A Framework to Evaluate Open Source Software (EvalOSS)

Bhim Upadhyaya

bhim.upadhyaya@west.cmu.edu

https://www.linkedin.com/in/bhim-upadhyaya-0648a223

Integrated Innovation Institute – Center for Open Source Investigation
Carnegie Mellon University, Silicon Valley

NASA Ames Research Park, Moffett Field, CA 94035, USA
May 2, 2017

Agenda

- Background
- Technology Landscape
- Business Landscape
- Social Landscape
- Analysis
- Design
  - Architecture and Computation Design
  - Ranking Algorithm
  - Application of Ranking Algorithm
- Conclusion
- Future Work

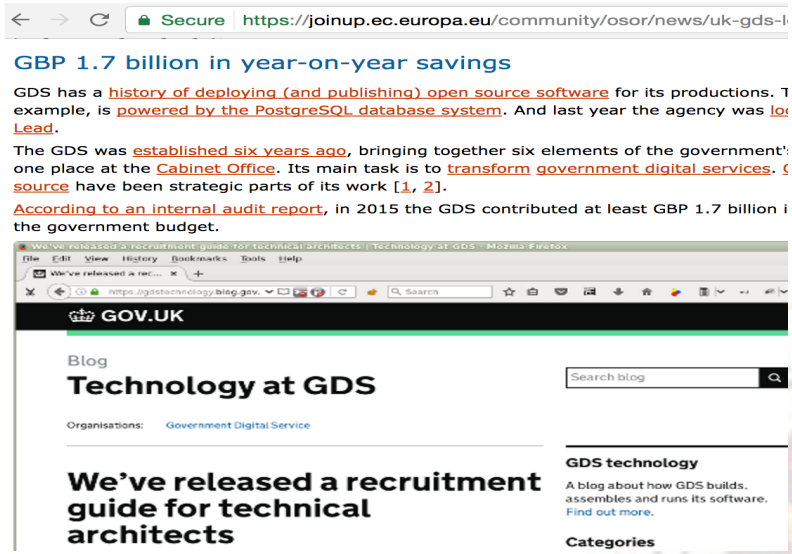
Background

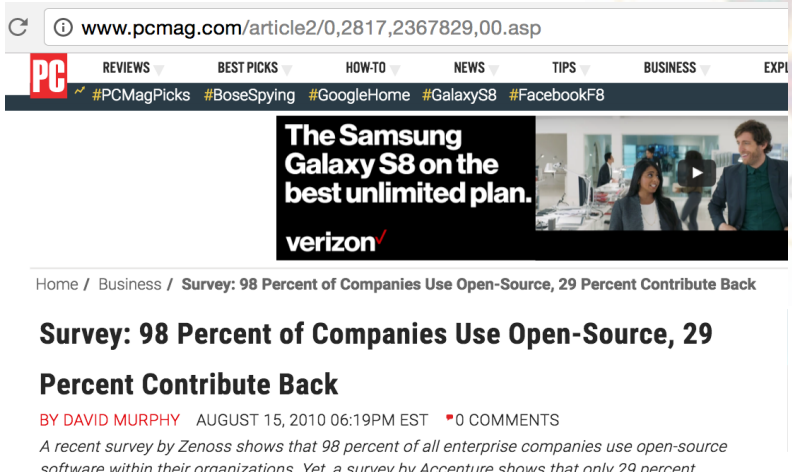
Claims

1. *OSS* is an important aspect of product suites.
2. *OSS* saves huge costs.
3. *OSS* is inevitable for organizations.

Evidence

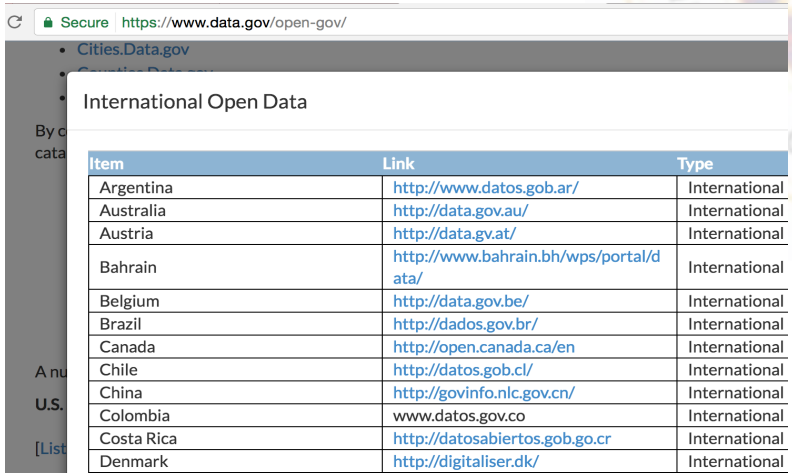
1. Earlier research work [1][2][3].
2. See UK government digital service (GDS) data.
3. See data and images.





Background contd...

