Creating Value with Software

Small Business Digitization Initiative (SBDI) - Day 3

By Marc Lijour

March 22, 2017

Table of Contents

Savoir-faire LINUX

Book of the Day

Learn about Business Analysis and *Business Process Management*

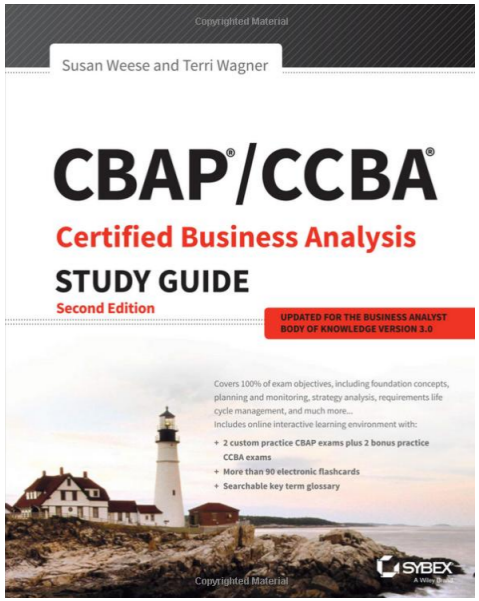


TABLE 9.17 The BPM life cycle

Activity	Description
Designing	Identifying processes and defining the current “as-is” state to determine the desired future “to-be” state, and analyzing the gap between current and future states
Modelling	Graphically representing the process to compare current and future states, and providing inputs to requirements and solution design specifications
Executing and Monitoring	Collecting data during the actual execution of the process to analyze value and recommending design improvement alternatives
Optimizing	Ongoing repetition and iteration of the other three phases to modify models and designs, remove inefficiencies, and add more value

Savoir-faire LINUX

Table of Contents

Short History of Software

The roots of Computer Science & Software Engineering

- early 17^{th} – Pascal's Calculator

Short History of Software

The roots of Computer Science & Software Engineering

- early 17^{th} – Pascal's Calculator
- 19^{th} – Charles Babbage's first *mechanical computer*, and Ada Lovelace's *Analytical Engine* and the 1^{st} *algorithm*

Short History of Software

The roots of Computer Science & Software Engineering

- early 17^{th} – Pascal's Calculator
- 19^{th} – Charles Babbage's first *mechanical computer*, and Ada Lovelace's *Analytical Engine* and the 1^{st} *algorithm*
- 1935 – Alan Turing for the first theory about *Software* (Did you see the movie "Imitation Game"?)

Short History of Software

The roots of Computer Science & Software Engineering

- early 17^{th} – Pascal's Calculator
- 19^{th} – Charles Babbage's first *mechanical computer*, and Ada Lovelace's *Analytical Engine* and the 1^{st} *algorithm*
- 1935 – Alan Turing for the first theory about *Software* (Did you see the movie "Imitation Game"?)
- late 1950's – academic field of *Computer Science*

Short History of Software

The roots of Computer Science & Software Engineering

- early 17^{th} – Pascal's Calculator
- 19^{th} – Charles Babbage's first *mechanical computer*, and Ada Lovelace's *Analytical Engine* and the 1^{st} *algorithm*
- 1935 – Alan Turing for the first theory about *Software* (Did you see the movie "Imitation Game"?)
- late 1950's – academic field of *Computer Science*
- 1968 – the term *Software Engineering* is coined, and another academic discipline is born

Short History of Software

The first big pieces of software

- 1950–60's – Software comes bundled with equipment (large and expensive); engineers can install, modify, configure

Short History of Software

The first big pieces of software

- 1950–60's – Software comes bundled with equipment (large and expensive); engineers can install, modify, configure
- 1970 – UNIX at AT&T Bell Labs

Short History of Software

The first big pieces of software

- 1950–60's – Software comes bundled with equipment (large and expensive); engineers can install, modify, configure
- 1970 – UNIX at AT&T Bell Labs
- 1980's – IBM PC and the rise of the Software Package Industry

Short History of Software

The first big pieces of software

- 1950–60's – Software comes bundled with equipment (large and expensive); engineers can install, modify, configure
- 1970 – UNIX at AT&T Bell Labs
- 1980's – IBM PC and the rise of the Software Package Industry
- 1990's – Mature desktop software, rise of the Internet

Short History of Software

The first big pieces of software

- 1950–60's – Software comes bundled with equipment (large and expensive); engineers can install, modify, configure
- 1970 – UNIX at AT&T Bell Labs
- 1980's – IBM PC and the rise of the Software Package Industry
- 1990's – Mature desktop software, rise of the Internet
- 2000's – .com bubble, e-commerce, Web2.0, Cloud, Mobile, Social Media, Video Streaming

Short History of Software

The first big pieces of software

- 1950–60's – Software comes bundled with equipment (large and expensive); engineers can install, modify, configure
- 1970 – UNIX at AT&T Bell Labs
- 1980's – IBM PC and the rise of the Software Package Industry
- 1990's – Mature desktop software, rise of the Internet
- 2000's – .com bubble, e-commerce, Web2.0, Cloud, Mobile, Social Media, Video Streaming
- 2010's – IoT, AI, VR, AR...

Savoir-faire LINUX

Short History of Software

The birth of UNIX (Pic by Peter Hamer [CC BY-SA 2.0], via Wikimedia Commons)

Figure: Ken Thompson (sitting) and Dennis Ritchie at PDP-11

Short History of Software

The Awakening of Freedom (Williams, 2002)

- Richard Stallman (RMS), *hacker* at the MIT Artificial Intelligence (AI) lab

Short History of Software

The Awakening of Freedom (Williams, 2002)

- Richard Stallman (RMS), *hacker* at the MIT Artificial Intelligence (AI) lab
- Both the *hacker* and academic cultures are about building, sharing, and improving

Short History of Software

The Awakening of Freedom (Williams, 2002)

- Richard Stallman (RMS), *hacker* at the MIT Artificial Intelligence (AI) lab
- Both the *hacker* and academic cultures are about building, sharing, and improving
- Suddenly, Xerox shockingly refuses to give the source code so RMS could make it work with the lab's new PDP11

Short History of Software

The Awakening of Freedom (Williams, 2002)

- Richard Stallman (RMS), *hacker* at the MIT Artificial Intelligence (AI) lab
- Both the *hacker* and academic cultures are about building, sharing, and improving
- Suddenly, Xerox shockingly refuses to give the source code so RMS could make it work with the lab's new PDP11
- RMS sees the printer as a Trojan horse in the AI lab

Short History of Software

The Awakening of Freedom (Williams, 2002)

- Richard Stallman (RMS), *hacker* at the MIT Artificial Intelligence (AI) lab
- Both the *hacker* and academic cultures are about building, sharing, and improving
- Suddenly, Xerox shockingly refuses to give the source code so RMS could make it work with the lab's new PDP11
- RMS sees the printer as a Trojan horse in the AI lab
- Also the rule of secrecy (NDA) violates *hacker ethics* and the *Golden Rule* (once bound to secrecy, one can't help his/her next of kind)

Savoir-faire LINUX

Short History of Software

The GNU Project and the Free Software Foundation (Stallman, 1998)

- in January 1984, RMS quits his job at MIT to write GNU software

Short History of Software

The GNU Project and the Free Software Foundation (Stallman, 1998)

- in January 1984, RMS quits his job at MIT to write GNU software
- in 1985, creates the Free Software Foundation (FSF)

Short History of Software

The GNU Project and the Free Software Foundation (Stallman, 1998)

- in January 1984, RMS quits his job at MIT to write GNU software
- in 1985, creates the Free Software Foundation (FSF)
- GNU General Public License

Short History of Software

The GNU Project and the Free Software Foundation (Stallman, 1998)

- in January 1984, RMS quits his job at MIT to write GNU software
- in 1985, creates the Free Software Foundation (FSF)
- GNU General Public License
- Software Freedom Law Center provides pro-bono legal services to developers of Free, Libre, and Open Source Software.

Short History of Software

The GNU Project and the Free Software Foundation (Stallman, 1998)

- in January 1984, RMS quits his job at MIT to write GNU software
- in 1985, creates the Free Software Foundation (FSF)
- GNU General Public License
- Software Freedom Law Center provides pro-bono legal services to developers of Free, Libre, and Open Source Software.
- St IGNUcius leads his adepts to the land of Freedoms

Savoir-faire LINUX

Short History of Software

The Four Freedoms (Stallman, 1998)

Quoting from RMS:

1. You have the freedom to run the program as you wish, for any purpose.

Short History of Software

The Four Freedoms (Stallman, 1998)

Quoting from RMS:

1. You have the freedom to run the program as you wish, for any purpose.
2. You have the freedom to modify the program to suit your needs. (To make this freedom effective in practice, you must have access to the source code, since making changes in a program without having the source code is exceedingly difficult.)

Short History of Software

The Four Freedoms (Stallman, 1998)

Quoting from RMS:

1. You have the freedom to run the program as you wish, for any purpose.
2. You have the freedom to modify the program to suit your needs. (To make this freedom effective in practice, you must have access to the source code, since making changes in a program without having the source code is exceedingly difficult.)
3. You have the freedom to redistribute copies, either gratis or for a fee.

Savoir-faire LINUX

Short History of Software

The Four Freedoms (Stallman, 1998)

Quoting from RMS:

1. You have the freedom to run the program as you wish, for any purpose.
2. You have the freedom to modify the program to suit your needs. (To make this freedom effective in practice, you must have access to the source code, since making changes in a program without having the source code is exceedingly difficult.)
3. You have the freedom to redistribute copies, either gratis or for a fee.
4. You have the freedom to distribute modified versions of the program, so that the community can benefit from your improvements.

Short History of Software

An introduction to Free/Libre Open Source Software (Intel, 2014)

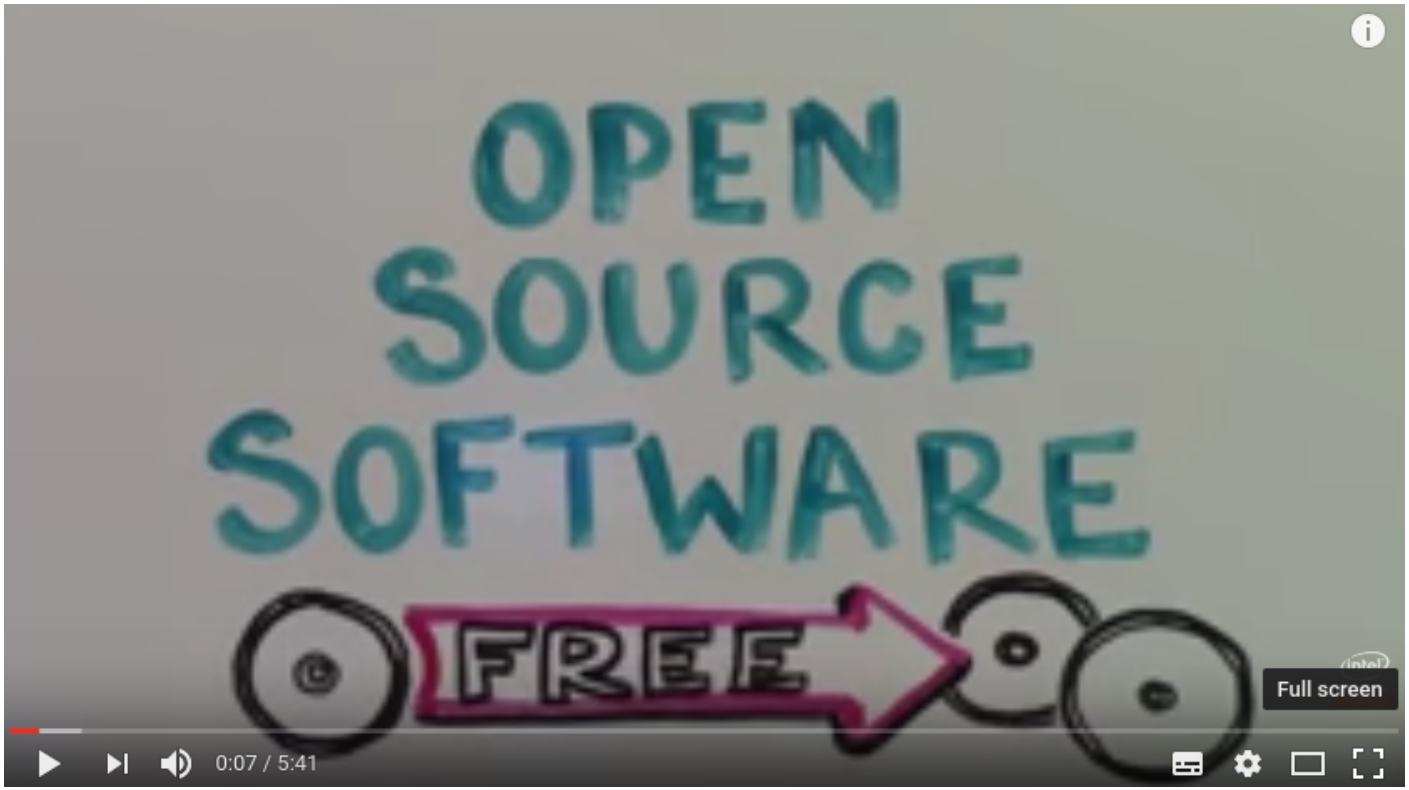


Figure: Credit: Intel Software (2014)
https://www.youtube.com/watch?v=Tyd0FO0tko8

