

Open Source  
in  
Visma Software R&D  
Bent Foyn, Dag Lorås, Dalip Dewan

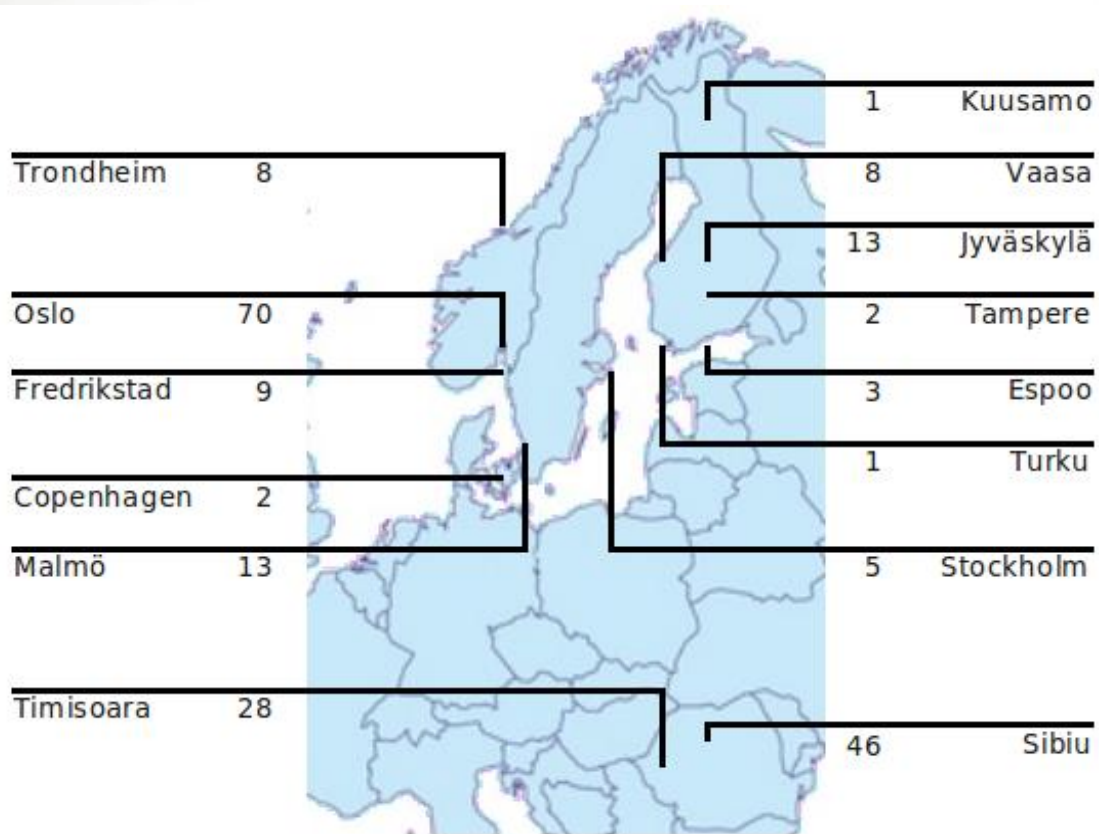
# Open Source in Visma Software R&D

1. Creating an Open Source culture  
– Dalip Dewan
2. Using Open Source assets as product base  
– Dag Lorås
3. Fueled by Open Source  
– Bent Foyn

Closed Source

Strong “local” cultures (Closed cooperation mushroom culture)

Proprietary Tools and Infrastructure



# What does the Open Source Culture Contribute?

- Process Improvements (How to build software)
- Infrastructure and Tools to Support Software Processes
- Actual Software Components and Platforms for Use

# Culture is OSS' Contribution!

- Open code, communication, information, interactions
- Loose Couplings and therefore Reliability, Trust
  - No control of the input or output chain
- Do one thing well
- Meritocracy
- Process facisim
- Learning Organizations
- Attitude of contribution and selv driven individuals
- Community creation

