Entrepreneurship and Open Source software

Ir. Robert VISEUR
Assistant, Faculté Polytechnique de Mons
robert.viseur@fpms.ac.be

Journée du Libre 2006
CRP Centre Henri Tudor

Luxembourg, 24 octobre 2006

ACADÉMIE UNIVERSITAIRE WALLONIE-BRUXELLES

Some definitions

- **Software** = set of programs, making a computer or a set of computers able to achieve a task.
- Free software = **software** such as each people having a copy can use it, study it, modify it and redistribute it.
- Free software ≈ Open Source software.
- Free software ≠ freeware / shareware / public domain.

Open Source and co-creation

- Open Source software as a sample of **co-creation**.
- Co-creation : consumer (user) becomes a productor of value / user as co-creator.
- Co-creator can help :
  - Promote the software
  - Distribute the software
  - Write source code
  - Write documentation
  - ...

Three specificities of Open Source software (1)

- Users are gathered in **communities**.

  Power is shared (balanced)

- Development model is based on **cooperation**.
  - Customer as user and developer

    Customer gives his workforce (co-creation).
  - Company can hire talented developers
  - Open Source good for B2C and B2B collaboration

    Reassuring environment for collaboration.

Three specificities of Open Source software (2)

- Specific **licenses**
  - Strong copyleft licenses (as GPL)
    - If a piece of source code is copyleft, all the source code becomes copyleft
    - Contributors love GPL (≈ 65% of free softwares) !
  - Soft copyleft licenses (as LGPL, MPL or CPL)
    - Copyleft is limited to piece of source code (not « viral »).
  - Permissive licenses (as BSD)
    - Source code under permissive license can be licensed under other licence (as proprietary license).

Open or not open ?
Make a calculation ! (1)

- Costs reduction

  More work for less money

  Source code, debug, feedback, promotion, distribution and so on, by contributors.

- Marketing gains

  Developments driven by market, word-of-mouth marketing,...

  → decreased risk,...

  → stronger brand, increased visibility,...

Open or not open ?
Make a calculation ! (2)

- New costs
  - Better software architecture for efficious collaborative work.
  - Community management.
- Revenues lost
  - Traditionnal *commercial software model* doesn't work well.

What is your balance sheet ?

A bit of marketing

- Five competitive forces by Porter :
  - Customers
  - Suppliers
  - Competitors
  - Substitutes
  - Barriers
- Who is the customer ?

  People who has been in contact with the company (buying, subscription to a waranty, use of a service as heldesk, inscription in a database, etc).

Main features : proprietary software

- Customers - Customer is often a prescriptor. Different of the final user.
- Suppliers - Software often based on tools edited by third party companies.
- Competitors - Competitor is often another commercial company in the same sector.
- Substitutes - Not many substitutes. Software is often a substitute (hardware equipment replaced by software).
- Barriers - Intellectual property (licenses, copyright, software patents, etc).

Main features : Open Source software (1)

- Customers – Customer can become supplier or competitor !

  Developers of Mambo angry with Miro. So creation of Joomla.

- Suppliers – Open Source editors can be suppliers each others and become competitors.

  Apache Geronimo based on Jonas, JBoss based on Jetty.

  Apache Tomcat used by IBM.

  Apache Geronimo « competing » with IBM Websphere.

Main features : Open Source software (2)

- Competitors - Competitor can became coopetitor.

  BEA and Borland join Eclipse, launched by IBM.

- Substitutes – Not a commercial product. Not in the market. Substitution relation.
- Barriers – Low barriers by licenses.
  - Necessity for other ways of differenciation.
  - Traditionnal *Commercial Software Model* doesn't work well.

Some figures about computers industry (1)

TIC market (1997, %, compared to 1994) :

Services :	$275 billions	(+53%)
Softwares :	$130 billions	(+73%)
Hardware :	$385 billions	(+67%)

Source : François Horn (2707138444 / p27)

Top 3 (2000) :

IBM :	$43,75 billions
Microsoft :	$23,845 billions
PWC :	$21,5 billions

Source : François Horn (2707138444 / p67)

License sales (1999) :

Microsoft :	$21,591 billions	(more than 90% of CA)
IBM :	$12,7 billions	(more than 25% of CA)
CA :	4,962 billions	(more than 90% of CA)

+ Oracle, Hewlett Packard, SAP,...

