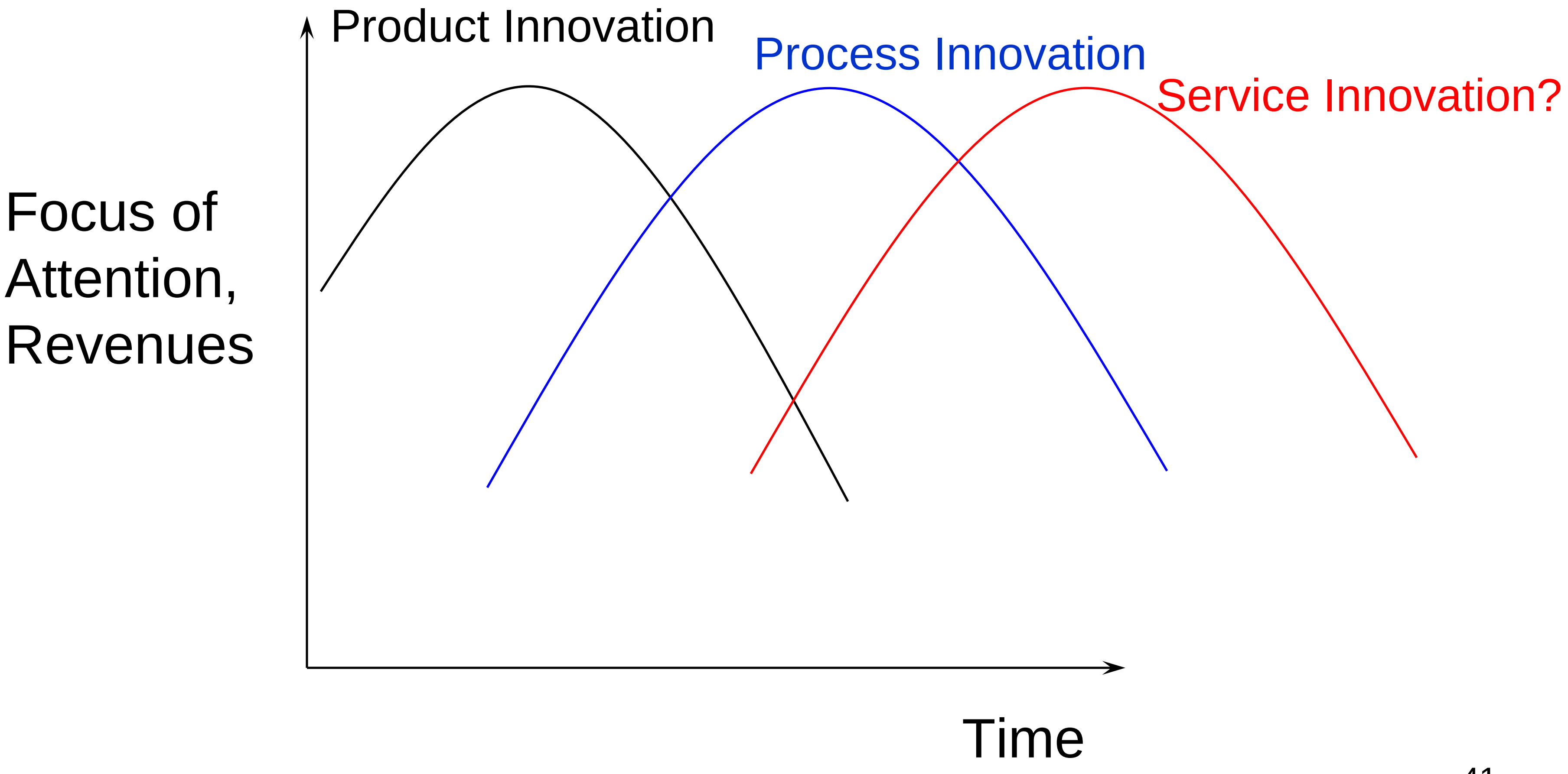
Why not manage the evolution strategically?

➔PLAN to become a hybrid business, from the start of the company

New Insights into Life-Cycle Dynamics & Managing Platform Transitions?



New Insights into a *Different Curve* – Product-Process + ***Services?***


Product Innovation
Process Innovation
Service Innovation?
Focus of Attention, Revenues
Time