Software Business Now and in the Future –

Fundamentals, Trends and Opportunities

Jyrki Kontio, Ph.D.

jyrki.kontio@iki.fi
http://www.jyrkikontio.fi/

jyrki.kontio@rdware.com
http://www.rdware.com/

Presenter Background

- Principal Consultant and Founder of R & D-Ware Oy
- Board member at
  - QPR Software Oyj
  - Webropol Oy
  - Finnish Software Entrepreneurs Association
  - Finnish Information Processing Association
- Professor of Software Business, Helsinki University of Technology, 2002 – 07
  - Adjunct professor of Software Engineering (1997-2000)
- Nokia, 1986 – 2002
  - Knowledge-based systems research and consulting at Research Center (1986-92)
  - Manager of the software engineering research group at Research Center (1992-94)
  - Quality manager at a business unit (1997-99)
  - Senior manager at Nokia Networks: process management (1999-2000)
  - Principal Scientist at Nokia Research Center, software capability (2001-02)
- Other experience
  - Senior researcher at University of Maryland in professor Basili’s research group (1994-96)
  - Software development and management in software houses and corporations (1982-1986)

Outline

- Why is Software Business important?
- What are the critical challenges in Software Business?
- How is Software Business different from other businesses?
- What are current and future trends and what do they mean?

Why is Software Business important?

- Declining work hours
- Aging population
- Price competition in manufacturing industries

- Substantial improvements are needed in all industries
  ➔ Software
- We need to find other growth industries
  ➔ Software

Kasvufoorumi 08

Growth Forum

Jyrki Kontio
Board Member
Software Entrepreneurs' Association
Finnish Information Processing Association FIPA
www.jyrkikontio.fi

Growth Forum

- A national, industry-driven initiative jointly with academia, goverment organizations and small and large software firms
- Objectives:
  - Understand how important software is for the Finnish economy
  - Find out the biggest challenges for growth
  - Propose concrete action to overcome these challenges

New Insights

- Industry has had continuous strong growth
- Over 2% of GNP
  - Substantial addtional impact on the economy
- <u>Willingness to grow</u> is the most important factor influencing the growth of a firm
- IT industry is very international already now
- Internationalization is the most important growth path for practically all growth-driven firms

Challenges

Industry's Internal Challenges

1. Sales and marketing (1)
2. Small company size (2)
3. Poor understanding of the market and the customer (3)
4. Difficulties in defining a growth strategy (4)

National Challenges

1. Lack of entepreneneria culture (5)
2. Small size of Venture Capital market (6)
3. Poor willingness to take risks (7)
4. Poor ability to take risks (8)