Source : François Horn (2707138444 / p103)

Five first suppliers of softwares = 5% (1993), 10% (1998), 21% (2000)

Source : François Horn (2707138444 / p68)

Some figures about computors industry (2)

- Concentration is a trend.
  - Some markets dominated by some big companies.

    → economic power, high barriers,...
- Services are important...
  - Installation, maintenance, process (ERP, CRM, SCM,...)
- But revenues from licenses, too !
  - Commercial edition is a kind of « mutualisation » (shared costs, low marginal cost)

Open Source business models

- Business models based on services
  - Based on components integration
  - Based on « product superstar »
- Business models based on edition
  - Based on packaging
  - Based on « mutualist » edition
  - Based on double licence
- Others

Services / components integration

- Same as proprietary services company : solutions based on third party components.
- But : importance of the collaboration with editing communities
  - Company as link between customer and community (« weak mutualisation »)
    - Always stick on main release !
  - Examples :
    - Good : Idealx and Samba, Linagora and Spip,...
    - Bad (?) : Spip vs Spipagora, forks of Spipagora (?)

Services / product superstar

- Specialization in a product / line of products
- Samples : Jboss, Zope, Jabber, Mozilla
  - Revenues often based on services
  - Jboss (« vs » Oracle) : big buzz, small company
  - Creation of a strong brand
- Interest in B2B collaboration
  - Direct cooperation (eZpublish, Jboss,...)
  - Independent structure (Objectweb, Zope Europe,...)
  - Collaboration with customer (see « mutualist edition »)

Edition / packaging

- Examples :
  - Covalent
    - Sales of licenses of products based on Apache softwares (Apache license is permissive).
    - Warning : community relations !
  - GNU/Linux distributors (SuSE, Red Hat, Mandriva, etc)
    - Integration of softwares (heterogenity of licenses, copyleft or not) in community (OpenSuse, Fedora, Cooker, etc).
    - Revenues from services (upgrade, support) and edition (box).

Edition / « mutualist » edition

- Key example : IdealX
- Principle :
  - Mutualist editor manages collaboration between customers.
  - Edition affordable for more specific needs.
  - Strong management competences needed !

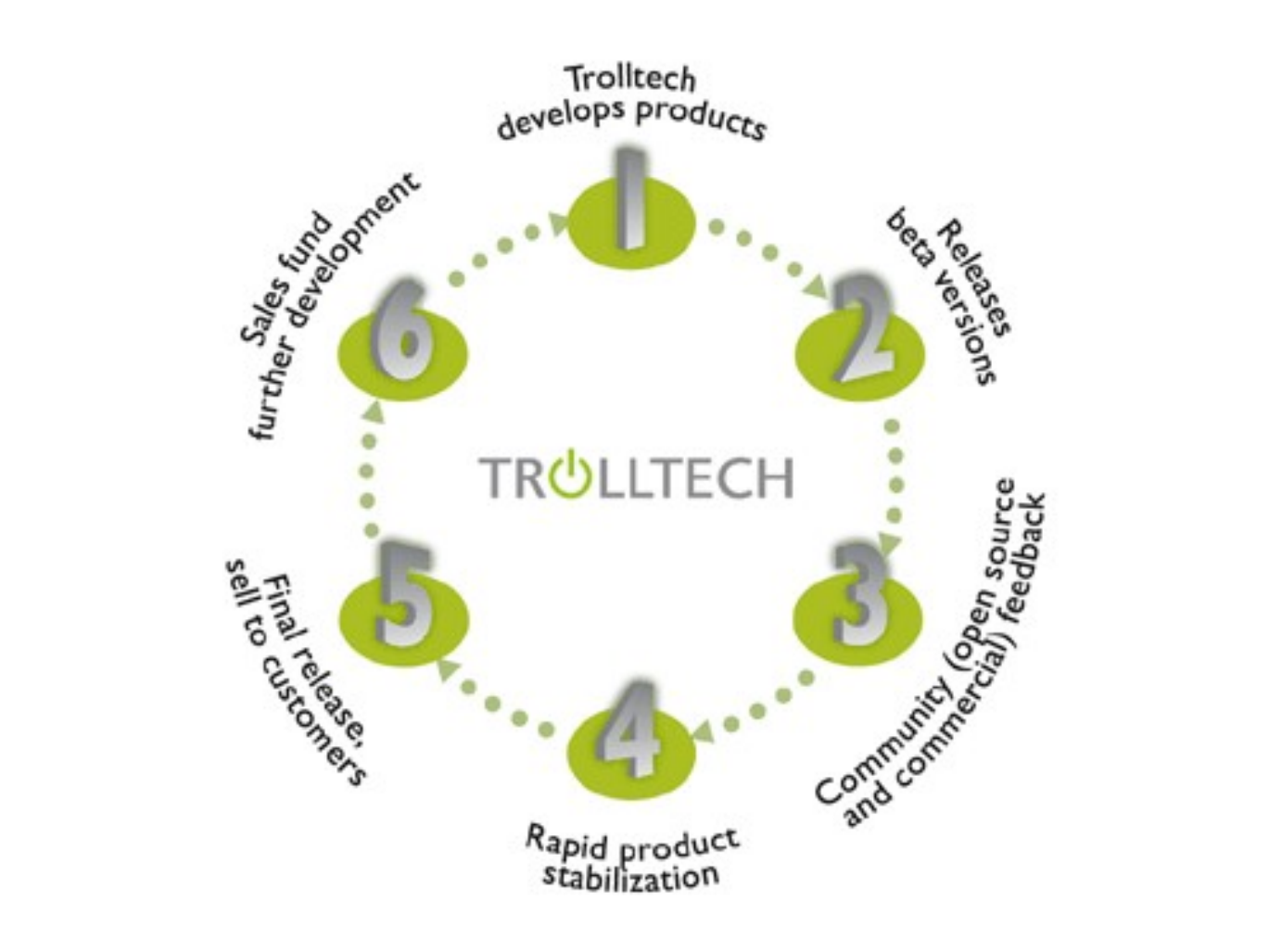
Edition / double licence (1)