OSS, Open Data, Open Government



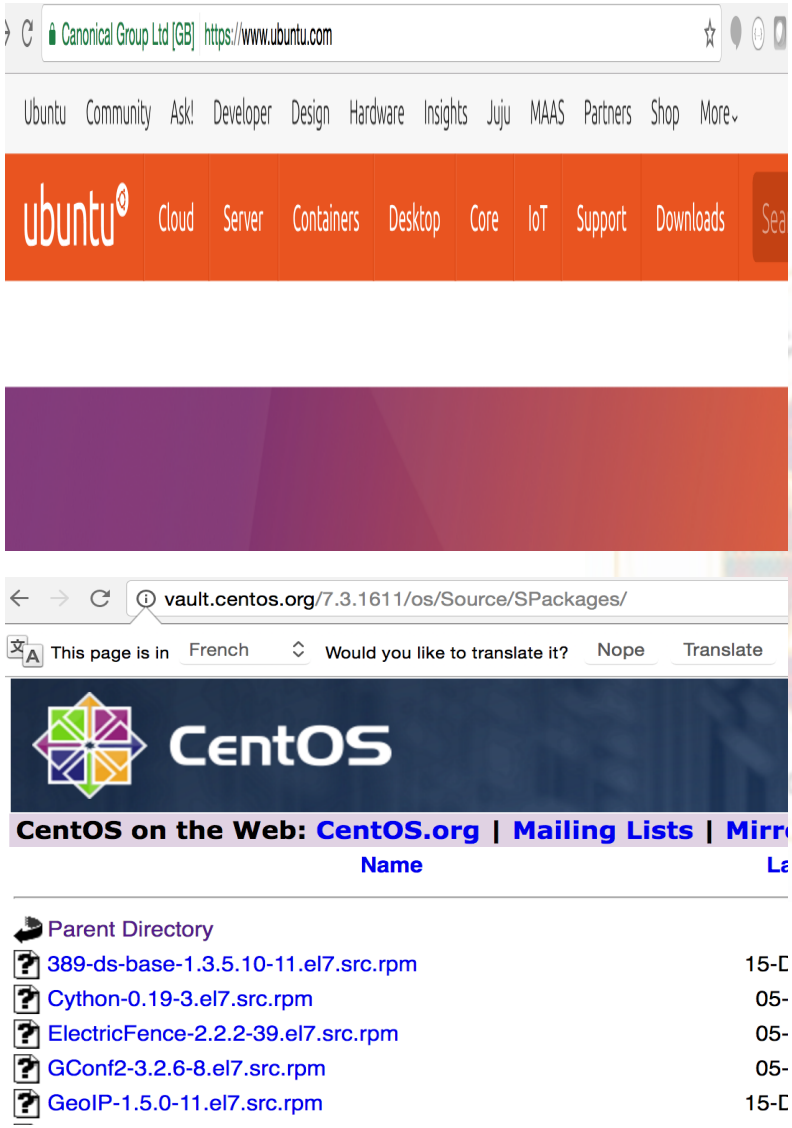
Technology Landscape

Sample Categories

1. Operating System
2. Office Automation
3. Programming Languages
4. Development Tools
5. Big Data Analysis

Technology Landscape - Operating System

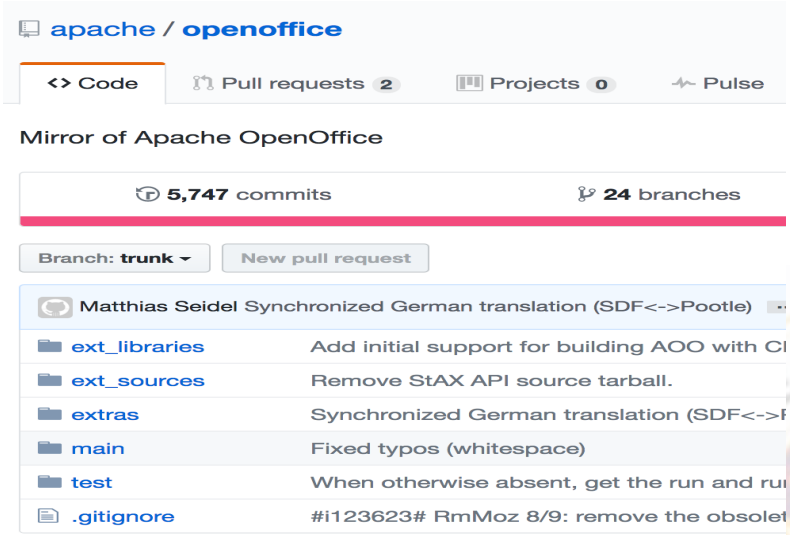
- Less friendly source code crawling
- Appreciated more for desktop
- Also used for server
- Look and feel: Mac + Windows
- Source code crawling is better
- Close links with enterprise edition
- Close links with rapdid development edition



Technology Landscape - Office Automation

OpenOffice

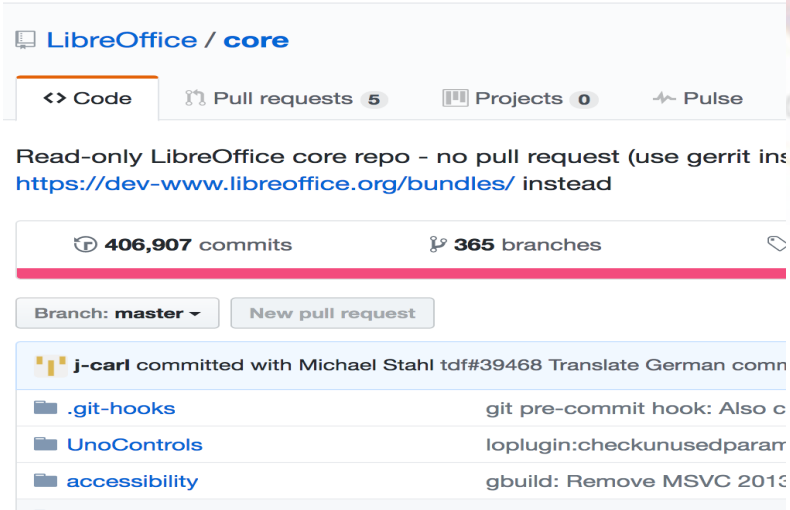
- Code structure
- Custom static code analysis