Global Challenges

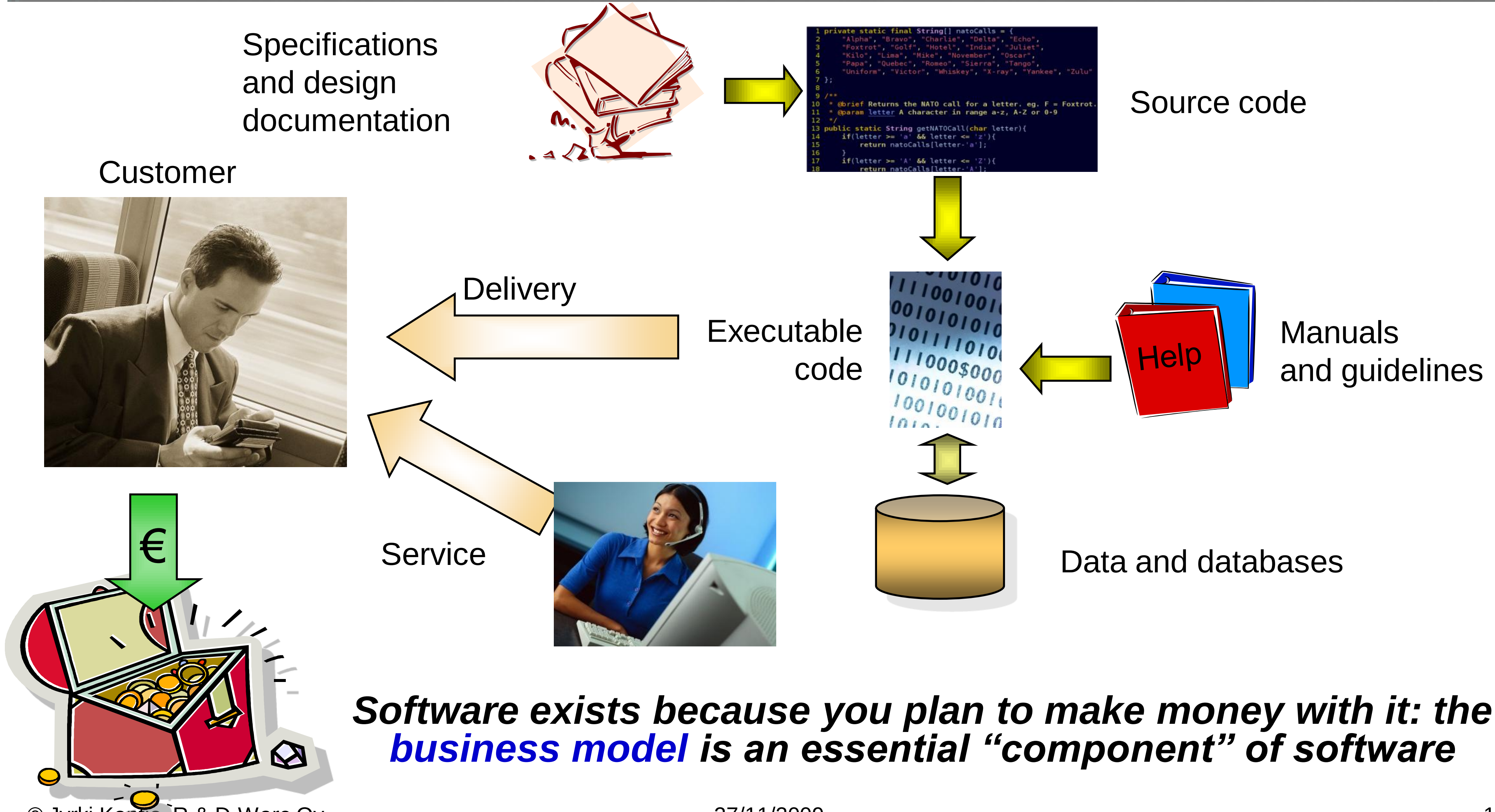
1. Long distance to markets and innovation hubs (10)
2. Global competition in products and solutions (12)

Challenges

Industry internal challenges	National challenges	Global challenges
1. Shortcomings in sales and marketing (1) 2. Small company size (2) 3. Poor knowledge of the market and the customer (3) 4. Difficulties in creating the growth strategy (4) 5. Insufficient partnership networks (9) 6. Leadership challenges during growth (11) 7. Managing the growth (13) 8. Increasing complexity of management due to growth (14) 9. Poor compatence and skills development (16)	1. Cultural anti-entrepreneurial climate (5) 2. Small size of the capital market (6) 3. Low willingness to take risks (7) 4. Poor ability to take risks (8) 5. Difficulties of measuring IT benefits (15) 6. Young people are not interested in the field (18)	1. Long distance from the markets and innovation centers (10) 2. Global competition in products and solutions (12) 3. Competition of other economic areas for workforce and firms (17) 4. Price competition from offshore competitors (19)

WHAT IS SOFTWARE BUSINESS?

What is Software?



```
private static final String[] natoCalls = {
    "Alpha", "Bravo", "Charlie", "Delta", "Echo",
    "Foxtrot", "Golf", "Hotel", "India", "Juliet",
    "Kilo", "Lima", "Mike", "November", "Oscar",
    "Papa", "Quebec", "Romeo", "Sierra", "Tango",
    "Uniform", "Victor", "Whiskey", "X-ray", "Yankee", "Zulu"
};

/**
 * @brief Returns the NATO call for a letter. eg. F = Foxtrot.
 * @param letter A character in range a-z, A-Z or 0-9
 */
public static String getNATOCall(char letter){
    if(letter >= 'a' && letter <= 'z'){
        return natoCalls[letter-'a'];
    }
    if(letter >= 'A' && letter <= 'Z'){
        return natoCalls[letter-'A'];
```

