

# Propositions

accompanying the thesis

## A Reference Architecture for Distributed Software Deployment

Sander van der Burg

1. Many of the non-functional requirements of a deployed service-oriented system can be realized by selecting an appropriate software deployment system. [this thesis]
2. The intention of the Filesystem Hierarchy Standard (FHS)<sup>1</sup> is to provide portability among Linux systems. However, due to ambiguity, over-specification, and legacy support, this standard limits adoption and innovation. This phenomenon applies to several other software standards as well.
3. To move the software engineering community as a whole forward, industry and academia should collaborate. Unfortunately, their Key Performance Indicators (KPIs) drive them apart, resulting in a prisoner's dilemma.
4. In software engineering, the use of social media, such as blogging and twitter, are an effective and efficient way to strengthen collaboration between industry and academia.
5. While the vision behind the definition of free software as described by the Free Software Foundation<sup>2</sup> to promote freedom is compelling, the actual definition is ambiguous and inadequately promoted.
6. Compared to the definition of free software provided by the Free Software Foundation, the definition of Open Source<sup>3</sup> as provided by the Open Source Initiative, fails to improve on freedom. While it has been more effectively promoted, it lacks a vision and does not solve ambiguity.
7. Just like making music is more than translating notes and rests into tones and pauses with specific durations, developing software systems is more than implementing functional requirements. In both domains, details, collaboration and listening to others are most important.
8. Multilingualism is a good quality quality for a software engineer as it raises awareness that in natural languages as well as in software languages and techniques, there are things that cannot be translated literally.
9. Stubbornness is both a positive as well as a negative trait of a researcher.

*These propositions are considered defensible and opposable, and as such have been approved by the supervisor, prof. dr. A. van Deursen.*

---

<sup>1</sup><http://www.linuxfoundation.org/collaborate/workgroups/lsb/fhs>

<sup>2</sup><http://www.gnu.org/philosophy/free-sw.html>

<sup>3</sup><http://opensource.org/osd>

# Stellingen

behorende bij het proefschrift

## A Reference Architecture for Distributed Software Deployment

Sander van der Burg

1. Veel van de niet-functionele eisen van een uitgerold service-oriented systeem kunnen gerealiseerd worden door een passend software deployment systeem te kiezen. [dit proefschrift]
2. De intentie van de Filesystem Hierarchy Standard (FHS)<sup>1</sup> is om portabiliteit te bieden tussen Linux systemen. Echter, vanwege ambigüiteit, over-specificatie, en legacy ondersteuning wordt adoptie en innovatie beperkt. Dit fenomeen is ook van toepassing op andere software standaarden.
3. Om de software engineering gemeenschap als geheel vooruit te bewegen, moeten praktijk en wetenschap samenwerken. Echter, hun Key Performance Indicators (KPIs) drijft ze uiteen, wat resulteert in een prisoner's dilemma.
4. In software engineering is het gebruik van sociale media, zoals bloggen en twitter, een effectief en efficiënt middel om de samenwerking tussen praktijk en wetenschap te versterken.
5. Hoewel de visie achter vrije software beschreven door de Free Software Foundation<sup>2</sup> om vrijheid te bevorderen overtuigend is, is de feitelijke definitie ambigu en onvoldoende gepromoot.
6. In vergelijking tot de definitie van vrije software gegeven door de Free Software Foundation, faalt de Open Source definitie gegeven door het Open Source Initiative<sup>3</sup> in het verbeteren van vrijheid. Hoewel deze effectiever wordt gepromoot, mist de definitie een visie en lost ze ambigüiteit niet op.
7. Net als het maken van muziek meer omvat dan het vertalen van noten en rusten in tonen en pauzes van specifieke duur, is het ontwikkelen van softwaresystemen meer dan het implementeren van functionele eisen. In beide domeinen zijn details, samenwerking en het luisteren naar anderen zeer belangrijk.
8. Meertaligheid is een goede kwaliteit van een software engineer, omdat het de bewustwording verbeterd van het feit dat zowel in natuurlijke talen, als in software-talen en technieken, niet alles letterlijk vertaald kan worden.
9. Koppigheid is zowel een positieve als negatieve karaktertrek van een onderzoeker.

*Deze stellingen worden opponeerbaar en verdedigbaar geacht en zijn als zodanig goedgekeurd door de promotor, prof. dr. A. van Deursen.*

---

<sup>1</sup><http://www.linuxfoundation.org/collaborate/workgroups/lsb/fhs>

<sup>2</sup><http://www.gnu.org/philosophy/free-sw.html>

<sup>3</sup><http://opensource.org/osd>