Office Automation

LibreOffice

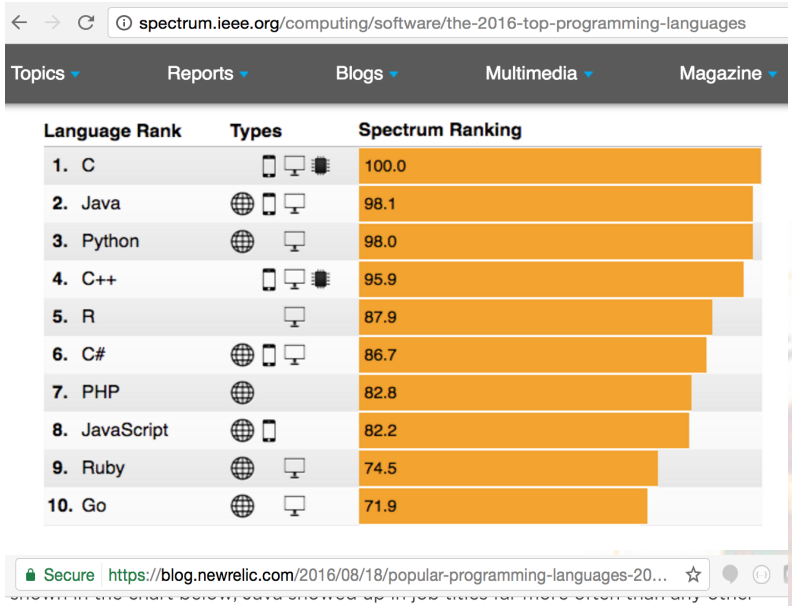
- Different from OpenOffice code structure
- Need to write custom static code analysis



Technology Landscape - Programming Languages

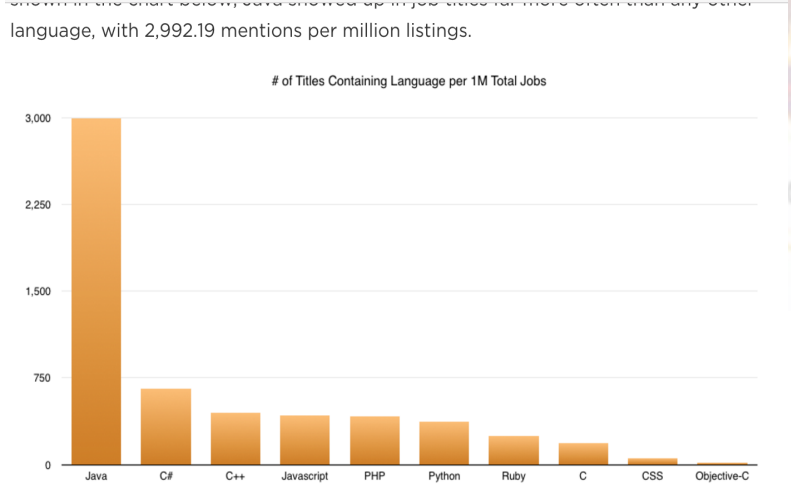
IEEE 2016 Rating

- Peer reviewed
- Scholarly
- More reliable



New Relic 2016 Rating

- Commercial company
- Focused on job listing
- Important information but need external validation



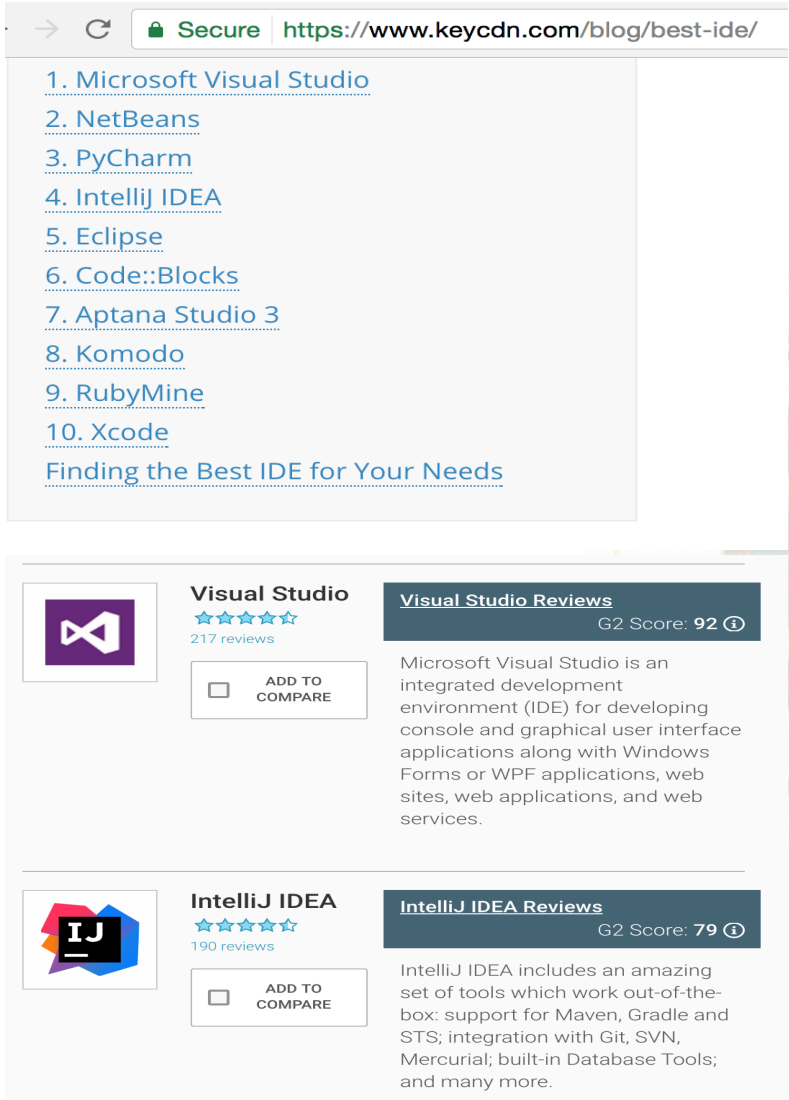
Technology Landscape - Development Tools