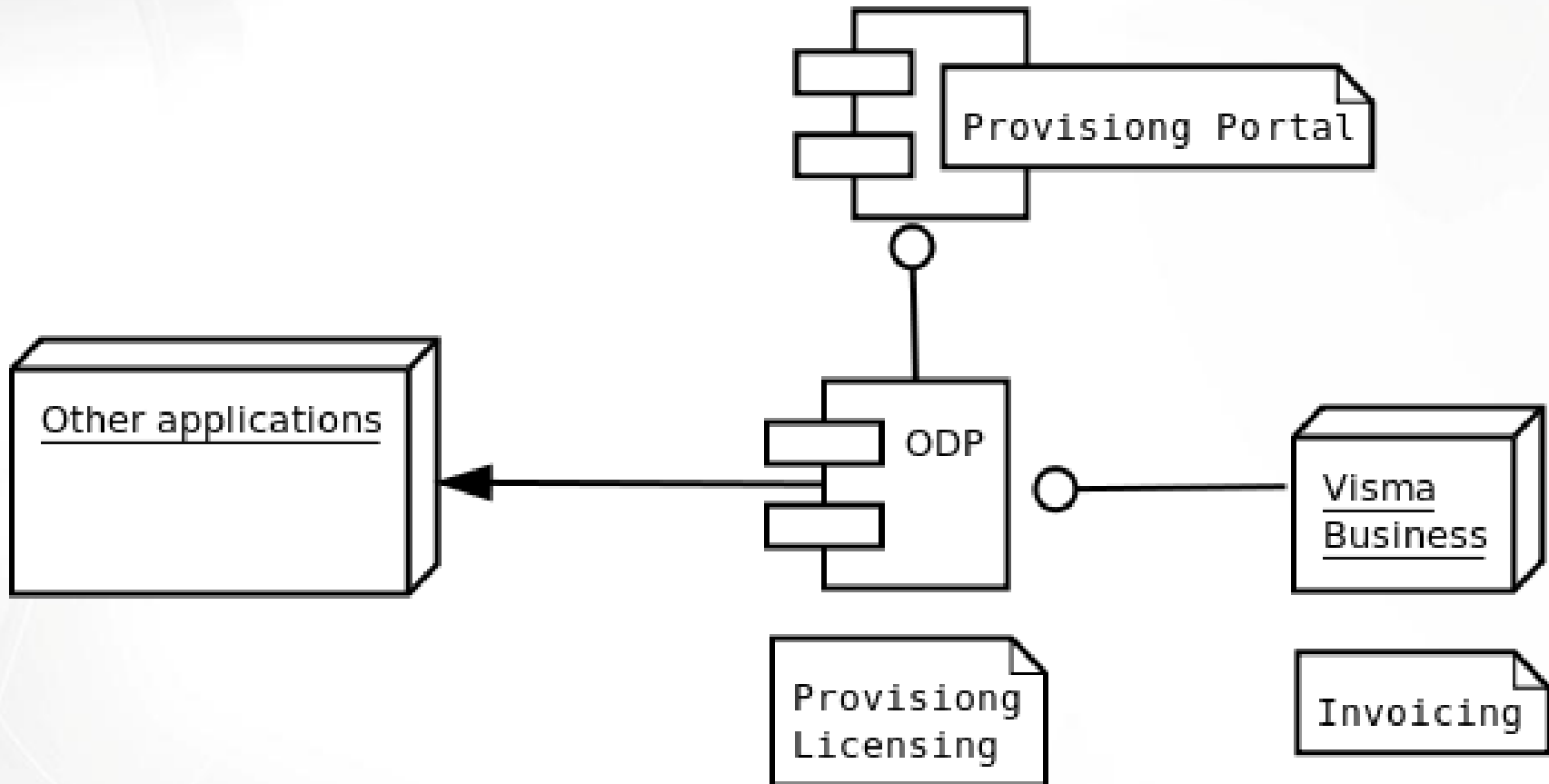
# Using Open Souce assets as product base

Dag Lorås

# Example of OSS use

- Build service
- No external distribution of software
- A web based provisioning system
- Handle license, user information, invoice etc.

# Architecture



# Components

- Liferay portal server
- Apache web server
- Glassfish
- MySQL database
- JQuery, IceFaces
- Drools rule engine, Log4j, slf4j, openshh
- Debian Linux
- Junit, JunitEE, Jmock, Findbugs, SourceMonitor, PMD, Jmeter, Selenium, Hudson, Netbeans, Eclipse, Nagios

# Benefits

- Flexibility
- Components selection to our needs
- Comfortable license costs
- Able to swap out components
- But, requires high competence

# **Visma R&D - Fueled by Open Source**

Bent Foyn

# Challenge

- 220 employees
- 18 locations
- Local infrastructure and tools only
- Development methodology:  
Change from Waterfall to SCRUM