- Examples : Trolltech, MySQL, Sleepycat,...
- Principles :
  - Copyleft licenses (GPL) don't permit proprietary development and business model, but is ideal for collaboration.
  - Commercial licenses permit proprietary development and finance company activities.

  → Revenues : 50 (licenses) / 50 (services) model.

  → Success of this model
- Alternative : « chronodegradable » licenses
  - Example : Ghostscrip / GsView

Edition / double licence (2)



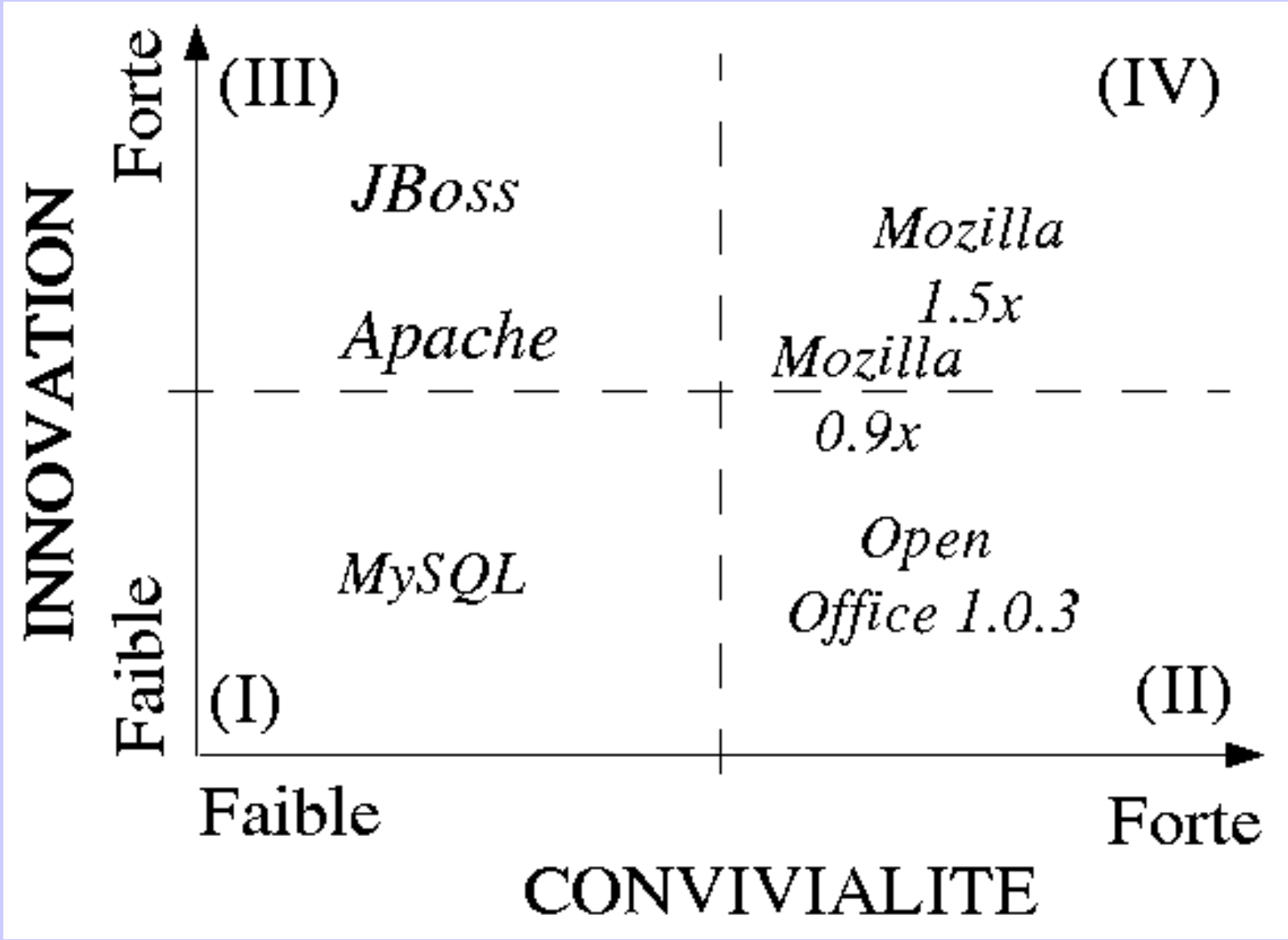
Others

- Open Source is a way for disrupting a controled market
  - Example : Open Office, Firefox, etc.
- Open source opened for mechandising
  - Example : O'Reilly
- Open the infrastructures
  - Competitive advantage is not in the ground (drivers, etc).

Proprietary and Open Source will coexist (1)

- First : Open Source software as commodities (Sendmail, Apache Web Server, etc).
- Next :
  - User end softwares (Open Office, Firefox, etc).
  - Business oriented softwares (ERP5, Compiere, SugarCRM, etc).
- New ways for proprietary industry :
  - Innovation : ASP (Salesforce.com), Web 2.0 (Google Docs), On Demand (IBM, Sun Microsystems), etc.
  - Usability (*look & feel*, extension by API, etc).

Proprietary and Open Source will coexist (2)



Simplified view - Innovation ↔ Usability (weak ↔ strong)

Open Source and company management (1)

- Best practices in co-creation :
  - Define the objectives.
  - Define the rights and the obligations of each.
  - Choose the good co-creators (segmentation)
    - Examples : B2B vs B2C, long tail vs lead users,...
  - Simplify the co-creation (toolkits, etc).
  - Control the communication (two ways).
  - If necessary, externalize the management of community.

Open Source and company management (2)

- Open Source company is opened for employees and contributors (transparency and participation).
  - Participative management outside and inside !
    - Define the rules, respect the rules.
    - More democraty in product management.
    - Community as communication vector.
      - Use it for strong brand management.

Open Source and company management (3)

- Community management as a new strong competence :
  - Well communicate
    - Who does what ?
    - Who communicates ?
    - ...
  - Choose and manage the good tools (IM, Skype, forums, etc).

Open models in future

- Knowledge of Open Source business models
- New domains :
  - Make attention to « Open » Signification !
  - Open Hardware
    - Community sharing hardware specification.
    - Examples :
      - Sun Microsystems and Sparc or picoJava.
      - Gaisler and LEON processor for ESA
  - Open Music
    - Success of Creative Commons licenses
    - Examples : Magnatune, Jamendo

Conclusion and questions

- Four key ideas for Open Source business
  - New property rules. New licenses. New business models.
  - New competence : community management. « Make software with » rather than « make software for »
  - Open Source model will not kill proprietary model. Two different ways for making business and creating value.
  - Software is a beginning.

Questions ?