KeyCDN

- Blogging
- Descriptive information
- intelliJ $\mapsto$ 4, Eclipse $\mapsto$ 5

G2Crowd

- Blogging
- Rating and descriptive
- intelliJ $\mapsto$ 4+ stars, Eclipse $\mapsto$ 4 stars



Technology Landscape - Big Data Analysis

Big Data Frameworks

- Large number of choices
- Diversity in implementation technology
- Hadoop Vs. Spark

www.predictiveanalyticstoday.com/bigdata-platforms-bigdata-analytics-software/

50 Bigdata Platforms and Bigdata Analytics Software

IBM Bigdata Analytics, HP Bigdata , SAP Bigdata Analytics, Microsoft Bigdata, Oracle Bigdata Analytics, Talend Open Studio, Teradata Bigdata Analytics, SAS Big data, Dell Bigdata Analytics, HPCC System Big data, Palantir Bigdata, Pivotal Bigdata, Google BigQuery, Pentaho Big Data Analytics, Amazon Web Service, Cloudera Enterprise Bigdata, Hortonworks Data Platform, FICO Bigdata Analytics, Cisco Bigdata, Splunk Bigdata Analytics, Fusion-io Bigdata, Intel Bigdata, Mu Sigma Bigdata, MicroStrategy Bigdata , Opera Solutions Bigdata, Redhat Bigdata, Informatica Bigdata, MarkLogic Bigdata, Vmware Bigdata, Syncsort Bigdata, SGI Bigdata, MongoDB , Guavus Bigdata, Alteryx Bigdata, 1010data Advanced Analytics, Actian Analytics Platform, MapR, Tableau Software bigdata, QlikView Bigdata, Attivio's Bigdata, DataStax Bigdata, Gooddata, Google Bigdata, Datameer, CSC Big Data Platform, Flytxt, Amdocs, Cisco Bigdata, Platfora, GE Bigdata

Code Base

cloudera / **hadoop-common**
forked from ekoontz/hadoop-common

<> Code | Pull requests 2 | Projects 0 | Pulse | Graphs

Branch: **trunk** | **hadoop-common** / src / java / org / apache / hadoop / **net** /

This branch is 153 commits ahead, 11 commits behind ekoontz:trunk.

Thomas White HADOOP-6926. SocketInputStream incorrectly implements read(). Contrib.

..

CachedDNSToSwitchMapping.java	HADOOP-6668. Apply audience and stability
DNS.java	HADOOP-6668. Apply audience and stability
DNSToSwitchMapping.java	HADOOP-6668. Apply audience and stability
NetUtils.java	HADOOP-6682. NetUtils:normalizeHostName
NetworkTopology.java	HADOOP-6884. Add LOG.isDebugEnabled() g

apache / **spark**
mirrored from git://git.apache.org/spark.git

<> Code | Pull requests 478 | Projects 0 | Pulse

Branch: **master** | **spark** / core / src / main / scala / org / apache

asmith26 committed with **srowen** [MINOR] Issue: Change "slice" vs "par

..

AsyncRDDActions.scala	[SPARK-13928] Move org.apa
BinaryFileRDD.scala	[SPARK-16575][CORE] partitic
BlockRDD.scala	[SPARK-19998][BLOCK MANA
CartesianRDD.scala	[SPARK-12692][BUILD][CORE

Business Landscape

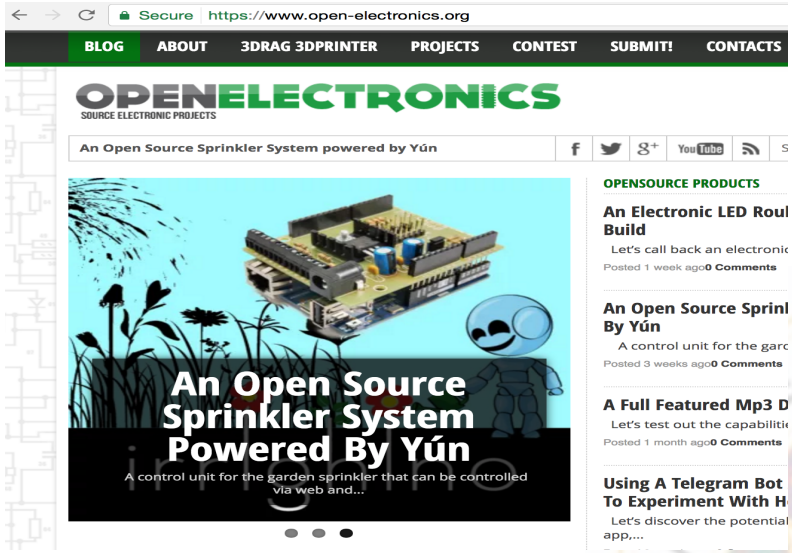
Sample Categories

1. Consumer Electronics
2. Managed Health Care
3. Retail
4. Manufacturing and Ag.
5. Online Payment

Business Landscape - Consumer Electronics

Open Eletronics

- OSS part of open electronics
- New challenge to evaluate



Consumer Electronics

OLPC

- Different problem to address
- Different strategy
- Different measure

Business Landscape - Managed Health Care

Connect

- Open source health information exchange
- Guide by HIPAA
- OSS influenced by HIPAA