Short History of Software

The difference between Free Software and Open Source Software (OSI, 1998)

- Open Source appears in 1998, out of a difference of perspective

Short History of Software

The difference between Free Software and Open Source Software (OSI, 1998)

- Open Source appears in 1998, out of a difference of perspective
- Open Source focuses on the development methodology and its business benefits

Short History of Software

The difference between Free Software and Open Source Software (OSI, 1998)

- Open Source appears in 1998, out of a difference of perspective
- Open Source focuses on the development methodology and its business benefits
- Free Software makes an ethical statement about user freedom and *rights*

Short History of Software

The difference between Free Software and Open Source Software (OSI, 1998)

- Open Source appears in 1998, out of a difference of perspective
- Open Source focuses on the development methodology and its business benefits
- Free Software makes an ethical statement about user freedom and *rights*
- Free/Libre Open Source Software (FLOSS) to (over)simplify

Software Licensing

Copyright & the need for Licenses

- IP Laws: Copyright, Patents, Trademark, Trade Secrets

Software Licensing

Copyright & the need for Licenses

- IP Laws: Copyright, Patents, Trademark, Trade Secrets
- Copyright law governs creative works www.cipo.ic.gc.ca/copyrights

Software Licensing

Copyright & the need for Licenses

- IP Laws: Copyright, Patents, Trademark, Trade Secrets
- Copyright law governs creative works www.cipo.ic.gc.ca/copyrights
- Copyright $\neq$ Public Domain

Software Licensing

Copyright & the need for Licenses

- IP Laws: Copyright, Patents, Trademark, Trade Secrets
- Copyright law governs creative works www.cipo.ic.gc.ca/copyrights
- Copyright $\neq$ Public Domain
- Copyright owners (often the authors) are granted exclusive rights for a period of time

Software Licensing

Copyright & the need for Licenses

- IP Laws: Copyright, Patents, Trademark, Trade Secrets
- Copyright law governs creative works www.cipo.ic.gc.ca/copyrights
- Copyright $\neq$ Public Domain
- Copyright owners (often the authors) are granted exclusive rights for a period of time
- A license is a legal document that grants some of those rights to others (e.g. software users), seldom under some conditions

Savoir-faire LINUX

Software Licensing

Copyright & the need for Licenses

- IP Laws: Copyright, Patents, Trademark, Trade Secrets
- Copyright law governs creative works www.cipo.ic.gc.ca/copyrights
- Copyright $\neq$ Public Domain
- Copyright owners (often the authors) are granted exclusive rights for a period of time
- A license is a legal document that grants some of those rights to others (e.g. software users), seldom under some conditions
- Software licenses are mostly dealing about copyright

Savoir-faire LINUX

Software Licensing

Copyright Ownership

- Individual Contributor

Software Licensing

Copyright Ownership

- Individual Contributor
- Employee working for a company

Software Licensing

Copyright Ownership

- Individual Contributor
- Employee working for a company
- Individuals & companies might want to assign their copyright (e.g. at the FSF)

Software Licensing

Copyright Ownership

- Individual Contributor
- Employee working for a company
- Individuals & companies might want to assign their copyright (e.g. at the FSF)
- Only copyright owners can sue

Software Licensing

Copyleft

- a twist on “Copyright”

Software Licensing

Copyleft

- a twist on “Copyright”
- Copyright licenses are often triggered by “distribution”

Software Licensing

Copyleft

- a twist on “Copyright”
- Copyright licenses are often triggered by “distribution”
- A spectrum of licenses with diverse conditions & limitations
  - Permissive Licenses (e.g. BSD License)

Software Licensing

Copyleft

- a twist on “Copyright”
- Copyright licenses are often triggered by “distribution”
- A spectrum of licenses with diverse conditions & limitations
  - Permissive Licenses (e.g. BSD License)
  - Weak Licenses (e.g. GNU Lesser General Public License)

Software Licensing

Copyleft

- a twist on “Copyright”
- Copyright licenses are often triggered by “distribution”
- A spectrum of licenses with diverse conditions & limitations
  - Permissive Licenses (e.g. BSD License)
  - Weak Licenses (e.g. GNU Lesser General Public License)
  - Viral Licenses (e.g. GNU General Public License)

Savoir-faire LINUX

Software Licensing

Copyleft

- a twist on “Copyright”
- Copyright licenses are often triggered by “distribution”
- A spectrum of licenses with diverse conditions & limitations
  - Permissive Licenses (e.g. BSD License)
  - Weak Licenses (e.g. GNU Lesser General Public License)
  - Viral Licenses (e.g. GNU General Public License)
- Other licenses (e.g. for documentation)

Savoir-faire LINUX

Software Licensing

Copyleft

- a twist on “Copyright”
- Copyright licenses are often triggered by “distribution”
- A spectrum of licenses with diverse conditions & limitations
  - Permissive Licenses (e.g. BSD License)
  - Weak Licenses (e.g. GNU Lesser General Public License)
  - Viral Licenses (e.g. GNU General Public License)
- Other licenses (e.g. for documentation)
- Derived works and the chain of title (Rosen, 2005)

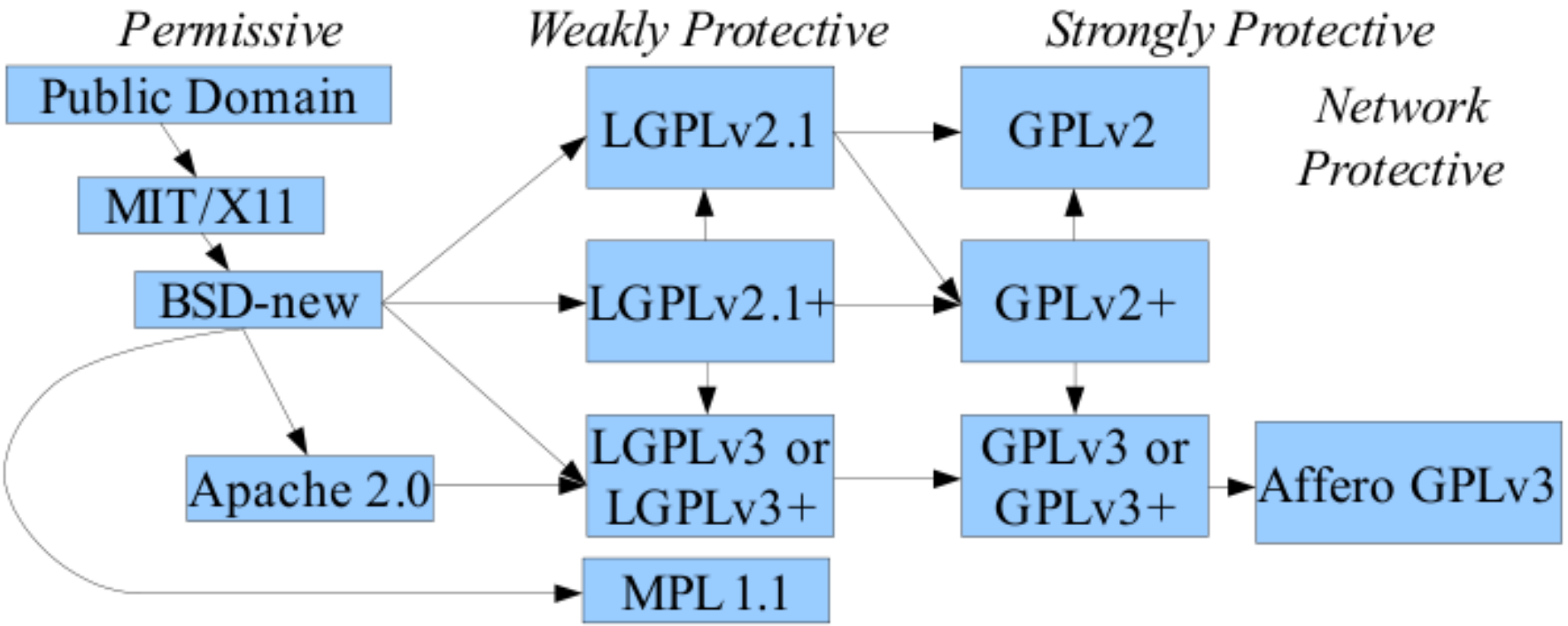
Software Licensing

Copyleft

- a twist on “Copyright”
- Copyright licenses are often triggered by “distribution”
- A spectrum of licenses with diverse conditions & limitations
  - Permissive Licenses (e.g. BSD License)
  - Weak Licenses (e.g. GNU Lesser General Public License)
  - Viral Licenses (e.g. GNU General Public License)
- Other licenses (e.g. for documentation)
- Derived works and the chain of title (Rosen, 2005)
- Mixing licenses –the question of license compatibility

Software Licensing

FLOSS License Compatibility (Picture by David A. Wheeler [CC BY-SA 3.0], via Wikimedia Commons)



Software Licensing

Managing Legal Complexity

- reduce legal risk

Software Licensing

Managing Legal Complexity

- reduce legal risk
- provide clarity and education

Software Licensing

Managing Legal Complexity

- reduce legal risk
- provide clarity and education
- best practice: internal company policy (Meeker, 2008)

Software Licensing

Managing Legal Complexity

- reduce legal risk
- provide clarity and education
- best practice: internal company policy (Meeker, 2008)
- understand how that plays out with company strategy

Software Licensing

Example: Odoo

- AGPL up to Odoo 8.0

Software Licensing

Example: Odoo

- AGPL up to Odoo 8.0
- LGPL from Odoo 9.0 forward –see the announcement

Software Licensing

Example: Odoo

- AGPL up to Odoo 8.0
- LGPL from Odoo 9.0 forward –see the announcement
- A change of business model

Software Licensing

Example: Odoo

- AGPL up to Odoo 8.0
- LGPL from Odoo 9.0 forward –see the announcement
- A change of business model
  - AGPL favours system integrators
  - LGPL favours the individual developers and the growth of an App Store

Software Licensing

Example: Odoo

- AGPL up to Odoo 8.0
- LGPL from Odoo 9.0 forward –see the announcement
- A change of business model
  - AGPL favours system integrators
  - LGPL favours the individual developers and the growth of an App Store
- What are the implication of switching to another license?