Software exists because you plan to make money with it: the business model is an essential "component" of software

Volendam Manifesto on Software Business – Expert Panel

Professor **Sjaak Brinkkemper**, Utrecht University

- ◆ Professor of product software, meeting organizer

Professor **Anthony Finkelstein**, University College London:

- ◆ A leading software engineering scholar in Europe, ICSE-04 general chair

Professor **Alan Hevner**, National Science Foundation and U of South Florida:

- ◆ Program coordinator at NSF, published a highly cited paper " Design Science Research in Information Systems" in MISQ

Professor **Jyrki Kontio**, Helsinki University of Technology

- ◆ Professor of Software Product Business, Head of the Software Business Laboratory

Professor **Alan MacCormack**, Harvard Business School:

- ◆ A leading U.S. scholar in software business and software architecting from business perspective

Professor **Tony Wasserman**, Carnegie Mellon West University:

- ◆ Experienced software entrepreneur, founder and president of IDE (Software through Pictures, reached Top 500 Inc. list)

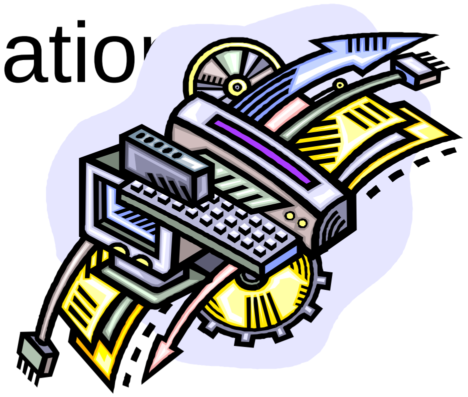
Definitions

- **Software Ecosystem** is a specific perspective to a subset of an economy where software constitutes a substantial part of the transactions between agents on the marketplace.
- **Software Business** refers to transactions between agents, trading software-based assets.
  - *Agents* are organizations or individuals;
  - *Assets* include financial assets, buildings, other resources, capabilities, and, of course, software
  - *Transactions* include monetary transactions, transfer of goods, provision of services, and, of course, delivery and deployment of ***software***
- **Software Business Research Field** studies how these assets are created to create value, how agents interact and what kind of transactions take place, and how technologies are used to support this business.
- We study issues related to bringing software assets to the market. Specifically, we study how these assets are developed to create value, how agents interact, and what kind of transactions take place, and how technologies are used to support this business.

Characteristics of Software

- Complexity
- Conformity
- Changeability
- Invisibility

- Configurability
- Digital good
- Human development process
- Technological change rate
- Extreme integration



Brooks: No Silver Bullet

Complexity

- ◆Each unit is unique and interacts with many other units
- ◆Abstraction (easily) hides complexity
- ◆Yet complexity is an essential feature of software

Conformity

- ◆There are no natural laws governing software
- ◆Human judgment has determined many of the “constants” in software
  - –many of them are not compatible
- ◆Software must conform to other disciplines

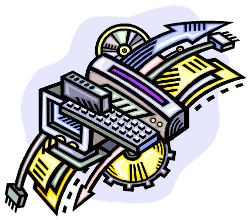
Changeability

- ◆Software is tied to other “systems” that evolve:
  - –user needs
  - –organization
  - –laws, society
  - –hardware platforms
  - –other software systems
- ◆Change is a given environmental factor

Invisibility

- ◆Can’t touch, feel, or smell
- ◆Models (abstractions) are always partial and multidimensional
- ◆It is difficult to understand what software is

Other Technical Characteristics of Software



Configurability

- Software can be configured near or at runtime into different configurations
- Mass customization at point of sale possible

Digital good

- Software is entirely digital
- Can be copied error-free
- Can be digitally transmitted

Human development process

- Software engineering is people dependent:
  - Human creativity
  - Subjectivity of specification and design
  - Dependence on teamwork

Technological change rate

- No other technology evolves so rapidly
  - ”Programming languages change in less than 10 years”
  - ”Application platforms change every 5 years”
  - ”Development paradigms change every 10 years”
  - ”Development environments are updated annually”