Managed Health Care

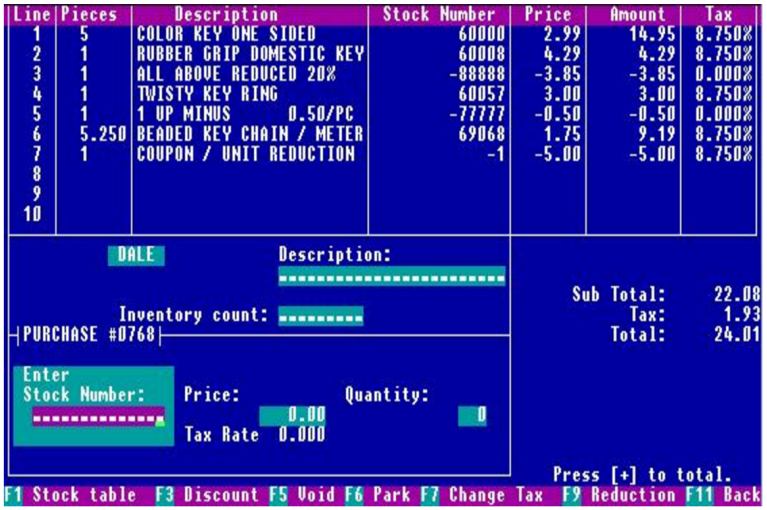
OurMed

- Over 20 categories of OSS
- Over 100 different products

Business Landscape - Retail

Keyhut

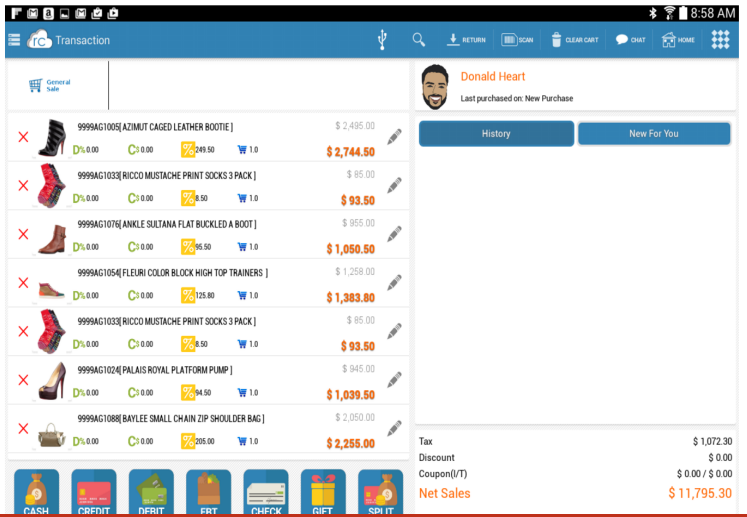
- Free POS system
- Can create multiple reports
- Can be run on a home computer too



Retail

ZeroPOS

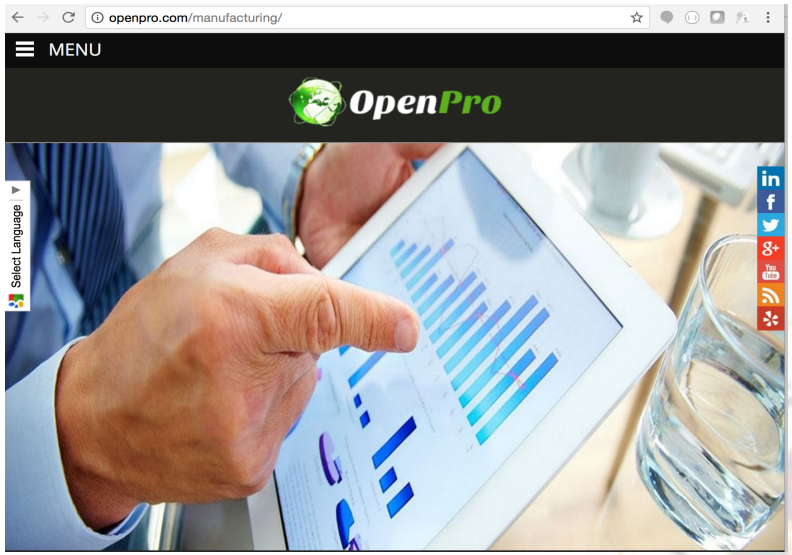
- Free cloud-based POS
- Provides unlimited stores, registers, users and products



Business Landscape - Manufacturing and Ag.