Savoir-faire LINUX

FLOSS and Business Strategy

FLOSS is everywhere

- 498 out of 500 supercomputers run Linux (see Linux.com, November 2016)

FLOSS and Business Strategy

FLOSS is everywhere

- 498 out of 500 supercomputers run Linux (see Linux.com, November 2016)
- Microsoft joined The Linux Foundation as a Platinum Member

FLOSS and Business Strategy

FLOSS is everywhere

- 498 out of 500 supercomputers run Linux (see Linux.com, November 2016)
- Microsoft joined The Linux Foundation as a Platinum Member
- Close to half the workload on Microsoft Azure run FLOSS

FLOSS and Business Strategy

FLOSS is everywhere

- 498 out of 500 supercomputers run Linux (see Linux.com, November 2016)
- Microsoft joined The Linux Foundation as a Platinum Member
- Close to half the workload on Microsoft Azure run FLOSS
- FLOSS is in the Space Station, Watches, Cars...

FLOSS and Business Strategy

FLOSS is everywhere

- 498 out of 500 supercomputers run Linux (see Linux.com, November 2016)
- Microsoft joined The Linux Foundation as a Platinum Member
- Close to half the workload on Microsoft Azure run FLOSS
- FLOSS is in the Space Station, Watches, Cars...
- Developers seem to take FLOSS for granted

FLOSS and Business Strategy

FLOSS is everywhere

- 498 out of 500 supercomputers run Linux (see Linux.com, November 2016)
- Microsoft joined The Linux Foundation as a Platinum Member
- Close to half the workload on Microsoft Azure run FLOSS
- FLOSS is in the Space Station, Watches, Cars...
- Developers seem to take FLOSS for granted
- The Open Source methodology is proven to be best (e.g. InnerSource at PayPal, HR at WIPRO)

Savoir-faire LINUX

FLOSS and Business Strategy

FLOSS is everywhere

- 498 out of 500 supercomputers run Linux (see Linux.com, November 2016)
- Microsoft joined The Linux Foundation as a Platinum Member
- Close to half the workload on Microsoft Azure run FLOSS
- FLOSS is in the Space Station, Watches, Cars...
- Developers seem to take FLOSS for granted
- The Open Source methodology is proven to be best (e.g. InnerSource at PayPal, HR at WIPRO)
- GitHub has 14M registered users

Savoir-faire LINUX

FLOSS and Business Strategy

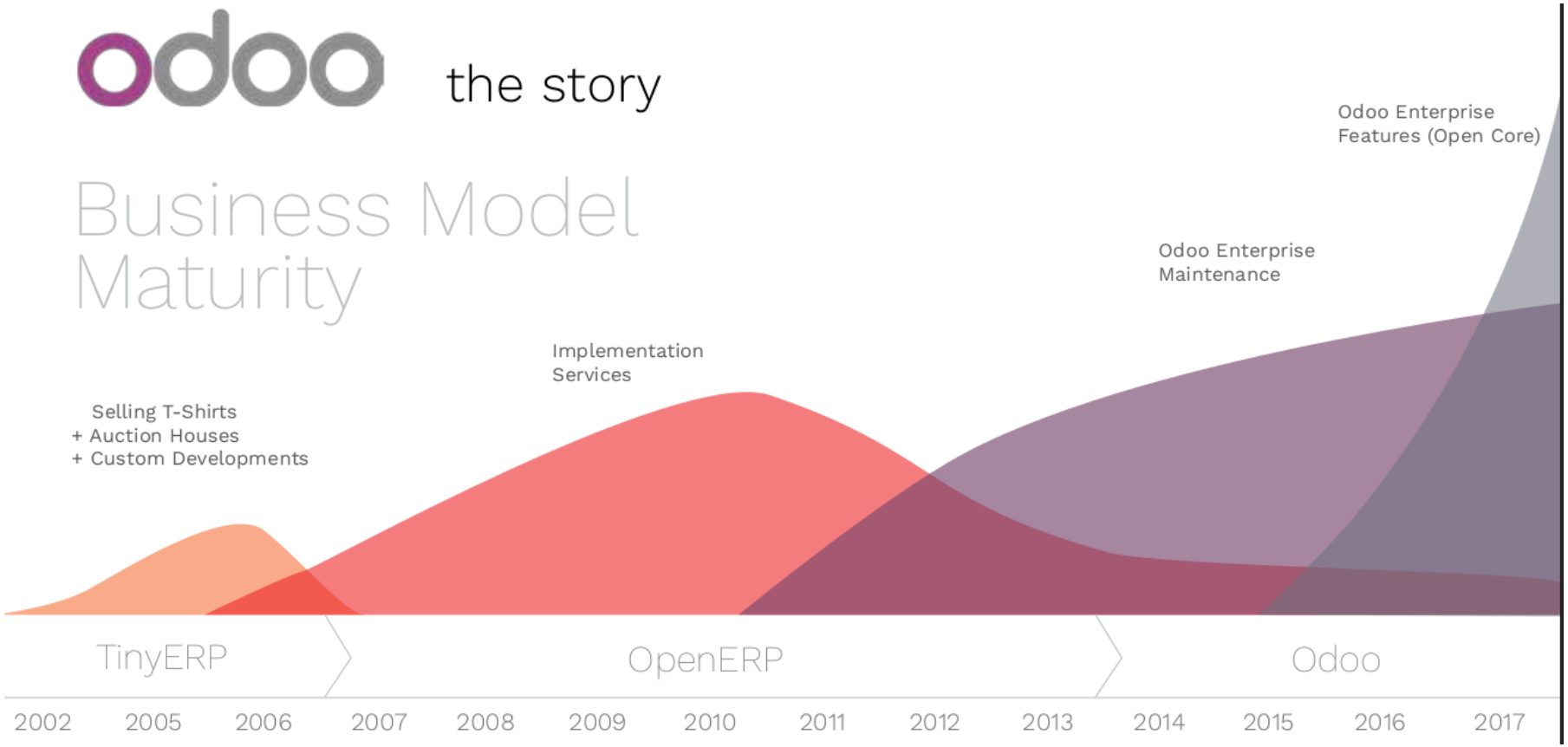
FLOSS is everywhere

- 498 out of 500 supercomputers run Linux (see Linux.com, November 2016)
- Microsoft joined The Linux Foundation as a Platinum Member
- Close to half the workload on Microsoft Azure run FLOSS
- FLOSS is in the Space Station, Watches, Cars...
- Developers seem to take FLOSS for granted
- The Open Source methodology is proven to be best (e.g. InnerSource at PayPal, HR at WIPRO)
- GitHub has 14M registered users
- In 2014, VCs invested $2.4B into FLOSS-focused companies

Savoir-faire LINUX

FLOSS and Business Strategy

How to make money with Open Source: some popular business models for providers



FLOSS and Business Strategy

How to make money with Open Source: some popular business models for providers

- Create your start up (e.g. Google, Facebook, Twitter)

FLOSS and Business Strategy

How to make money with Open Source: some popular business models for providers

- Create your start up (e.g. Google, Facebook, Twitter)
- System Integrator (e.g. Savoir-faire Linux)

FLOSS and Business Strategy

How to make money with Open Source: some popular business models for providers

- Create your start up (e.g. Google, Facebook, Twitter)
- System Integrator (e.g. Savoir-faire Linux)
- Software Publisher (e.g. Odoo SA, Red Hat)

FLOSS and Business Strategy

How to make money with Open Source: some popular business models for providers

- Create your start up (e.g. Google, Facebook, Twitter)
- System Integrator (e.g. Savoir-faire Linux)
- Software Publisher (e.g. Odoo SA, Red Hat)
- Platform play –to generate network externalities (e.g. Google Android, and Chrome/Chromium)

Savoir-faire LINUX

FLOSS and Business Strategy

How to make money with Open Source: some popular business models for providers

- Create your start up (e.g. Google, Facebook, Twitter)
- System Integrator (e.g. Savoir-faire Linux)
- Software Publisher (e.g. Odoo SA, Red Hat)
- Platform play –to generate network externalities (e.g. Google Android, and Chrome/Chromium)
- Outsourcing and cost-sharing (e.g. CMC and the Bombardier C-Series case study)

Savoir-faire LINUX

FLOSS and Business Strategy

How to make money with Open Source: some popular business models for providers

- Create your start up (e.g. Google, Facebook, Twitter)
- System Integrator (e.g. Savoir-faire Linux)
- Software Publisher (e.g. Odoo SA, Red Hat)
- Platform play –to generate network externalities (e.g. Google Android, and Chrome/Chromium)
- Outsourcing and cost-sharing (e.g. CMC and the Bombardier C-Series case study)
- Setting the pace and imposing standards (e.g. Google Tensor Flow)

Savoir-faire LINUX

FLOSS and Business Strategy

How to make money with Open Source: some popular business models for providers

- Create your start up (e.g. Google, Facebook, Twitter)
- System Integrator (e.g. Savoir-faire Linux)
- Software Publisher (e.g. Odoo SA, Red Hat)
- Platform play –to generate network externalities (e.g. Google Android, and Chrome/Chromium)
- Outsourcing and cost-sharing (e.g. CMC and the Bombardier C-Series case study)
- Setting the pace and imposing standards (e.g. Google Tensor Flow)
- Software as a Service (e.g. Odoo cloud service)

Savoir-faire LINUX

FLOSS and Business Strategy

How to make money with Open Source: some popular business models for providers

- Create your start up (e.g. Google, Facebook, Twitter)
- System Integrator (e.g. Savoir-faire Linux)
- Software Publisher (e.g. Odoo SA, Red Hat)
- Platform play –to generate network externalities (e.g. Google Android, and Chrome/Chromium)
- Outsourcing and cost-sharing (e.g. CMC and the Bombardier C-Series case study)
- Setting the pace and imposing standards (e.g. Google Tensor Flow)
- Software as a Service (e.g. Odoo cloud service)
- Cutting down cost of R&D (e.g. Linksys routers, Tom Tom GPS, Bosh)

Savoir-faire LINUX

FLOSS and Business Strategy

How to make money with Open Source: some popular business models for providers

- Create your start up (e.g. Google, Facebook, Twitter)
- System Integrator (e.g. Savoir-faire Linux)
- Software Publisher (e.g. Odoo SA, Red Hat)
- Platform play –to generate network externalities (e.g. Google Android, and Chrome/Chromium)
- Outsourcing and cost-sharing (e.g. CMC and the Bombardier C-Series case study)
- Setting the pace and imposing standards (e.g. Google Tensor Flow)
- Software as a Service (e.g. Odoo cloud service)
- Cutting down cost of R&D (e.g. Linksys routers, Tom Tom GPS, Bosh)
- Open Core (e.g. Odoo)

Savoir-faire LINUX

FLOSS and Business Strategy

How to make money with Open Source: some popular business models for providers

- Create your start up (e.g. Google, Facebook, Twitter)
- System Integrator (e.g. Savoir-faire Linux)
- Software Publisher (e.g. Odoo SA, Red Hat)
- Platform play –to generate network externalities (e.g. Google Android, and Chrome/Chromium)
- Outsourcing and cost-sharing (e.g. CMC and the Bombardier C-Series case study)
- Setting the pace and imposing standards (e.g. Google Tensor Flow)
- Software as a Service (e.g. Odoo cloud service)
- Cutting down cost of R&D (e.g. Linksys routers, Tom Tom GPS, Bosh)
- Open Core (e.g. Odoo)
- Industry Consortiums (e.g. Eclipse, Linux Foundation and its many projects)

FLOSS and Business Strategy

The consumer perspective: how to take advantage of FLOSS

- Free as in Freedom, and as Free to try (low barrier to adoption) –leading to an innovation culture and technological leadership

FLOSS and Business Strategy

The consumer perspective: how to take advantage of FLOSS

- Free as in Freedom, and as Free to try (low barrier to adoption) –leading to an innovation culture and technological leadership
- Extreme customization: fit software to the business, and not the other way around (on the other end proprietary software can be slow to adopt by employees sticking to the old ways)