## Solution – a cocktail of OS products

- Common infrastructure – everyone in the same domain
- Quality system: **TWiki**
- Info infrastructure: **TWiki**
- Issue tracking: Visma CRM -> **Bugzilla**
- Hours reporting: MS Project -> **Bugzilla**
- Forum: **phpBB**
- Automatic testing: **CruiseControl**
- Live communication
  - Skype / Microsoft Office Communicator
  - Video conference
- Visma support: **VBulletin, OTRS** (ITIL)

# Example: TWiki + Bugzilla + CruiseControl

- TWiki Project backlog with iteration status
- TWiki requirements
- CruiseControl
- Bugzilla issue search
- Bugzilla hours

Orig. Est.	Current Est.	Hours Worked	Hours Left	%Complete	Gain	Deadline
<input type="text" value="80.0"/>	105.5	105.5 + <input type="text" value="0"/>	<input type="text" value="0.0"/>	100	-25.5	<input type="text"/> (YYYY-MM-DD)
<a href="#">Summarize time (including time for issues blocking this issue)</a>						

# Example from CruiseControl.NET



[Documentation](#)

Dashboard > local > AccountingOffice

version : 1.4.2.14

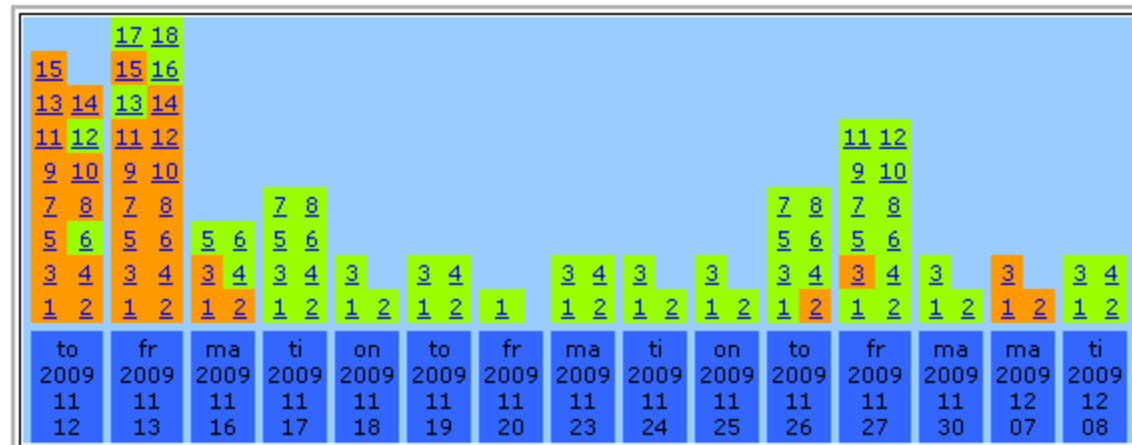
- [Project Report](#)
- [Latest Build](#)
- [View All Builds](#)
- [View Statistics](#)
- [View Server Log](#)
- [Project Configuration](#)
- [View Modification History](#)

## Project Report for AccountingOffice

Click [here](#) for the most recent build report.

### Build Overview

63%



Recent Builds
2009-12-08 08:48:07 (297)
2009-12-08 08:34:45 (296)
2009-12-08 08:31:12 (295)
2009-12-08 07:47:00 (294)
2009-12-07 15:22:00 (Failed)
2009-12-07 15:09:47 (Failed)
2009-12-07 15:03:06 (Failed)
2009-11-30 08:17:34 (293)
2009-11-30 07:40:56 (292)
2009-11-30 07:31:57 (291)
<a href="#">Show All</a>

This page rendered at: 2009-12-08 13:22:26

DEVELOPED BY  
**ThoughtWorks**



# Open Source experience so far

- TWiki works very well. Many useful plugins
- TWiki serves as an important glue between our development systems. Easy syntax and quick pick-up.
- Bugzilla well suited, but needs to be configured. Version 3.2-> has nicer GUI (dusk skin)
- Bugzilla need integration to support and hosting organizations systems
- Bugzilla need improved timesheet view for time tracking
- CruiseControl works very well – easy daily reports from automatic tests
- We may need better Scrum support – Burndown charts etc.

# Open Source maturity

1. Use it for internal processes
2. Build products on it
3. Share and give back to the community