OpenPro

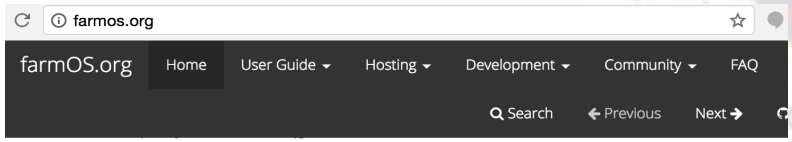
- Open source ERP system
- Modules: Manufacturing, Financials, Distribution, CRM, E-Commerce



Manufacturing and Ag.

farmOS.org

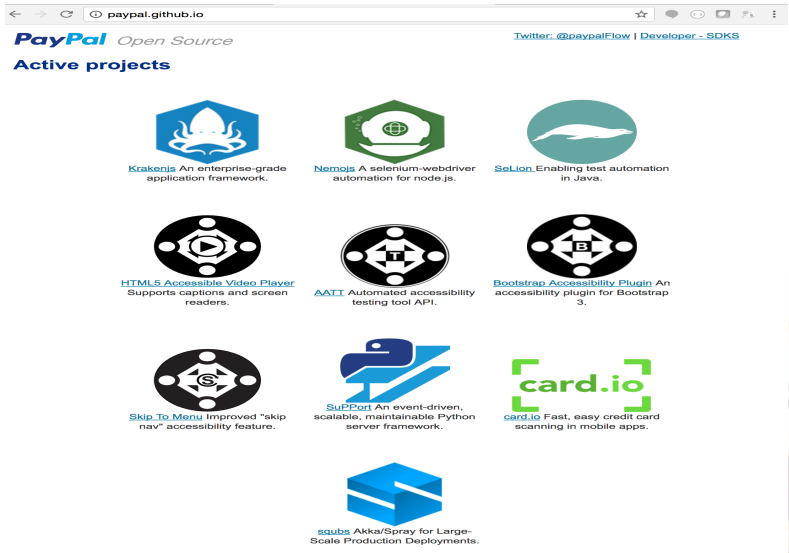
- Web application: Drupal
- Farm management, planning, and record keeping



Business Landscape - Online Payment

PayPal

- Wide variety of OSS projects
- Infrastructure is driven by OSS



Online Payment

cyclos

- Project of a network of Social TRade Organization (STRO)
- Money movement

Social Landscape

- *Social media support*: Community based development requires networking and online interactions. A society supportive of social media is critical.
- *Government support for OSS*: At least government approval is required; could be in the form of free economics.
- *Investment model*: Individual and community based investment; donation model.
- *Regulations*: Should not restrict open development; regulations supporting free economic models are helpful
- *Openness*: Openness in manufacturing devices that support OSS has positive impact; this parameter is partially controllable.

Analysis

Observations

1. Different projects have different priorities and hence the decisions.
2. Criteria differ within the same company.
3. Business have fairly stable parameters compared to projects or departments.
4. Business parameters differ from one business to another.
5. Social parameters are the most stable.

Inferences

1. There is no uniformity for priority parameters.
2. Criteria cannot be fixed.
3. Might be able to use fixed categories.
4. Business parameters cannot be fixed. Fixed categories with varying weight might increase parameter stability.
5. Might be able to treat social parameters as fixed input.

Design

Design Considerations

A Exhaustive coverage of the information aspects
   1. Expert evaluation
   2. Community opinion
   3. Documentation
   4. Code base inference

B Primarily a quantitative approach

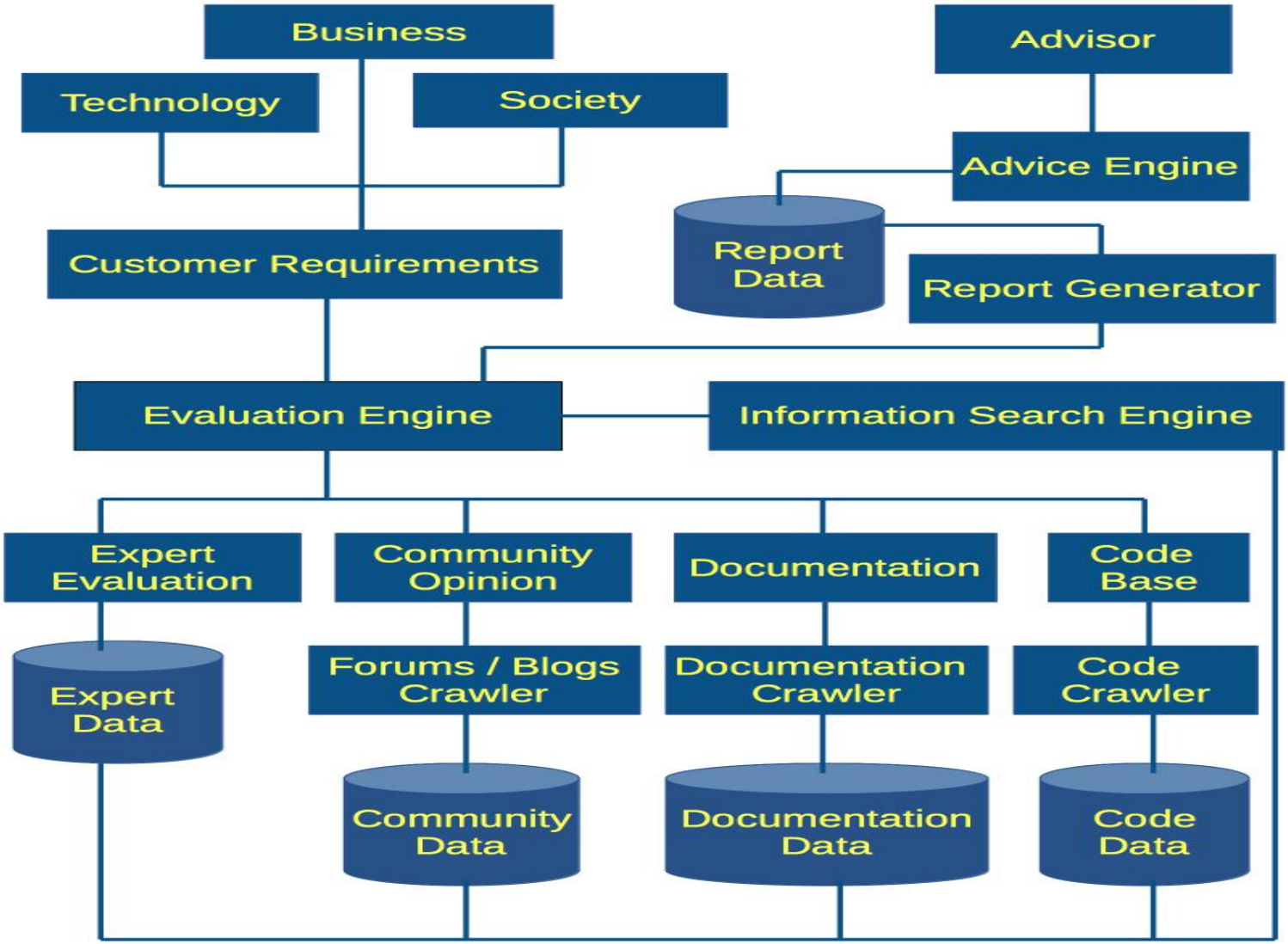
C Exhaustive coverage of requirement aspects – B, T, S