Extreme Integration

- Software is **technically** linked to many other systems
- Software is closely tied to **processes** of an organization
- Software creates many **social** interactions and dependencies

From Technical to Business Characteristics

Characteristics of Software

- Complexity
- Conformity
- Changeability
- Invisibility
- Configurability
- Digital good
- Human development process
- Technological change rate
- Extreme integration

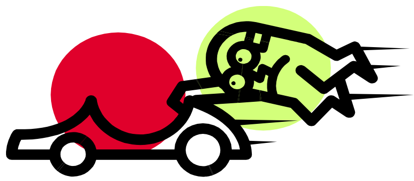
→

Implications to Business

- Easily deployed
- Economies of scale
- Economies of scope
- Instant scalability
- Difficulty of engineering
- Experience good
- Lock-in
- Network effects

Software is Easily Deployed

- Software can be taken to use easily, regardless of time and place
  - Downloaded on demand
  - Automatic installers
  - No storage costs
  - Instant and low cost access in many situations
- Advances in usability have made instant use common
- Network access required
- Examples:
  - Software upgrades, e.g., anti-virus software
  - Mobile phone game downloads
  - Time-based capacity in a network



Software has Economies of Scale

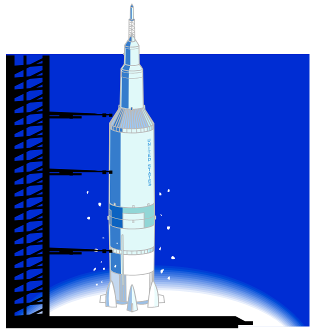
- Definition:
  - Economies of scale exist when the cost of production/manufacturing decreases when large numbers of the good are produced
- Once developed, software has tremendous economies of scale
- Integration and training costs reduce immediate economies of scale
  - Standardizing (”productizing”) services and integration will improve economies of scale
- Examples:
  - Microsoft products, any packaged, high volume software

Software has Economies of Scope

- Definition:
  - “Fewer inputs, such as effort and time, are needed to produce a greater variety of outputs.
  - Greater business value is achieved by jointly producing different outputs. Producing each output independently fails to leverage commonalities that affect costs.” (www.sei.cmu.edu/productlines/glossary.html)
- Good architecting and business planning increase potential for economies of scope
- Examples:
  - IT companies offering products for the telecom sector
  - Google offering its generic search engine also for company intranets

Software has Instant Business Scalability

- Software has nearly instant business scalability: possibility to deliver large quantities with relatively small change in the use of resources
- In practice, scalability is limited through:
  - Human sales efforts
  - Logistics (traditional delivery)
  - Invoicing
  - Support requests
- Examples:
  - Internet allows easy and fast distribution
  - Netscape invested in high-volume business applications early
  - Amazon.com built up scalability early



Difficulty (and cost) of Engineering

- Development of the first version of software is costly and time-consuming
- Business case for software is largely dependant on the volume of use
- It is really, really hard to estimate and deliver cost, schedule and functionality accurately
- Examples:
  - Software customization
  - New release development

Software is an Experience Good

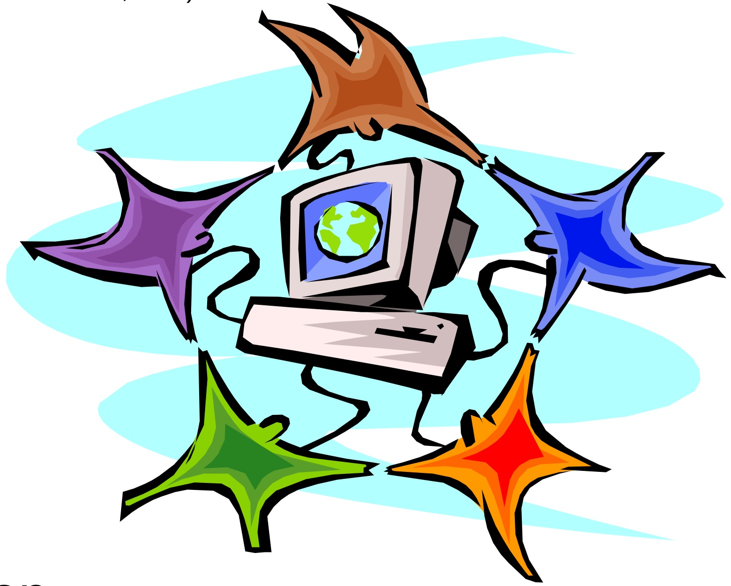
- **Experience good**: product characteristics are difficult to observe in advance, but these characteristics can be ascertained upon consumption or use
- Any description or metric about software is bound to be an abstract summary of software
- Examples:
  - SAP
  - Network management software
  - iTunes software