FLOSS and Business Strategy

The consumer perspective: how to take advantage of FLOSS

- Free as in Freedom, and as Free to try (low barrier to adoption) –leading to an innovation culture and technological leadership
- Extreme customization: fit software to the business, and not the other way around (on the other end proprietary software can be slow to adopt by employees sticking to the old ways)
- Control cost (in a monopolistic scenario, software vendors can stop investing in R&D and/or ignore customer complaints)

FLOSS and Business Strategy

The consumer perspective: how to take advantage of FLOSS

- Free as in Freedom, and as Free to try (low barrier to adoption) –leading to an innovation culture and technological leadership
- Extreme customization: fit software to the business, and not the other way around (on the other end proprietary software can be slow to adopt by employees sticking to the old ways)
- Control cost (in a monopolistic scenario, software vendors can stop investing in R&D and/or ignore customer complaints)
- Reduce and eliminate vendor lock-in (asking for APIs, ability to run FLOSS)

Savoir-faire LINUX

FLOSS and Business Strategy

The consumer perspective: how to take advantage of FLOSS

- Free as in Freedom, and as Free to try (low barrier to adoption) –leading to an innovation culture and technological leadership
- Extreme customization: fit software to the business, and not the other way around (on the other end proprietary software can be slow to adopt by employees sticking to the old ways)
- Control cost (in a monopolistic scenario, software vendors can stop investing in R&D and/or ignore customer complaints)
- Reduce and eliminate vendor lock-in (asking for APIs, ability to run FLOSS)
- Security, Privacy, and Control (especially against the "cloud" solutions)

Savoir-faire LINUX

FLOSS and Business Strategy

The consumer perspective: how to take advantage of FLOSS

- Free as in Freedom, and as Free to try (low barrier to adoption) –leading to an innovation culture and technological leadership
- Extreme customization: fit software to the business, and not the other way around (on the other end proprietary software can be slow to adopt by employees sticking to the old ways)
- Control cost (in a monopolistic scenario, software vendors can stop investing in R&D and/or ignore customer complaints)
- Reduce and eliminate vendor lock-in (asking for APIs, ability to run FLOSS)
- Security, Privacy, and Control (especially against the "cloud" solutions)
- Sovereign Strategy: maintain and nurture core competencies to secure a competitive advantage

FLOSS and Business Strategy

The consumer perspective: how to take advantage of FLOSS

- Free as in Freedom, and as Free to try (low barrier to adoption) –leading to an innovation culture and technological leadership
- Extreme customization: fit software to the business, and not the other way around (on the other end proprietary software can be slow to adopt by employees sticking to the old ways)
- Control cost (in a monopolistic scenario, software vendors can stop investing in R&D and/or ignore customer complaints)
- Reduce and eliminate vendor lock-in (asking for APIs, ability to run FLOSS)
- Security, Privacy, and Control (especially against the "cloud" solutions)
- Sovereign Strategy: maintain and nurture core competencies to secure a competitive advantage
- Creates and strengthens a local labour market and healthy competition (same as the ability to choose between a car agency and the neighbourhood mechanic)

FLOSS and Business Strategy

Individuals benefit the most from *user rights* and *Freedoms*

- Privacy vs. Proprietary software "spying" (e.g. always-on systems like Amazon Alexa pose a real threat to privacy)

FLOSS and Business Strategy

Individuals benefit the most from *user rights* and *Freedoms*

- Privacy vs. Proprietary software "spying" (e.g. always-on systems like Amazon Alexa pose a real threat to privacy)
- Ability to use one's own media vs. DRM and other blocking mechanisms (see https://www.defectivebydesign.org)

FLOSS and Business Strategy

Individuals benefit the most from *user rights* and *Freedoms*

- Privacy vs. Proprietary software "spying" (e.g. always-on systems like Amazon Alexa pose a real threat to privacy)
- Ability to use one's own media vs. DRM and other blocking mechanisms (see https://www.defectivebydesign.org)
- Software benefits flow down to end-users vs. Tivoization

FLOSS and Business Strategy

Individuals benefit the most from *user rights* and *Freedoms*

- Privacy vs. Proprietary software "spying" (e.g. always-on systems like Amazon Alexa pose a real threat to privacy)
- Ability to use one's own media vs. DRM and other blocking mechanisms (see https://www.defectivebydesign.org)
- Software benefits flow down to end-users vs. Tivoization
- Equal and fair access to Internet and Computing Resources (e.g. Net Neutrality)

Savoir-faire LINUX

FLOSS and Business Strategy

Individuals benefit the most from *user rights* and *Freedoms*

- Privacy vs. Proprietary software "spying" (e.g. always-on systems like Amazon Alexa pose a real threat to privacy)
- Ability to use one's own media vs. DRM and other blocking mechanisms (see https://www.defectivebydesign.org)
- Software benefits flow down to end-users vs. Tivoization
- Equal and fair access to Internet and Computing Resources (e.g. Net Neutrality)
- Keeping democracies honest and incentivizing civic engagement (e.g. voting machines, Code.org)

FLOSS and Business Strategy

The drawback of FLOSS (Eghbal, 2016)

- Key infrastructure projects lack funding (Eghbal, 2016)

FLOSS and Business Strategy

The drawback of FLOSS (Eghbal, 2016)

- Key infrastructure projects lack funding (Eghbal, 2016)
- Companies could contribute more

FLOSS and Business Strategy

The drawback of FLOSS (Eghbal, 2016)

- Key infrastructure projects lack funding (Eghbal, 2016)
- Companies could contribute more
- There are not enough maintainers vs. contributors

FLOSS and Business Strategy

The drawback of FLOSS (Eghbal, 2016)

- Key infrastructure projects lack funding (Eghbal, 2016)
- Companies could contribute more
- There are not enough maintainers vs. contributors
- A lot too many people don't care about licensing (e.g. 85% of the projects on GitHub don't have a license)

FLOSS and Business Strategy

The drawback of FLOSS (Eghbal, 2016)

- Key infrastructure projects lack funding (Eghbal, 2016)
- Companies could contribute more
- There are not enough maintainers vs. contributors
- A lot too many people don't care about licensing (e.g. 85% of the projects on GitHub don't have a license)
- Is FLOSS dying from its success (like smartphones are now called phones)? (see Nadia Eghbal)

Savoir-faire LINUX

FLOSS and Business Strategy

Make the world a better place: examples from the FSF High Priority List

- Real-time voice and video chat – try Ring beta2 lead by Savoir-faire Linux

FLOSS and Business Strategy

Make the world a better place: examples from the FSF High Priority List

- Real-time voice and video chat – try Ring beta2 lead by Savoir-faire Linux
- Free phone operating system (e.g. Replicant removes the pieces that are not free in Android)

FLOSS and Business Strategy

Make the world a better place: examples from the FSF High Priority List

- Real-time voice and video chat – try Ring beta2 lead by Savoir-faire Linux
- Free phone operating system (e.g. Replicant removes the pieces that are not free in Android)
- Better hardware/firmware, drivers, and 3D capabilities

FLOSS and Business Strategy

Make the world a better place: examples from the FSF High Priority List

- Real-time voice and video chat – try Ring beta2 lead by Savoir-faire Linux
- Free phone operating system (e.g. Replicant removes the pieces that are not free in Android)
- Better hardware/firmware, drivers, and 3D capabilities
- Accessibility

FLOSS and Business Strategy

Make the world a better place: examples from the FSF High Priority List

- Real-time voice and video chat – try Ring beta2 lead by Savoir-faire Linux
- Free phone operating system (e.g. Replicant removes the pieces that are not free in Android)
- Better hardware/firmware, drivers, and 3D capabilities
- Accessibility
- Security

Acquiring an IT Solutions

Consider the pros &cons of the main options

- Build

Acquiring an IT Solutions

Consider the pros &cons of the main options

- Build
- Buy (for a time: license)

Acquiring an IT Solutions

Consider the pros &cons of the main options

- Build
- Buy (for a time: license)
- Rent (as a service)

Acquiring an IT Solutions

Consider the pros &cons of the main options

- Build
- Buy (for a time: license)
- Rent (as a service)
- Collaborate

Acquiring an IT Solutions

Consider the pros &cons of the main options

- Build
- Buy (for a time: license)
- Rent (as a service)
- Collaborate
- Other?

Acquiring an IT Solutions

Best choice will depend on the situation

- Do you even have a choice? Sometimes there is no software out there that fits the bill. . .

Acquiring an IT Solutions

Best choice will depend on the situation

- Do you even have a choice? Sometimes there is no software out there that fits the bill...
- How expensive is the alternative out there? (e.g. video processing system developed by Savoir-faire Linux for the TV Industry)

Acquiring an IT Solutions

Best choice will depend on the situation

- Do you even have a choice? Sometimes there is no software out there that fits the bill...
- How expensive is the alternative out there? (e.g. video processing system developed by Savoir-faire Linux for the TV Industry)
- Do you have the capabilities (in-house) to build and to maintain a software you need?

Acquiring an IT Solutions

Best choice will depend on the situation

- Do you even have a choice? Sometimes there is no software out there that fits the bill...
- How expensive is the alternative out there? (e.g. video processing system developed by Savoir-faire Linux for the TV Industry)
- Do you have the capabilities (in-house) to build and to maintain a software you need?
- Are these capabilities a core competency for your business?

Savoir-faire LINUX

Acquiring an IT Solutions

Best choice will depend on the situation

- Do you even have a choice? Sometimes there is no software out there that fits the bill...
- How expensive is the alternative out there? (e.g. video processing system developed by Savoir-faire Linux for the TV Industry)
- Do you have the capabilities (in-house) to build and to maintain a software you need?
- Are these capabilities a core competency for your business?
- Is data and intelligence critical to your business?

Savoir-faire LINUX

Acquiring an IT Solutions

Best choice will depend on the situation

- Do you even have a choice? Sometimes there is no software out there that fits the bill...
- How expensive is the alternative out there? (e.g. video processing system developed by Savoir-faire Linux for the TV Industry)
- Do you have the capabilities (in-house) to build and to maintain a software you need?
- Are these capabilities a core competency for your business?
- Is data and intelligence critical to your business?
- Can you live with a commodity (low-cost) solution that fits somewhat most but not all of your needs?

Savoir-faire LINUX

Acquiring an IT Solutions

Best choice will depend on the situation

- Do you even have a choice? Sometimes there is no software out there that fits the bill...
- How expensive is the alternative out there? (e.g. video processing system developed by Savoir-faire Linux for the TV Industry)
- Do you have the capabilities (in-house) to build and to maintain a software you need?
- Are these capabilities a core competency for your business?
- Is data and intelligence critical to your business?
- Can you live with a commodity (low-cost) solution that fits somewhat most but not all of your needs?
- Do you want support locally or are you fine with off-shore support?

Acquiring an IT Solutions

Comparing Apples to Apples

Comparing purchase price of FLOSS vs. Proprietary Solutions

FLOSS can be downloaded for free, while proprietary software requires a payment. Their cost structure differs. How do we compare apples to apples?