D Scalability

E Reconfiguration engines
   1. Evaluation engine
   2. Forums / Blogs crawlers
   3. Documentation crawlers
   4. Code crawlers

F Customized report generation

Design - Architecture and Computation Design



Design - Ranking Algorithm

Ranking Algorithm

1. Input value and weight for each input parameter: $R_{in} = (R_{value}, R_{weight})$
2. Populate the input set: $R = \{R_1, R_2, \ldots, R_n\}$
3. Search and fetch all potential candidate products: $P = \{P_1, P_2, \ldots, P_n\}$
4. Compute impact of each product: $I(P) = \sum_{i=1}^{k}(\sum_{j=1}^{m} S_j * w_j) * R_{(i)weight}$
5. Create an impact sequence: $I = [I(P_1), I(P_2), \ldots, I(P_n)]$
6. Create a ranking map: $Z = [1 \mapsto I(P_h), 2 \mapsto I(P_s), \ldots, n \mapsto I(P_t)]$
7. Emit a selection or a selection set

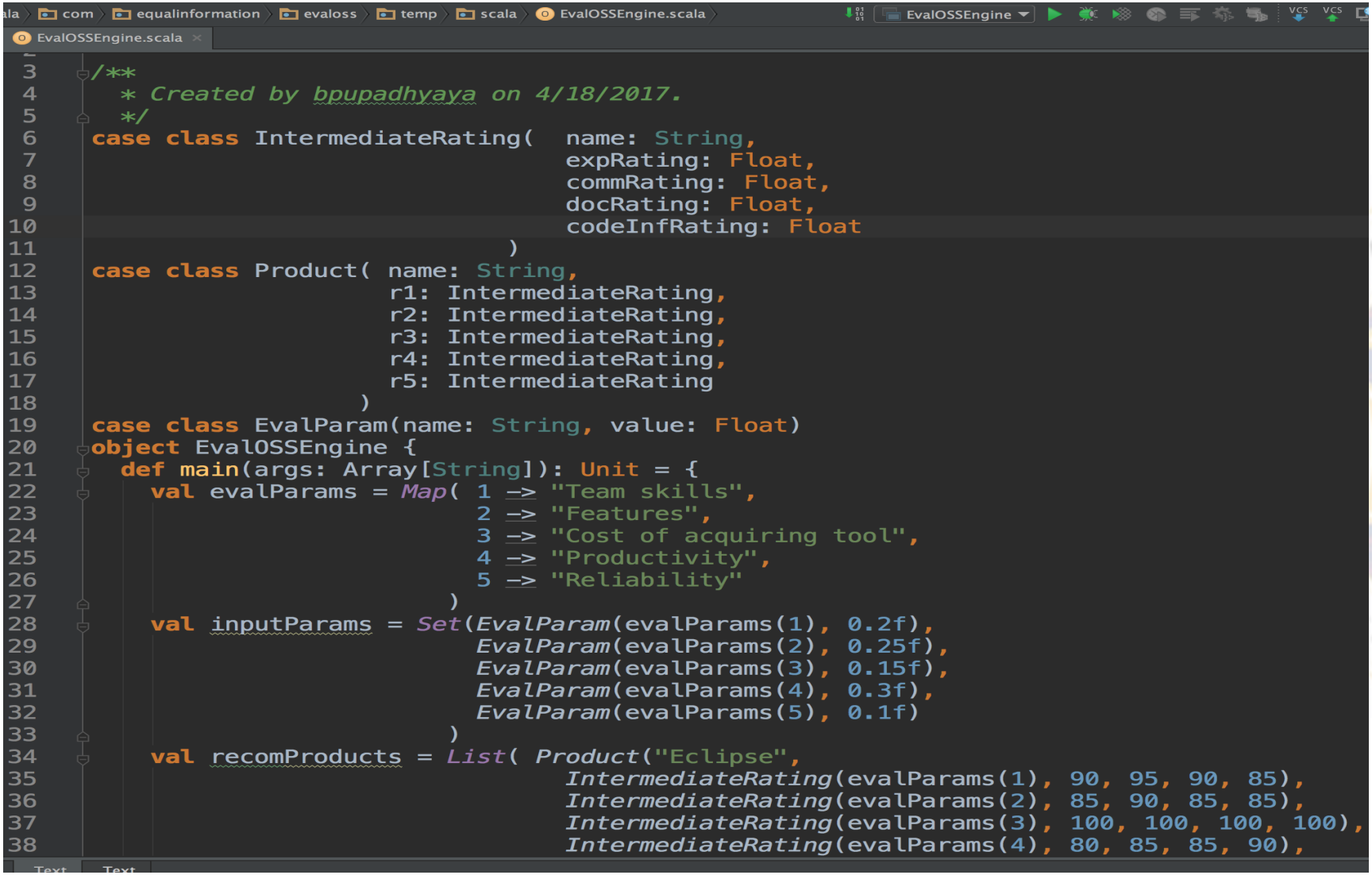
Design - Application of Ranking Algorithm

