

What We Do at Work

Niall O'Connell
Executive Director of Engineering

Openet
Dublin



March 3rd 2015

UCD Computer Science Presentation




AGENDA

- Who we are, what we do..
- How we build, from code to shipment
- Builds, FOSS and Export Controls
- Q/A

Openet – Telecoms ISV since 1999

- Openet builds Network Grade Software for Telecoms Operators worldwide
 - Openet is based in Dublin, with offices in USA and Malaysia
 - We hire graduates every year 😊
- Openet.com/careers





Building Products

- Many different product lines, all building/delivering in parallel
- We use an Agile development methodology
- Continuous build and automated testing
- Solutions are the new 'Builds'
- Tools
 - Desktop Linux environments – Eclipse, local unit, automated test.
 - NMake – Makefiles are blueprint of how you build code
 - Source Control – SVN, Git/Stash
 - Native Compilers – Intel ICC, Oracle Java Compiler
 - Jenkins – Continuous Integration 'controller/scheduler'
 - Automated Test Suite, daily and nightly test profiles – Selenium, JMeter, Custom Test Automation suites
 - Maven, Artifactory repositories for 3rd Party s/w

Openet builds different products with lots of parallel activity.

Agile development methodology enables us to prioritise customer needs in a more efficient way

Automation is a must-have, for a successful Agile development culture/approach
Solutions are where you modularise your software offering leading to agility, where you can build new or modify existing components with little overhead of testing/building a large build. Linux Kernel and Packages/RPMs are a good example of this.
Tools used in a typical modern agile software development environment



Builds are becoming Solutions

- Big single build vs individual components
 - Modularisation and Components rather than build single builds
 - Release management is changing
 - Build Builds are moving to individual components; each are versioned
 - More scalable
 - Component Based Software Engineering – needs newer infrastructure
 - OSGi is an example of how you can deploy components in the field
 - Enables faster release time to customers
 - Telco customers could wait ~2 years after initial GA release due to large scale testing
 - Customers want more Agility

Big Monolithic builds are hard to test in a continuous build, unless you have a fully automated test suite

Modularisation enables you to break up big builds into set of independent components; all which can change independently and maintaining the APIs they expose. Linux Kernel and RPM Packages are good example.

To be able to offer customers modular builds, ISVs will have support an infrastructure to manage builds made up of components rather than large builds. YUM Repository is a good example of this in the Linux distribution.

OSGi is an example of a Java Framework which enables modules to be deployed in such a manner using individual components with different life cycles

Operators in the Telecoms worlds sometimes have a lot of testing to do before they bring a product to market, as lots of vendors, inter-operability testing needs to take place

Software Customers want to be able to request just the changes they require, rather than having to test everything regardless of whether they have changed it or need features updated in that area.



What is 3rd Party Software

- Third Party Software is any software that is publicly available or offered commercially.
- Just because it 'publicly available or is commercial' does not mean it is correct for a individual or enterprise to use
- Free Open Source Software (FOSS) may NOT mean it is 'free':
 - Free to re-use with no support
 - Free to re-use with no license to use generate code
 - Free to re-use for personal use
 - License would cause severe IP problems
- All software used in Design, Building, Testing and Run-time is subject to such criteria.

When building software, you need to be aware of 3rd Party software and it's copyright.



What is Export Control

- Export controls comprise of:
 - European Union regulations/directives for Union General Export Authorisations (UGEAs)
 - Irish legislation – Control of Exports Act 2008.
- Export Control is designed to control initially items such as military use or items that control exports of dual-use goods (i.e. goods with both a military and civilian use).
- This means that software with **encryption** capabilities are deemed to be of 'dual-use'.
- Reference: Annex 1 to the Council Regulation (EC) No 428/2009:
<http://ec.europa.eu/transparency/regdoc/rep/3/2014/EN/3-2014-7567-EN-F1-1-ANNEX-1.Pdf>

If you build software for export, you need to be aware of the export laws...

Thank You

