

# BUSINESS ASPECTS OF SOFTWARE ENGINEERING

DR. VIVEK NALLUR

VIVEK.NALLUR@SCSS.TCD.IE

# OUTLINE OF THIS TALK

- Business Aspects
- Legal Aspects

# STRATEGIC DECISIONS

Big software projects are strategically important for organizations

Incumbent upon senior management to understand the costs, options, risks, and strategic implications of software projects

*Required:* Someone who is familiar with both, computing aspects of the project and the strategic direction of the company

# CHIEF INFORMATION OFFICER

- Very senior in the organizational hierarchy
- Usually, the equivalent of the *vice-president* in decision-making authority
- Is familiar with both, technology as well as business requirements
- Sometimes also known as *Chief Technology Officer*

# PRODUCT MANAGER

- Responsible for viewing the software project purely as a *Product*
  - *Function*: What does it do?
  - *Market*: Who is it for?
  - *Sales*: What price, advertising, sales channels are available?
  - *Legal*: What licenses, legal protections shall apply to the product?

# IN-HOUSE COMPUTING DEPARTMENT

- Organization
  - Central computing department reporting through a CIO
  - Decentralized model with computing distributed across operational units
- Expertise
  - Full in-house software design and implementation
  - Outsourced with project-management in-house
  - Complete outsourcing

# SOFTWARE HOUSE

*A software house* is a company that develops customized software for other organizations. E.g., Computer Associates, Infosys, etc.

- Consulting
- Packages (solutions)
- Custom software
- Maintenance
- Education

Emphasis usually on highly visible processes, acceptance tests, and well-defined contracts

# BESPOKE SOFTWARE

- Highly Specialized Environment
  - Large applications (e.g., air traffic control)
  - Highly-specific software (e.g., high performance computing labs)
- Development Options
  - In-house development
  - Contractor-based development
- Maintenance Options
  - In-house
  - Contractor-based



# PACKAGES WITH MODIFICATIONS

- General purpose software written by software House
  - Modified for client's needs
  - Client licenses code for self-modification
- Business Considerations
  - Modifications in-house or by software house
  - Maintenance in-house or by software house

## PACKAGES WITH MODIFICATIONS - II

- Legal Issues
  - Access to source code
  - Ownership of modifications
  - Lock-in of vendor

*Examples:* Corporate pay-roll systems, accounting systems for small business, etc.

# PACKAGES WITHOUT MODIFICATIONS

- Software licensed in binary form only
  - Typically has many configuration options
  - Possibly standalone or designed to be included in other applications
- Legal Considerations
  - Clear distinction between license and binary package
  - *Examples:* Database systems, virtualization software, mathematical and scientific computation packages

# EMBEDDED SYSTEMS

- Software is bundled with hardware
  - Original form of computer software development
  - Product is seen as hardware product, even if software consumes major fraction of cost
  - *Examples: Cars, HVACs, etc.*

# OUTSOURCING

---

<i>Concept</i>	Contract with software house to develop software for an organization
----------------	--

---

<i>Benefits</i>	Software house better organized for development. No need to build an in-house team. Complex projects can be designed and executed
-----------------	---

---

<i>Disadvantages</i>	Organizational goals may clash
----------------------	--------------------------------

---

Considerable project-management expertise needed

# FREE-LANCE DEVELOPMENT

You and your friends decide to form a small company and write bespoke software

How much should you charge?

You plan to work 40 hours a week, for 50 weeks a year and plan to earn 100,000 euros.

$$\text{Hourly rate} = 100000 / (40 * 50) = 50/\text{hr}$$

## LET'S DO SOME ELEMENTARY BUSINESS MATH

<i>Salary</i>	100000
<i>Benefits</i>	30000
<i>Rent, equipment, etc</i>	20000
<i>Fees, services, etc</i>	30000
<i>Travel, misc. expenditure</i>	10000
<i>Total Expenditure</i>	190000

## FREE-LANCE DEVELOPMENT - II

Hours worked	2000
<i>less admin</i>	400
<i>less marketing</i>	200
Billable hours	1400

Hourly Rate =  $190000 / 1400 = 135.7/\text{hr}$



## FREE-LANCE DEVELOPMENT - III

- You *must* have a contract with the customer, containing:
  - Is payment based on hours-spent? Upon feature completion? Upon project completion?
  - What are the acceptance tests?
  - Who owns the final software? What license does the customer have?
  - Who provides the hardware?
  - *When* does payment take place?

## PACKAGED SOFTWARE (COSTS)

- Initial development cost: 1 million euros
- Cost of packaging, distribution per copy: 5 euros
- Cost of maintenance, per copy: 15 euros
- Selling price per copy: 75 euros

You sold 20,000 copies. Did you make a profit?

# PACKAGED SOFTWARE - PROFIT/LOSS

- Profit of 10,000 euros?
- Not necessarily.
  - Are you in debt, from startup costs? What's the interest rate?
  - What's the corporate tax rate?
  - When are sales realized?
  - When are bills due?

## UNORTHODOX BUSINESS MODELS: SHAREWARE

- Software is developed and then distributed for free
- Developer requests a fee, if you find the software useful
- Developer requests a fee, if you need help running the software

# UNORTHODOX BUSINESS MODELS: OPEN-SOURCE SOFTWARE

- Software is developed and the source is available freely
- Individual developers hired by company to maintain/enhance it
- Software foundations that function entirely on donated time-and-money

# OPEN-SOURCE LICENSES

- GNU AGPL
- MIT License
- BSD License
- Apache License
- Perl Artistic License
- ...

More (and clear) information available at:

<http://choosealicense.com/licenses/>

Even if software is free,  
packaging/distribution/customizations/other-services can  
be a fairly good business. E.g., *RedHat*, *Canonical*, etc.

# OBLIGATIONS

- Ethical Obligation
  - If a software house creating bespoke software, inform clients about possible effects of licenses of software you use.
  - E.g., GPL v3 is aggressively viral. BSD is not
- Legal Obligation
  - Packaged Software (either as binary or incl. src code): Full disclosure about licenses of sub-components used

## **LECTURE ANNOUNCEMENT**

Lecture on 7th December cancelled! (SFI review)



**THAT'S ALL, FOLKS!**

---

Questions? Comments?