Acquiring an IT Solutions

Comparing Apples to Apples: Total Cost of Ownership (TCO)

- Cost of building or buying

Acquiring an IT Solutions

Comparing Apples to Apples: Total Cost of Ownership (TCO)

- Cost of building or buying
- Cost of maintainance

Acquiring an IT Solutions

Comparing Apples to Apples: Total Cost of Ownership (TCO)

- Cost of building or buying
- Cost of maintainance
- Cost of replacing (refurbishing, getting the upgrade, migrating data)

Acquiring an IT Solutions

Comparing Apples to Apples: Total Cost of Ownership (TCO)

- Cost of building or buying
- Cost of maintainance
- Cost of replacing (refurbishing, getting the upgrade, migrating data)
- Cost of tailoring to the business

Acquiring an IT Solutions

Comparing Apples to Apples: Total Cost of Ownership (TCO)

- Cost of building or buying
- Cost of maintainance
- Cost of replacing (refurbishing, getting the upgrade, migrating data)
- Cost of tailoring to the business
- Cost of security

Acquiring an IT Solutions

Comparing Apples to Apples: Total Cost of Ownership (TCO)

- Cost of building or buying
- Cost of maintainance
- Cost of replacing (refurbishing, getting the upgrade, migrating data)
- Cost of tailoring to the business
- Cost of security
- Cost of risk (e.g. provider bankrupcy, product phased out, patent case)

Acquiring an IT Solutions

Comparing Apples to Apples: Total Cost of Ownership (TCO)

- Cost of building or buying
- Cost of maintainance
- Cost of replacing (refurbishing, getting the upgrade, migrating data)
- Cost of tailoring to the business
- Cost of security
- Cost of risk (e.g. provider bankrupcy, product phased out, patent case)
- Cost of the team (HR, facilities)

Acquiring an IT Solutions

Comparing Apples to Apples: Total Cost of Ownership (TCO)

- Cost of building or buying
- Cost of maintainance
- Cost of replacing (refurbishing, getting the upgrade, migrating data)
- Cost of tailoring to the business
- Cost of security
- Cost of risk (e.g. provider bankrupcy, product phased out, patent case)
- Cost of the team (HR, facilities)
- Other costs (see https://www.business-case-analysis.com/total-cost-of-ownership.html)

Acquiring an IT Solutions

Think IT Strategy before planning the procurement process

- FLOSS can generate 90% saving *for the right projects* –see http://oss-watch.ac.uk/resources/procurement-infopack

Acquiring an IT Solutions

Think IT Strategy before planning the procurement process

- FLOSS can generate 90% saving *for the right projects* –see http://oss-watch.ac.uk/resources/procurement-infopack
- Can big projects be broken down in smaller, manageable projects? (Allow small firms to compete.)

Acquiring an IT Solutions

Think IT Strategy before planning the procurement process

- FLOSS can generate 90% saving *for the right projects* –see http://oss-watch.ac.uk/resources/procurement-infopack
- Can big projects be broken down in smaller, manageable projects? (Allow small firms to compete.)
- Can the intended solution be developed incrementally over time? It is often the best approach for ERP projects, as people need time to learn the system and to adjust their practice.

Acquiring an IT Solutions

Think IT Strategy before planning the procurement process

- FLOSS can generate 90% saving *for the right projects* –see http://oss-watch.ac.uk/resources/procurement-infopack
- Can big projects be broken down in smaller, manageable projects? (Allow small firms to compete.)
- Can the intended solution be developed incrementally over time? It is often the best approach for ERP projects, as people need time to learn the system and to adjust their practice.
- Eliminate unnecessary barriers (leave open door for all types of software, FLOSS or proprietary, and all types of businesses)

Savoir-faire LINUX

Acquiring an IT Solutions

Think IT Strategy before planning the procurement process

- FLOSS can generate 90% saving *for the right projects* –see http://oss-watch.ac.uk/resources/procurement-infopack
- Can big projects be broken down in smaller, manageable projects? (Allow small firms to compete.)
- Can the intended solution be developed incrementally over time? It is often the best approach for ERP projects, as people need time to learn the system and to adjust their practice.
- Eliminate unnecessary barriers (leave open door for all types of software, FLOSS or proprietary, and all types of businesses)
- Have a back-up plan in case the solution goes bust

Savoir-faire LINUX

Acquiring an IT Solutions

Implementing the Solution

- ERPs deal with business process (i.e. an opportunity for process innovation)

Acquiring an IT Solutions

Implementing the Solution

- ERPs deal with business process (i.e. an opportunity for process innovation)
- Change is hard –use a sound methodology (e.g. Kotter's 8-step process)

Acquiring an IT Solutions

Implementing the Solution

- ERPs deal with business process (i.e. an opportunity for process innovation)
- Change is hard –use a sound methodology (e.g. Kotter's 8-step process)
- Incremental Innovation: improve a process, then automate it in software, then learn from the new process, then improve it...

Acquiring an IT Solutions

Implementing the Solution

- ERPs deal with business process (i.e. an opportunity for process innovation)
- Change is hard –use a sound methodology (e.g. Kotter's 8-step process)
- Incremental Innovation: improve a process, then automate it in software, then learn from the new process, then improve it...
- Example: opening an e-commerce service

Creating Value with Software

References

Eghbal, N. (2016, July). *Roads and Bridges: The Unseen Labor Behind Our Digital Infrastructure*. Retrieved from http://www.fordfoundation.org/library/reports-and-studies/roads-and-bridges-the-unseen-labor-behind-our-digital-infrastructure

Intel. (2014). Open source basics. Retrieved from https://www.youtube.com/watch?v=Tyd0FO0tko8

Meeker, H. (2008). *The open source alternative: understanding risks and leveraging opportunities*. Wiley.

OSI. (1998). History of the Open Source Initiative (OSI). Retrieved from https://opensource.org/history

Rosen, L. (2005). *Open Source Licensing: Software Freedom and Intellectual Property Law*. Prentice Hall PTR.

Stallman, R. M. (1998). The GNU Project. Retrieved from https://www.gnu.org/gnu/thegnuproject.html

Williams, S. (2002, March). *Free as in Freedom*. Retrieved from http://www.oreilly.com/openbook/freedom/

Table of Contents

Table of Contents

Savoir-faire LINUX

Salary discrepancy across regions

Note

California salaries are $15–$20K higher than any other part of the US, with a median of $139K.

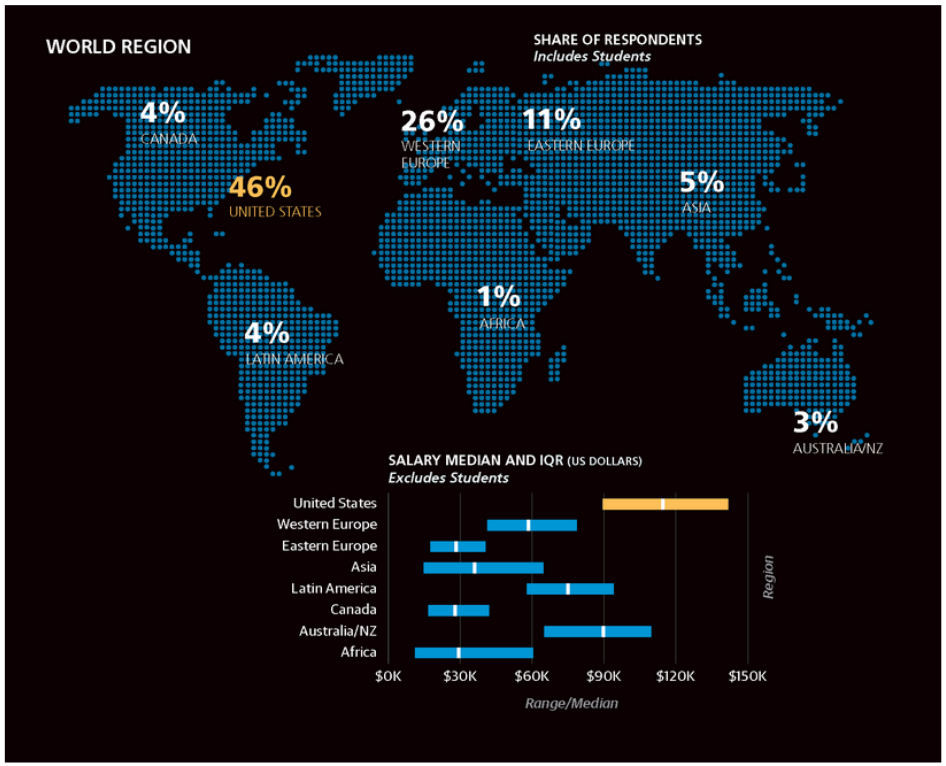


Figure: see O'Reilly Salary Survey (Suda and Magoulas, 2017)

References I

Suda, B. & Magoulas, R. (2017, April). 2017 software development salary survey. *O'Reilly*. Retrieved from https://www.oreilly.com/ideas/2017-software-development-salary-survey

Table of Contents

Global Competition

Purchasing Power Parity (PPP)

- Idea that a basket of common goods would have the same pricing value across countries and currencies

Global Competition

Purchasing Power Parity (PPP)

- Idea that a basket of common goods would have the same pricing value across countries and currencies
- Assumptions:
  - no transaction cost (currency exchange)

Global Competition

Purchasing Power Parity (PPP)

- Idea that a basket of common goods would have the same pricing value across countries and currencies
- Assumptions:
  - no transaction cost (currency exchange)
  - no trade barriers

Global Competition

Purchasing Power Parity (PPP)

- Idea that a basket of common goods would have the same pricing value across countries and currencies
- Assumptions:
  - no transaction cost (currency exchange)
  - no trade barriers
- Useful to think about:
  - Arbitrage

Global Competition

Purchasing Power Parity (PPP)

- Idea that a basket of common goods would have the same pricing value across countries and currencies
- Assumptions:
  - no transaction cost (currency exchange)
  - no trade barriers
- Useful to think about:
  - Arbitrage
  - Undervaluation and overvaluation due to market expectation (e.g. US President Trump & Mexico)

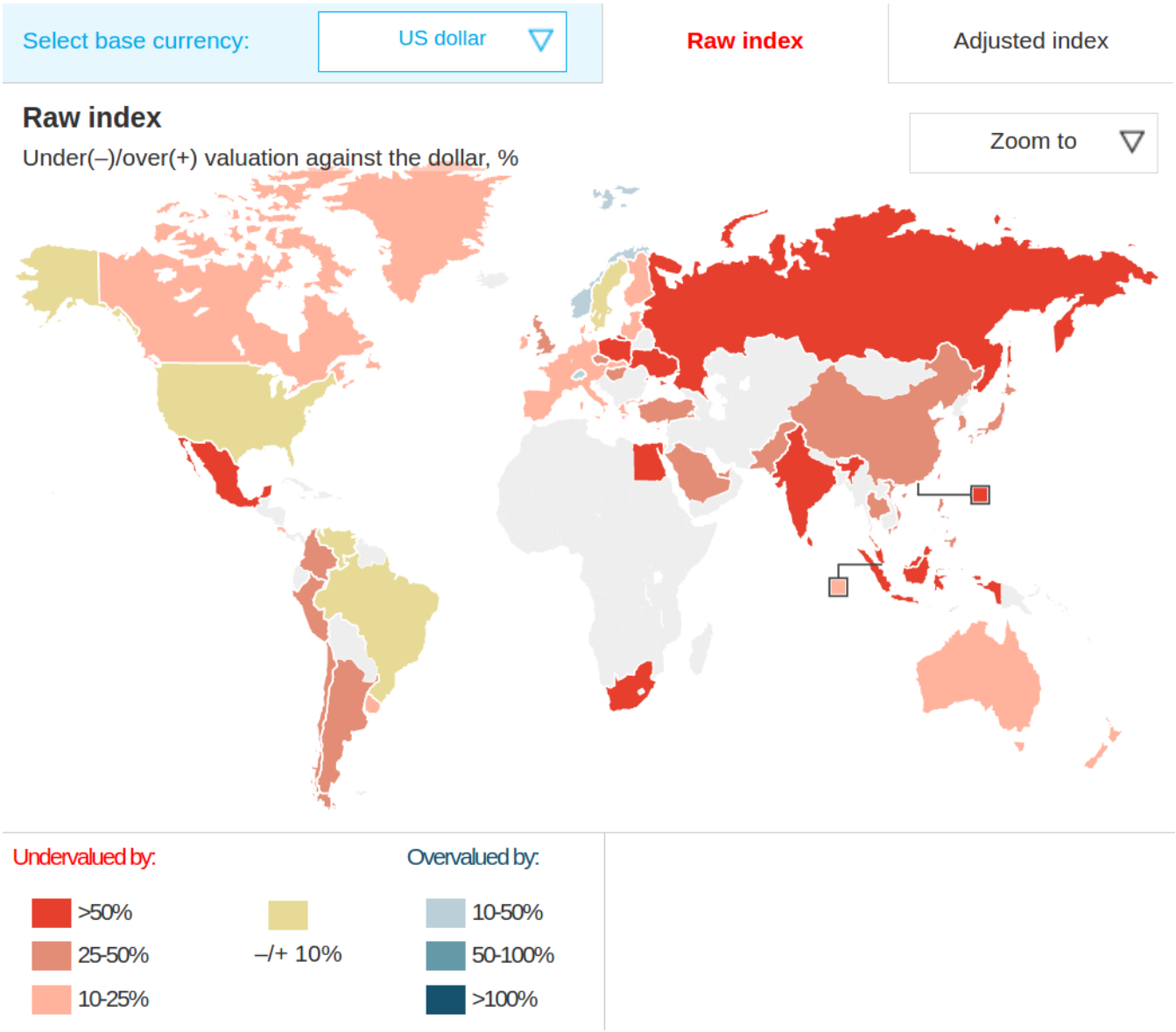
Global Competition

Purchasing Power Parity (PPP)