Software has Strong Lock-in Effects

- Definition:
  - Initial sales or ownership of products create an investment for the customer that makes him reluctant to change products or vendors
- Software lock-in occurs, e.g., through
  - Customization of software
  - Integration of software with other systems
  - Learning curve costs
  - Customer specific knowledge in services
- Examples:
  - Mobile phone
  - SAP
  - Microsoft Windows operating system

Software has Strong Network Externalities

- Definition:
  - “effects on a user of a product or service of others using the same or compatible products or services” (http://www.answers.com/)
  - “a cost or benefit that arises from production and falls on someone else than the producer” (Bowman & Ambrosini, 2000)
- Examples
  - Positive network externalities:
    - Consulting service available
    - Fax machine, MS-Office
    - GSM standard
    - Facebook
  - Negative network externalities:
    - Other users on a shared Internet connection

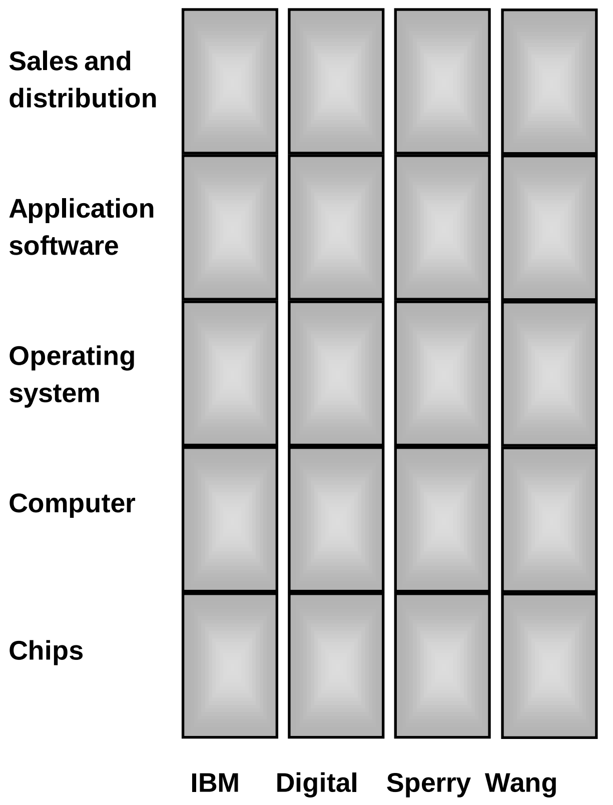


Trends affecting Software Business