			E	C	D	B	Intermediate I(P)	Rw	I(P)
			0.35	0.1	0.15	0.4			
	R1	Team skills	90	95	90	85	88.5	0.2	17.7
	R2	Features	85	90	85	85	85.5	0.25	21.375
Eclipse	R3	Cost of acquiring tool	100	100	100	100	100	0.15	15
	R4	Productivity	80	85	85	90	85.25	0.3	25.575
	R5	Reliability	90	85	95	90	90.25	0.1	9.025
			155.75	45.5	68.25	180			88.675
	R1	Team skills	80	90	85	90	85.75	0.2	17.15
	R2	Features	95	95	100	95	95.75	0.25	23.9375
Intellij	R3	Cost of acquiring tool	90	90	95	90	90.75	0.15	13.6125
	R4	Productivity	95	100	100	95	96.25	0.3	28.875
	R5	Reliability	90	95	95	90	91.25	0.3	27.375
			157.5	47	71.25	184			110.95

Conclusion

- Observed widely varying parameters across projects, departments, and businesses.
- *Non Uniformity* $\leftrightsquigarrow$ *Dynamism*
- Covered three main requirements aspects - B, T, S
- Covered exhaustive computational aspects.
- Easy to interpretation the results.
- Complex to implement.

Future Work


com
equalinformation
evaloss
temp
scala
EvalOSSEngine.scala
EvalOSSEngine
EvalOSSEngine.scala
/**
* Created by bpupadhyaya on 4/18/2017.
*/
case class IntermediateRating(name: String,
expRating: Float,
commRating: Float,
docRating: Float,
codeInfRating: Float
)
case class Product(name: String,
r1: IntermediateRating,
r2: IntermediateRating,
r3: IntermediateRating,
r4: IntermediateRating,
r5: IntermediateRating
)
case class EvalParam(name: String, value: Float)
object EvalOSSEngine {
def main(args: Array[String]): Unit = {
val evalParams = Map(1 -> "Team skills",
2 -> "Features",
3 -> "Cost of acquiring tool",
4 -> "Productivity",
5 -> "Reliability"
)
val inputParams = Set(EvalParam(evalParams(1), 0.2f),
EvalParam(evalParams(2), 0.25f),
EvalParam(evalParams(3), 0.15f),
EvalParam(evalParams(4), 0.3f),
EvalParam(evalParams(5), 0.1f)
)
val recomProducts = List(Product("Eclipse",
IntermediateRating(evalParams(1), 90, 95, 90, 85),
IntermediateRating(evalParams(2), 85, 90, 85, 85),
IntermediateRating(evalParams(3), 100, 100, 100, 100),
IntermediateRating(evalParams(4), 80, 85, 85, 90),

References

[Wasserman et al., 2011] A. I. Wasserman, et al.
OSSpal: Finding and Evaluating Open Source Software, AFDA, 2011.

[Stol and Babar, 2010] K. J. Stol and M. Ali Babar.
A Comparision Framework for Open Source Software Evaluation Methods, IFIP Advances in Information and Communication Technology, vol 319, 2010.

OSSpal.
Available online at http://osspal.org as at April, 2017.

Center for Medicare and Medicaid Services (CMMS), *jRAVEN*.
Available online at `https://www.cms.gov/medicare/quality-initiatives-patient-assessment-instruments/nursinghomequalityinits/nhqimds30technicalinformation.html` as at April, 2017.

Microsoft.
Visual Studio Community Edition, Available online at `https://www.visualstudio.com/vs/community` as at April, 2017.

Eclipse Foundation, `https://eclipse.org/org/` as at March, 2017.

Red Hat Community, `http://community.redhat.com/` as at April, 2017.

CentOS, `https://www.centos.org/` as at Feb, 2017.

GitHub, `https://github.com/` as at April, 2017.

References contd...

OpenBRR.

Business Readiness Rating for Open Source. Available online at `http://www.immagic.com/eLibrary/ARCHIVES/GENERAL/CMU_US/C050728W.pdf` as at April, 2017.

Wikipedia.

List of Linux distributions. Available online at `https://en.wikipedia.org/wiki/List_of_Linux_distributions` as at April, 2017.

LibreOffice, `https://www.libreoffice.org/` as at March, 2017.

Apache OpenOffice, `https://openoffice.apache.org/` as at March, 2017.

[Grigore, 2017] R. Grigore.

Java Generics are Turing Complete. Available online at `https://arxiv.org/pdf/1605.05274.pdf` as at April, 2017.

[Madey and Freeh, 2017] G. Madey and V. Freeh.

The Open Source Software Development Phenomenon: An Analysis Based on Social Network Theory, Eighth Americas Conference on Information Systems. Available online at `http://www3.nd.edu/~oss/Papers/amcis_oss.pdf` as at Feb, 2017.

[Shull et al., 2017] F. Shull, et al.

Open Systems Architecture: Progress and Challenges, Software Engineering Institute, Carnegie Mellon University. Available online at `http://resources.sei.cmu.edu/asset_files/Presentation/2015_017_001_447415.pdf` as at April, 2017.