- Idea that a basket of common goods would have the same pricing value across countries and currencies
- Assumptions:
  - no transaction cost (currency exchange)
  - no trade barriers
- Useful to think about:
  - Arbitrage
  - Undervaluation and overvaluation due to market expectation (e.g. US President Trump & Mexico)
  - Policies such as currency fluctuation and trade barriers

Global Competition

Currency Valuation: The Big Mac Index (The Economist, 2017)



Global Competition

Tariffs

- Tax on the circulation of goods

Global Competition

Tariffs

- Tax on the circulation of goods
- See Canada's Customs Tariff (Canada Border Services Agency, 2017)

Global Competition

Tariffs

- Tax on the circulation of goods
- See Canada's Customs Tariff (Canada Border Services Agency, 2017)
- Countries set their own policies, alone or in association (e.g. NAFTA, TPP, CETA)

Global Competition

Tariffs

- Tax on the circulation of goods
- See Canada's Customs Tariff (Canada Border Services Agency, 2017)
- Countries set their own policies, alone or in association (e.g. NAFTA, TPP, CETA)
- Policies derive from political viewpoints (changing over time), for example

Global Competition

Tariffs

- Tax on the circulation of goods
- See Canada's Customs Tariff (Canada Border Services Agency, 2017)
- Countries set their own policies, alone or in association (e.g. NAFTA, TPP, CETA)
- Policies derive from political viewpoints (changing over time), for example
  - Canada cuts $48M in tariffs on food ingredients to boost manufacturing, as reported on *CBC News* (January 20, 2017)

Global Competition

Tariffs

- Tax on the circulation of goods
- See Canada's Customs Tariff (Canada Border Services Agency, 2017)
- Countries set their own policies, alone or in association (e.g. NAFTA, TPP, CETA)
- Policies derive from political viewpoints (changing over time), for example
  - Canada cuts $48M in tariffs on food ingredients to boost manufacturing, as reported on *CBC News* (January 20, 2017)
  - US President Trumps threatened German car makers with a 35% import tariff, as reported on the *Financial Post* (January 16, 2017)

Global Competition

Tariffs

- Tax on the circulation of goods
- See Canada's Customs Tariff (Canada Border Services Agency, 2017)
- Countries set their own policies, alone or in association (e.g. NAFTA, TPP, CETA)
- Policies derive from political viewpoints (changing over time), for example
  - Canada cuts $48M in tariffs on food ingredients to boost manufacturing, as reported on *CBC News* (January 20, 2017)
  - US President Trumps threatened German car makers with a 35% import tariff, as reported on the *Financial Post* (January 16, 2017)
- Impact on the Manufacturing Industry, Global Supply Chain, Foreign Investment, Currencies...

Global Competition

Non-Tariff Barriers

- Circulation of staff delivering services across borders

Global Competition

Non-Tariff Barriers

- Circulation of staff delivering services across borders
- Licensing (e.g. cryptography)

Global Competition

Non-Tariff Barriers

- Circulation of staff delivering services across borders
- Licensing (e.g. cryptography)
- Standards

Global Competition

Non-Tariff Barriers

- Circulation of staff delivering services across borders
- Licensing (e.g. cryptography)
- Standards
- Quotas

Global Competition

Non-Tariff Barriers

- Circulation of staff delivering services across borders
- Licensing (e.g. cryptography)
- Standards
- Quotas
- Labeling and Packaging conditions

Global Competition

Non-Tariff Barriers

- Circulation of staff delivering services across borders
- Licensing (e.g. cryptography)
- Standards
- Quotas
- Labeling and Packaging conditions
- Intellectual Property Laws (e.g. patents)

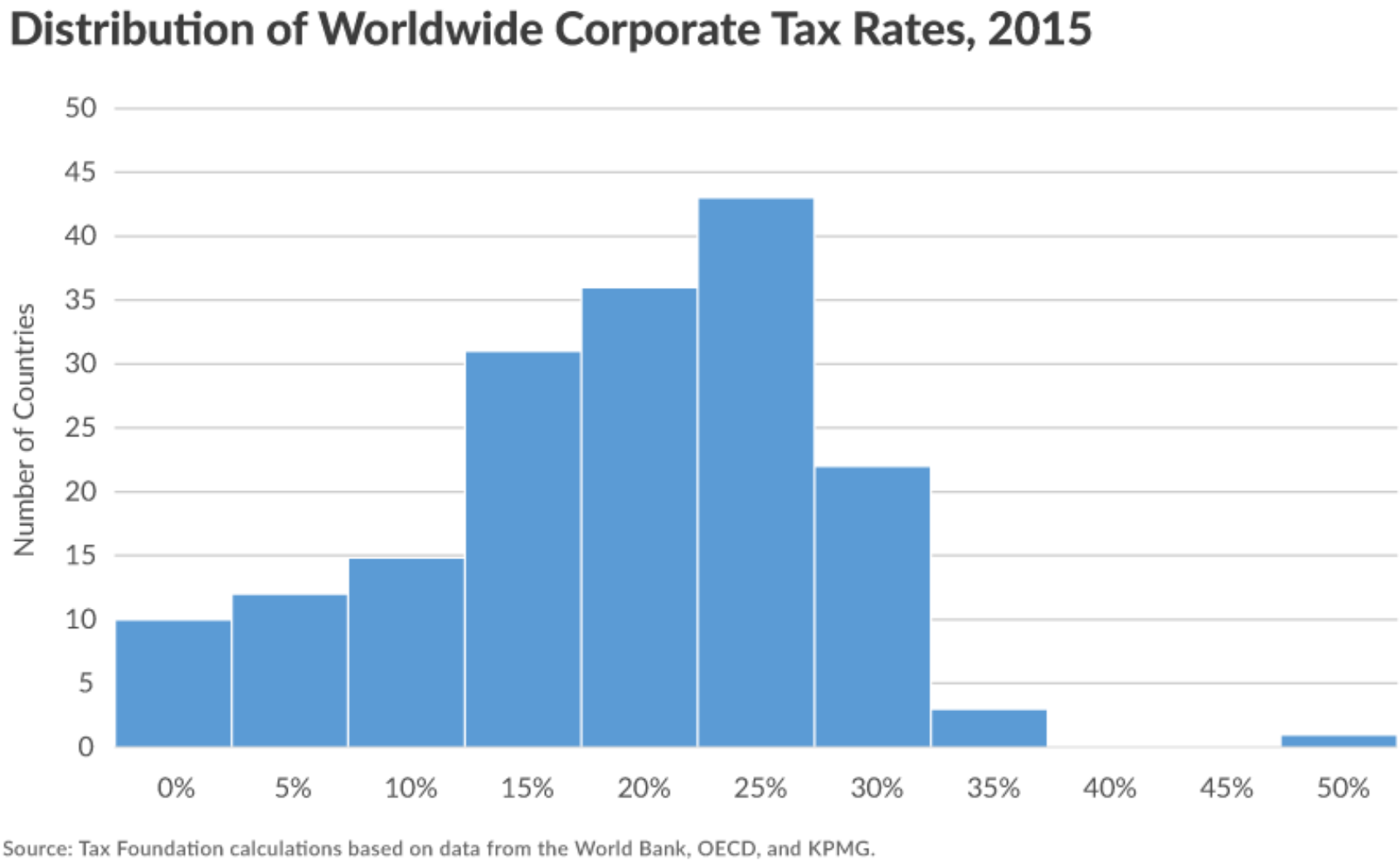
Global Competition

Non-Tariff Barriers

- Circulation of staff delivering services across borders
- Licensing (e.g. cryptography)
- Standards
- Quotas
- Labeling and Packaging conditions
- Intellectual Property Laws (e.g. patents)
- Protected Designation of Origin (e.g. Champagne, Feta cheese)