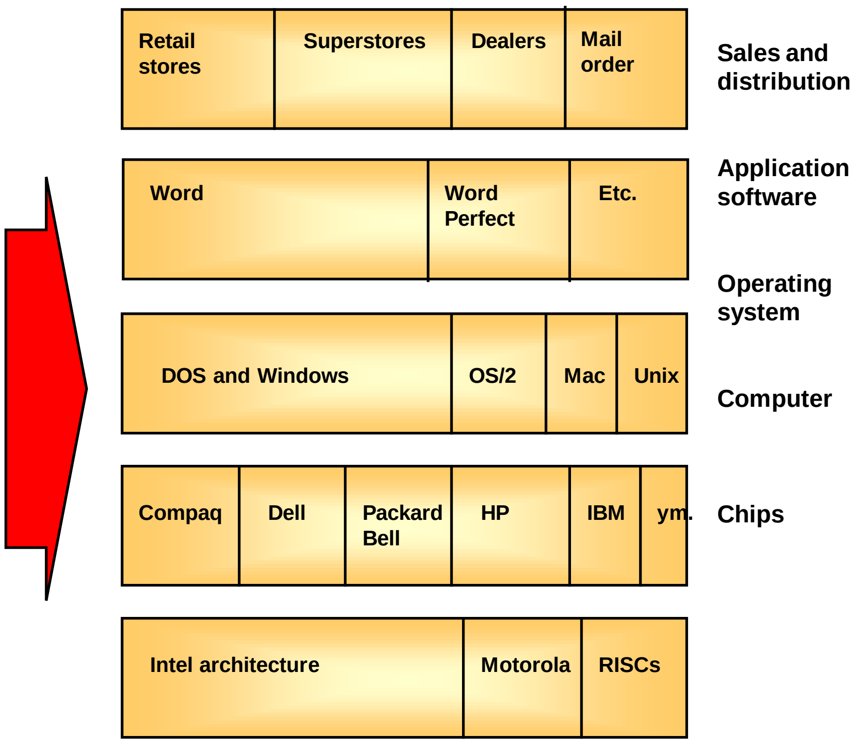
- **Softwarization** – “ohjelmistoituminen”
  - Services and functionality in traditional products are offered through software
- **Servicization** – “palvelullistuminen”
  - Software functionality is increasingly offered as a “service”
- **Hybridization** – ”hybridisoituminen”
  - Functionality, services and content together create value to users
- **Productization** – ”tuotteistuminen”
  - Both software and services are being ”productized” for efficiency and profitability
- **Componentization** – ”komponentoituminen”
  - Components are increasingly used as building blocks of systems
- **Communization** – ”yhteisöllistyminen”
  - User communities are involved in the development of software
- **Commoditization** – “tavallistuminen”
  - Hardware and basic software functionality are becoming standard and available (cheap or free)
- **Internetization** – ”internetistyminen”
  - Internet is the platform – and it’s mobile

Example of an Industry Change

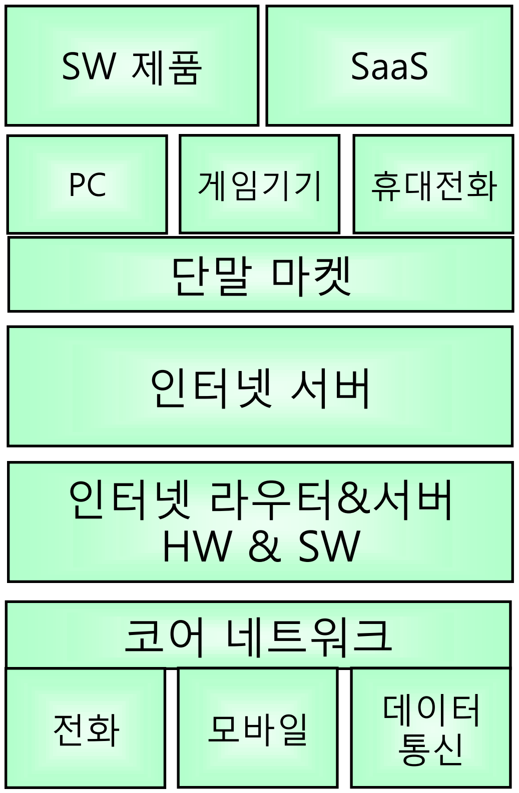
Vertical Software Industry Structure -- ca. 1980's