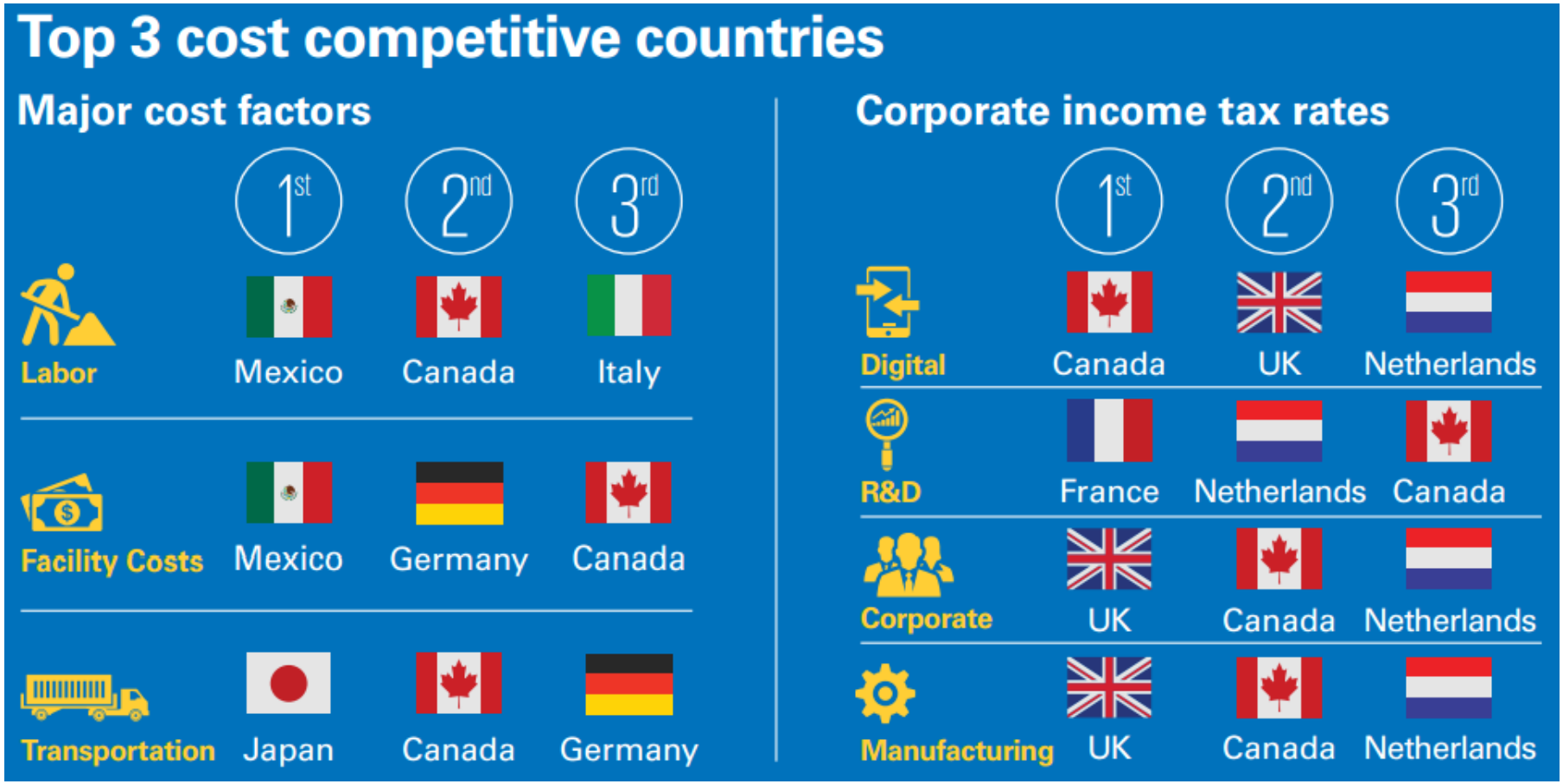
Global Competition

Corporate Tax Rates



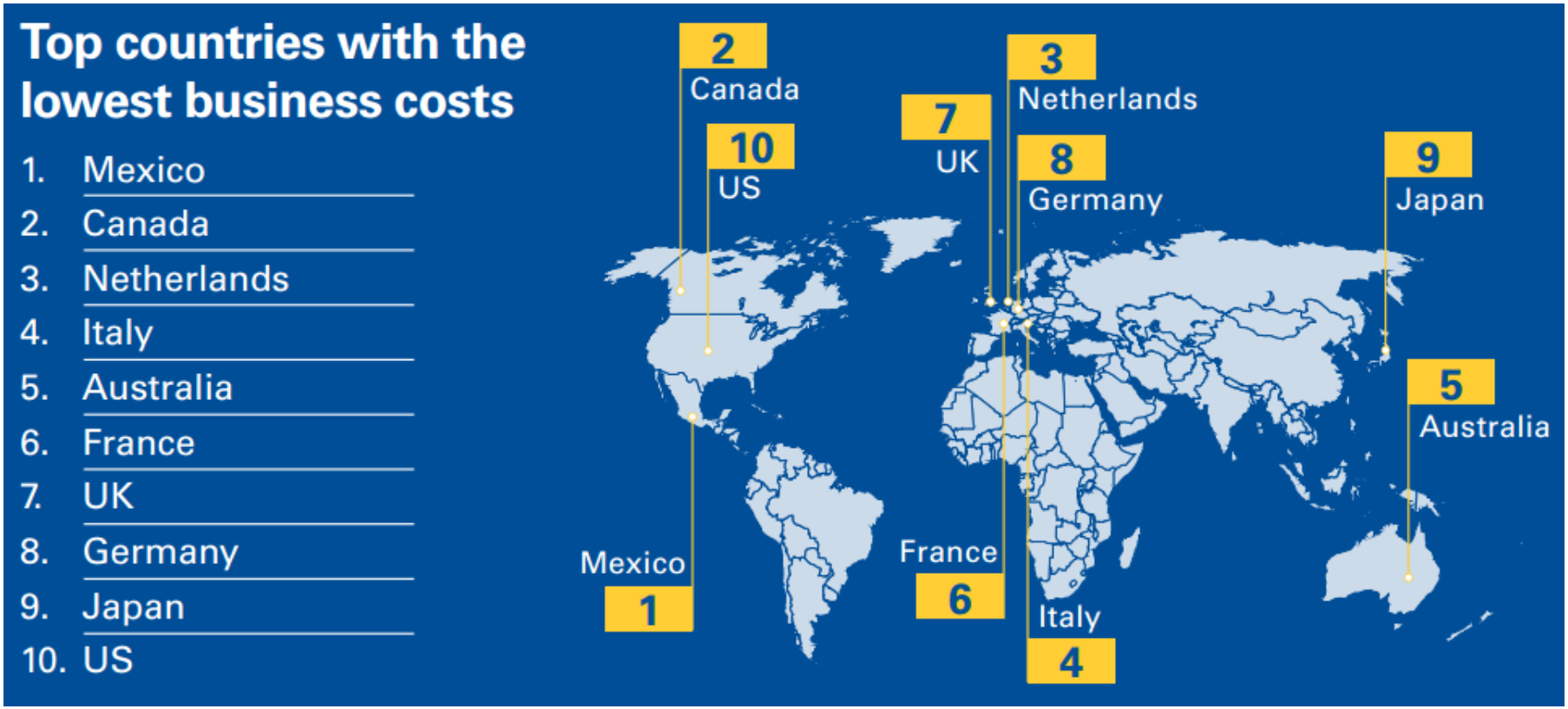
Global Competition

Corporate Income Tax Rates Competition (G7 Top 3 from KPMG, 2016)



Global Competition

Countries with the lowest business cost (G7 Top 10 from KPMG, 2016)



Global Competition

Business Cost Advantage: Greater Toronto vs USA (York Region, 2016)

- 13% lower Corporate Income Tax than U.S. avg. (State-Fed)
- 50% lower employer healthcare costs
- Lower salary costs (up to 45% in Tech) and employee attrition in tech/management roles than in the U.S.
- 40%-60% R&D cost reduction via SR&ED incentive program

Metro Toronto business cost advantage vs. U.S. metro areas

Source: KPMG Competitive Alternatives 2014

Metro Area	Software Development / Digital Media	Electronic Systems Development & Testing	Financial Services
vs. New York City	21%	19%	20%
vs. Los Angeles	17%	15%	10%
vs. San Francisco	21%	21%	18%
vs. Denver	12%	7%	2%
vs. Chicago	16%	11%	9%
vs. Boston	18%	13%	14%
vs. Raleigh	11%	14%	-1%
vs. Austin	11%	7%	1%
vs. U.S. average	16%	13%	10%

Global Competition

Manufacturing Competitiveness (Global Top 10 from Deloitte, 2016)

2016 (Current)		
Rank	Country	Index score (100=High) (10 = Low)
1	China	100.0
2	United States	99.5
3	Germany	93.9
4	Japan	80.4
5	South Korea	76.7
6	United Kingdom	75.8
7	Taiwan	72.9
8	Mexico	69.5
9	Canada	68.7
10	Singapore	68.4

Global Competition

Global Competitiveness Index (Global Top 15 from World Economic Forum, 2016)

	Economy	Score[1]	Prev.[2]	Trend[3]
1	Switzerland	5.76	1	
2	Singapore	5.68	2	
3	United States	5.61	3	
4	Germany	5.53	5	
5	Netherlands	5.50	8	
6	Japan	5.47	6	
7	Hong Kong SAR	5.46	7	
8	Finland	5.45	4	
9	Sweden	5.43	10	
10	United Kingdom	5.43	9	
11	Norway	5.41	11	
12	Denmark	5.33	13	
13	Canada	5.31	15	
14	Qatar	5.30	16	
15	Taiwan, China	5.28	14	

Global Competition

VC Money Invested in North American Cities (Thomson Reuters, 2016)

Metro Region	First 3Q 2016 Rank	2015 Rank	First 3Q 2016 VC Invested (CAD $ Millions)	First 3Q 2016 North American Market Share	Change in Rank From 2015
San Francisco	1	1	$15,981	28.3%	-
San Jose	2	2	$7,829	13.9%	-
New York City	3	3	$6,126	10.9%	-
Boston	4	4	$5,280	9.4%	-
Los Angeles	5	5	$3,267	5.8%	-
Washington D.C.	6	6	$1,154	2.0%	-
San Diego	7	8	$1,066	1.9%	+1 ▲
Chicago	8	9	$915	1.6%	+1 ▲
Orange County	9	10	$888	1.6%	+1 ▲
Seattle	10	7	$795	1.4%	-3 ▼
Montreal	11	16	$736	1.3%	+5 ▲
Philadelphia	12	15	$651	1.2%	+3 ▲
Toronto	13	14	$645	1.1%	+1 ▲
Austin	14	13	$552	1.0%	-1 ▼
Houston	15	24	$523	0.9%	+9 ▲
Vancouver	20	19	$315	0.6%	+1 ▲
Kitchener-Waterloo	21	26	$295	0.5%	+5 ▲

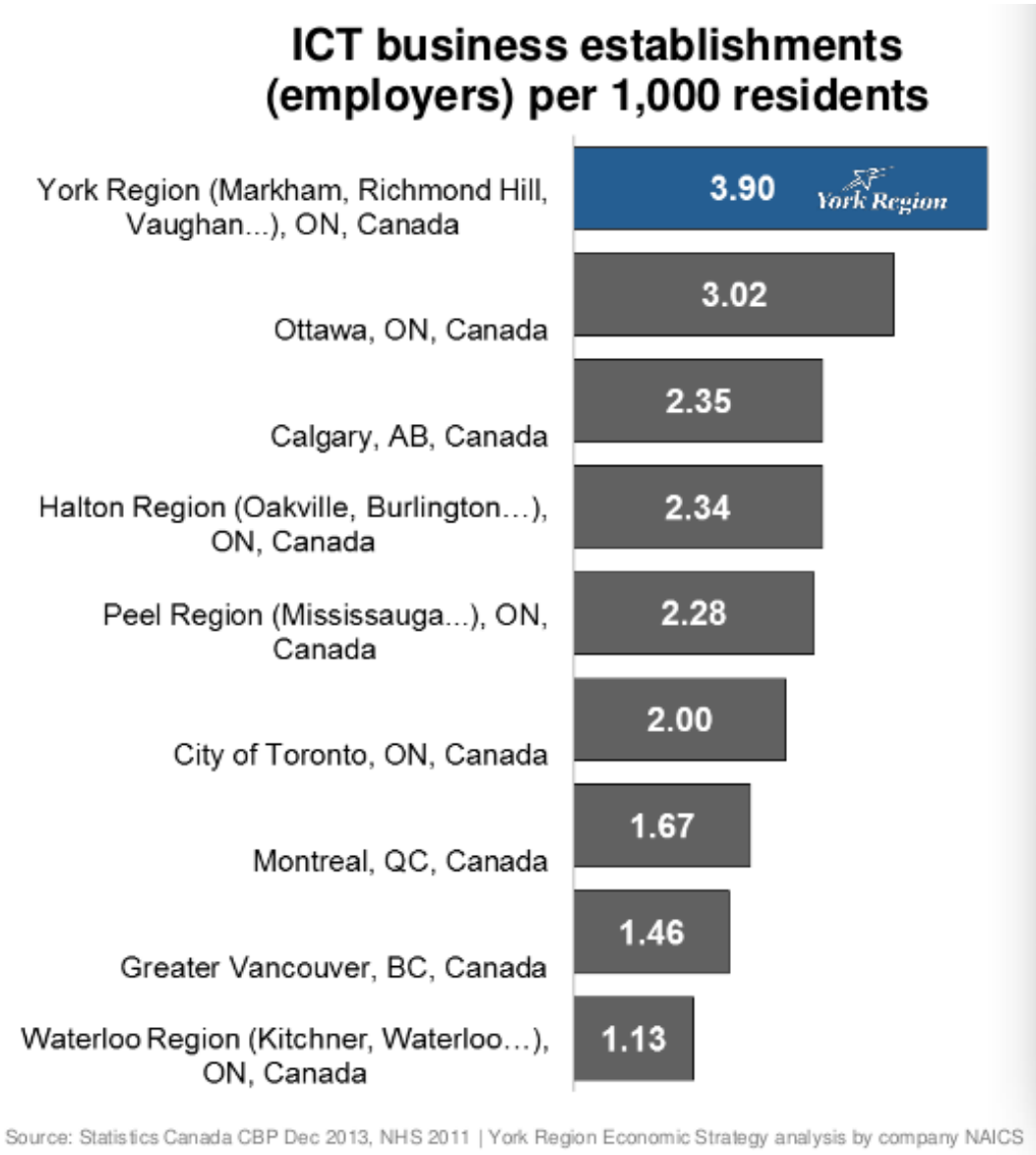
THOMSON REUTERS

11

Contributors to our venture capital & private equity analyses are entitled to packages of additional data. Please contact us to participate.

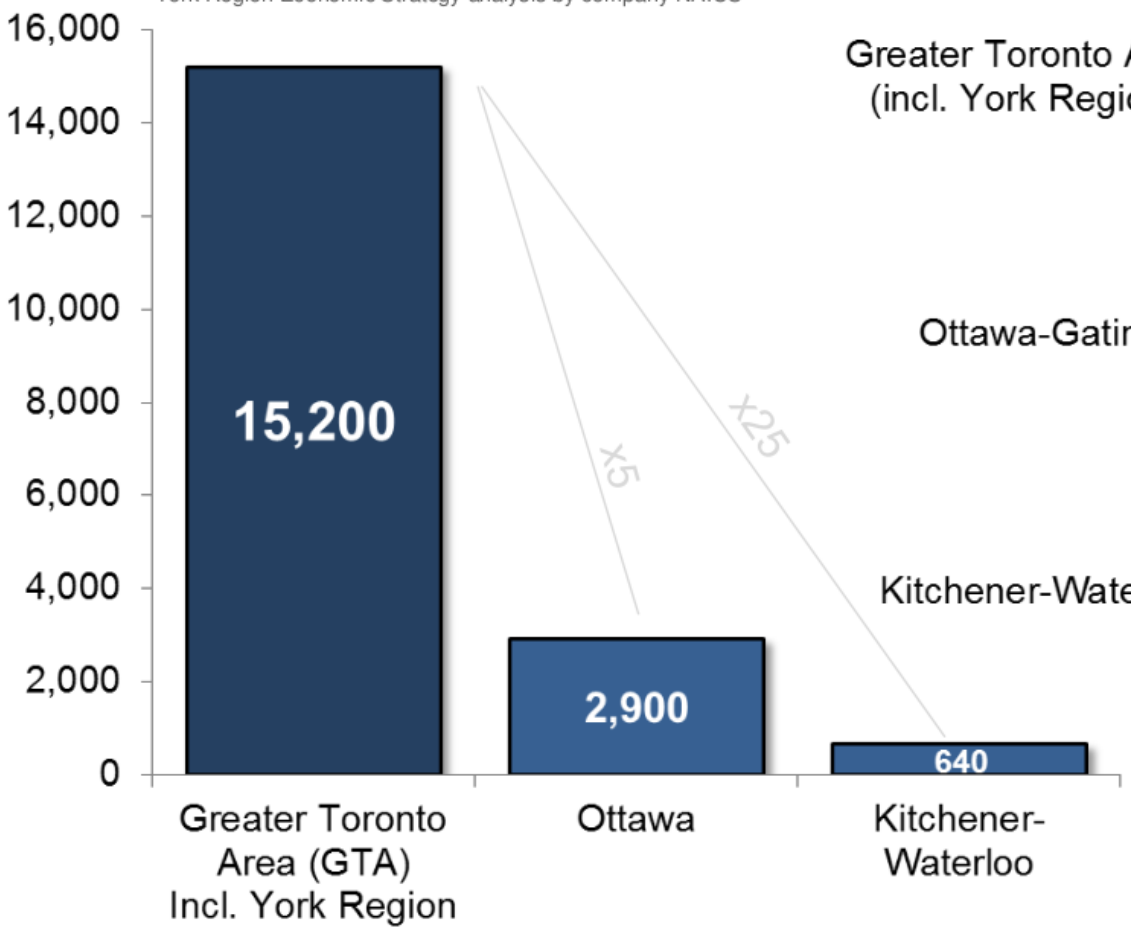
Global Competition

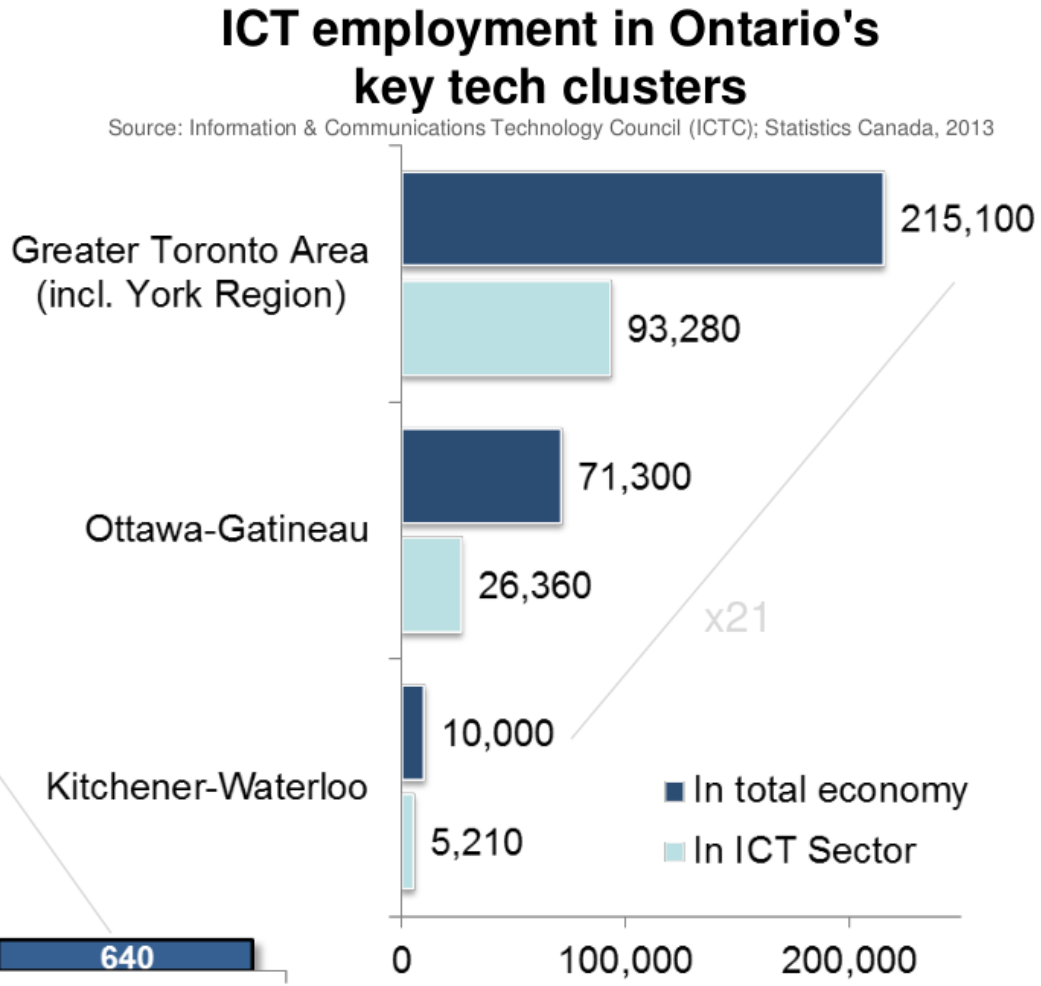
ICT Cluster Density across Canada (York Region, 2016)



Global Competition

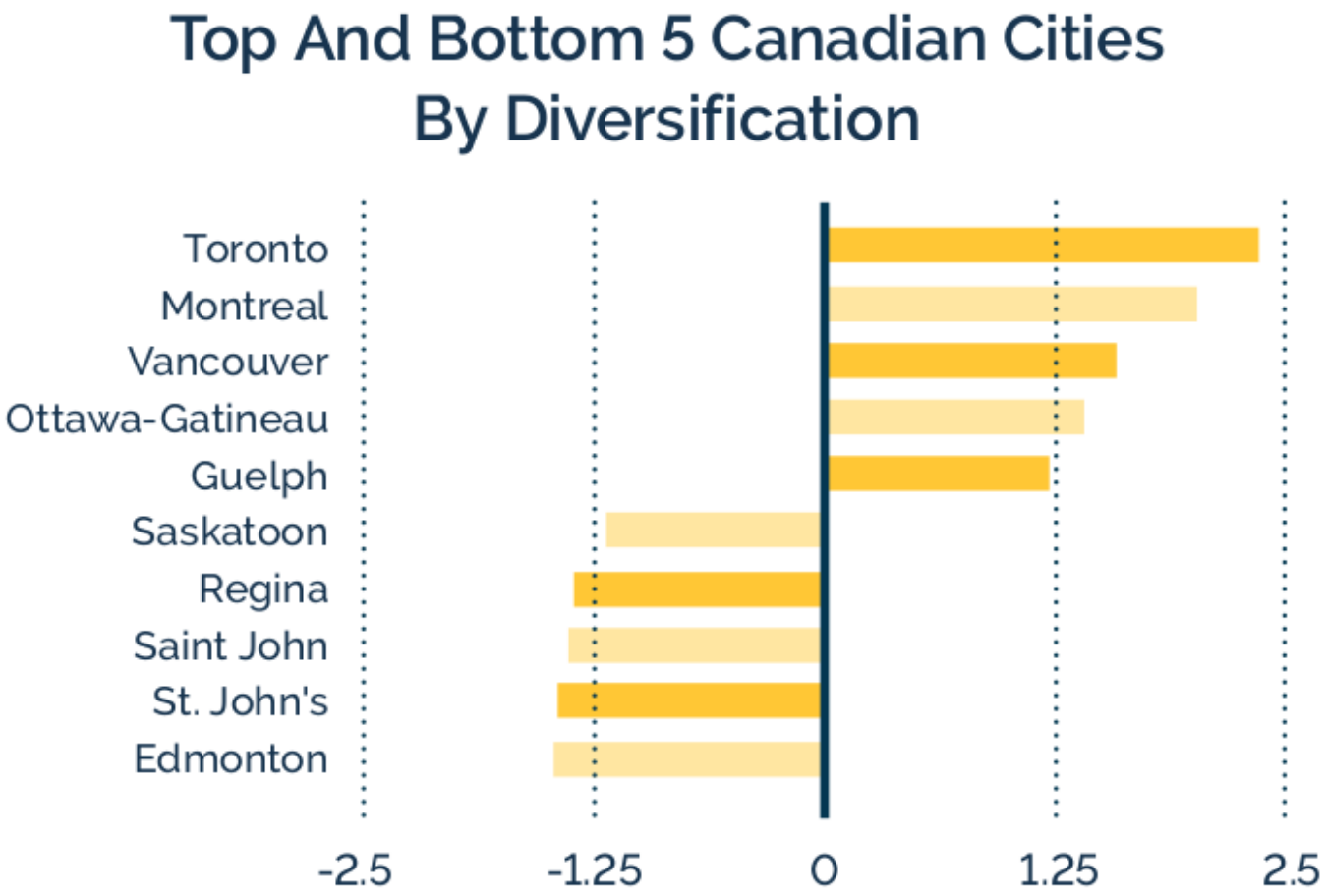
Size of Ontario ICT Clusters (York Region, 2016)





Global Competition

Top Canadian Cities by Diversification (TechTO, 2016)



Global Competition – References I

Canada Border Services Agency. (2017, January). Customs Tariff 2017. Retrieved from http://www.cbsa-asfc.gc.ca/trade-commerce/tariff-tarif/2017/menu-eng.html

Deloitte. (2016). 2016 Global Manufacturing Competitiveness Index. Retrieved from https://www2.deloitte.com/global/en/pages/manufacturing/articles/global-manufacturing-competitiveness-index.html

KPMG. (2016). Canada ranks 1st in G7 for corporate tax rate and cost competitiveness. Retrieved from http://www.investinontario.com/spotlights/canada-ranks-1st-g7-corporate-tax-rate-and-cost-competitiveness

McGregor, J. (2017, January). Canada cuts $48M in tariffs to boost food manufacturing. *CBC News*. Retrieved from http://www.cbc.ca/news/politics/food-ingredient-tariff-cuts-1.3942870

Global Competition – References II

Reuters. (2017, January). Trump threatens German carmakers with 35% U.S. import tariff. *Financial Post*. Retrieved from http://business.financialpost.com/news/transportation/trump-threatens-german-carmakers-with-35-u-s-import-tariff

TechTO. (2016). How Technology Is Changing Toronto Employment. Retrieved from https://techtoronto.org/Report2016/

The Economist. (2017, January). The big mac index. Retrieved from http://www.economist.com/content/big-mac-index

World Economic Forum. (2016). Global Competitiveness Report 2015–2016. Retrieved from http://www3.weforum.org/docs/gcr/2015-2016/Global_Competitiveness_Report_2015-2016.pdf