Horizontal industry structure ca. 1995

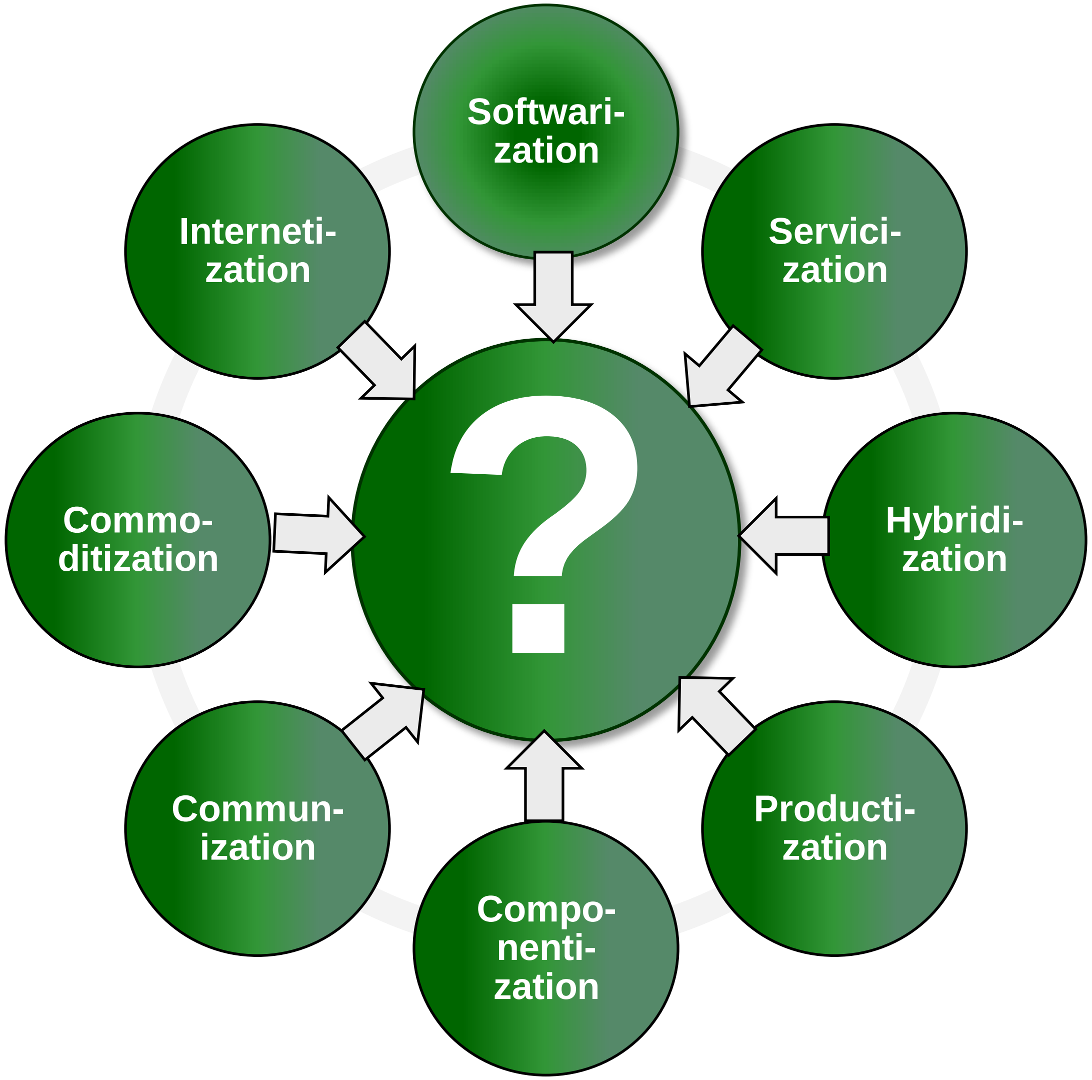


The Converging Communications Network Industry Structure… TBD



Andy Grove, 1996

Trends have Compound Effects



Implications of Trends

Softwarization	• Expanding markets and opportunities • New domains
Servicization	• Easier buying decision, easier use, harder dev't • Lower prices, more customers
Hybridization	• Synergies between products, services and contents
Productization	• Ability to identify the common needs of many • More competition, more potential
Componentization	• Your software is build from existing components • Your software will be a component
Communization	• Customers are closer and have more power • Network effects have major potential
Commoditization	• Your software will become obsolete – unless you differentiate
Internetization	• Accesssibility and distribution revolution • Online depedence

Conclusions

- Software drives the economy
- Most critical issues determining the success of a software firm are non-technical
- Software business has unique chracteristics – master them
- A substantial change is taking place – risks and opportunities are plentiful

Thank you

QUESTIONS?

Further info:

- http://www.rdware.com

Download a copy of these slides:

- http://www.jyrkikontio.fi/material.php