Software Engineering Process and Practices for Data Science

Junhua Ding, PhD
Department of Information Science
University of North Texas
Software Crisis

• The difficulty to deliver **useful** and **efficient** software in the required **time** with planned **budget**.

• Coined at the first NATO Software Engineering Conference in **1968** at Garmisch, Germany.

• “The major cause of the software crisis is that the **machines have become several orders of magnitude more powerful**! To put it quite bluntly: as long as there were no machines, programming was no problem at all; when we had a few weak computers, programming became a mild problem, and now we have gigantic computers, programming has become an equally gigantic problem”.

  — Edsger Dijkstra, The Humble Programmer (EWD340), Communications of the ACM, 72 Turing Award Lecture
What is Software

• A collection of computer instructions and data that tell the computer how to work.

• Software = Algorithms + Data

• Powerful Computer → Powerful Software → Complex Algorithms + Big Data

• Software Engineering has been widely and successfully used for building Algorithms (Functions), but Not for Big Data.
Questions

• How should we build data intensive software?

• How can we integrate software engineering into data science for building data intensive software?
Examples (Why do we need Software Engineering?)

Rajpurkar and et al. introduced a deep learning system (ChexNet) for diagnosing pneumonia diseases based on chest X-ray images. They claimed "We find that ChexNet exceeds average radiologist performance on pneumonia detection on both sensitivity and specificity". (ref: https://arxiv.org/abs/1711.05225)

But Oakden-Raynera, a radiologist student and machine learning researcher, questioned the dataset used by ChexNet. He said: "I believe the ChestXray14 dataset, as it exists now, is not fit for training medical AI systems to do diagnostic work. (1). how accurate are the labels, (2). what do the labels actually mean, medically, and (3). how useful are the labels for image analysis".

(ref: https://lukeoakdenrayner.wordpress.com/2018/01/24/chexnet-an-in-depth-review/)

What data should we need, and how to evaluate the them?
A small number of bad samples added to the training data would diminish learning robustness. Bad samples can be easily generated using GAN.


• ``These results question the validity of a number of fMRI studies and may have a large impact on the interpretation of weakly significant neuroimaging results."

• ``Despite the popularity of fMRI as a tool for studying brain function, the statistical methods used have rarely been validated using real data. Validations have instead mainly been performed using simulated data, but it is obviously very hard to simulate the complex spatiotemporal noise that arises from a living human subject in an MR scanner."
Product search using catalog image as query. The system return similar looking image but the similarity was not very high.
Actionable Auditing: Investigating the Impact of Publicly Naming Biased Performance Results of Commercial AI Products, by I. Deborah Raji, and J. Buolamwini, AAAI 2019,

- To analyze gender and skin type performance disparities in commercial facial analysis models.

<table>
<thead>
<tr>
<th>Company</th>
<th>All</th>
<th>Females</th>
<th>Males</th>
<th>Darker</th>
<th>Lighter</th>
<th>DF</th>
<th>DM</th>
<th>LF</th>
<th>LM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Face ++</td>
<td>-8.3</td>
<td>-18.7</td>
<td>0.2</td>
<td>-13.9</td>
<td>-3.9</td>
<td>-30.4</td>
<td>0.6</td>
<td>-8.5</td>
<td>-0.3</td>
</tr>
<tr>
<td>MSFT</td>
<td>-5.72</td>
<td>-9.70</td>
<td>-2.45</td>
<td>-12.01</td>
<td>-0.45</td>
<td>-19.28</td>
<td>-5.67</td>
<td>-1.06</td>
<td>0.00</td>
</tr>
<tr>
<td>IBM</td>
<td>-7.69</td>
<td>-10.74</td>
<td>-5.17</td>
<td>-14.24</td>
<td>-1.93</td>
<td>-17.73</td>
<td>-11.37</td>
<td>-4.43</td>
<td>-0.04</td>
</tr>
</tbody>
</table>

Table 1: Overall Error on Pilot Parliaments Benchmark, August 2018 (%)

<table>
<thead>
<tr>
<th>Company</th>
<th>All</th>
<th>Females</th>
<th>Males</th>
<th>Darker</th>
<th>Lighter</th>
<th>DF</th>
<th>DM</th>
<th>LF</th>
<th>LM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Target Corporations</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Face ++</td>
<td>1.6</td>
<td>2.5</td>
<td>0.9</td>
<td>2.6</td>
<td>0.7</td>
<td>4.1</td>
<td>1.3</td>
<td>1.0</td>
<td>0.5</td>
</tr>
<tr>
<td>MSFT</td>
<td>0.48</td>
<td>0.90</td>
<td>0.15</td>
<td>0.89</td>
<td>0.15</td>
<td>1.52</td>
<td>0.33</td>
<td>0.34</td>
<td>0.00</td>
</tr>
<tr>
<td>IBM</td>
<td>4.41</td>
<td>9.36</td>
<td>0.43</td>
<td>8.16</td>
<td>1.17</td>
<td>16.97</td>
<td>0.63</td>
<td>2.37</td>
<td>0.26</td>
</tr>
<tr>
<td>Non-Target Corporations</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Amazon</td>
<td>8.66</td>
<td>18.73</td>
<td>0.57</td>
<td>15.11</td>
<td>3.08</td>
<td>31.37</td>
<td>1.26</td>
<td>7.12</td>
<td>0.00</td>
</tr>
<tr>
<td>Kairos</td>
<td>6.60</td>
<td>14.10</td>
<td>0.60</td>
<td>11.10</td>
<td>2.80</td>
<td>22.50</td>
<td>1.30</td>
<td>6.40</td>
<td>0.00</td>
</tr>
</tbody>
</table>
Software Life Cycle vs. Data Science Life Cycle

**THE SOFTWARE DEVELOPMENT CYCLE**

1. PLANNING
2. ANALYSIS
3. DESIGN
4. IMPLEMENTATION
5. TESTING & INTEGRATION
6. MAINTENANCE

**DATA SCIENCE LIFECYCLE**

1. BUSINESS UNDERSTANDING
   - Ask relevant questions and define objectives for the problem that needs to be tackled.

2. DATA MINING
   - Gather and scrape the data necessary for the project.

3. DATA CLEANING
   - Fix the inconsistencies within the data and handle the missing values.

4. DATA EXPLORATION
   - Form hypotheses about your defined problem by visually analyzing the data.

5. FEATURE ENGINEERING
   - Select important features and construct more meaningful ones using the raw data that you have.

6. PREDICTIVE MODELING
   - Train machine learning models, evaluate their performance, and use them to make predictions.

7. DATA VISUALIZATION
   - Communicate the findings with key stakeholders using plots and interactive visualizations.

---


http://sudeep.co/data-science/Understanding-the-Data-Science-Lifecycle/, by Sudeep Agarwal
Agile Development

The Scrum Software Development Process

Inputs from customers, team, managers & execs.

Product Owner

Product Backlog

1. Prioritized list of what is required: features, bugs...
2. Sprint Planning Meeting
3. Team selects starting at top as much as it can commit to deliver by end of Sprint
4. Task Breakout
5. Sprint Backlog

1-4 week Sprint

Sprint end date and team deliverable do not change

Sprint Master

Daily Stand Up Meeting

Sprint Review

Finished Work

Sprint Retrospective

https://www.maxxor.com/software-development-process
DevOps

https://medium.com/devopslinks/devops-without-devops-tools-3f1deb451b1c, by Jagatveer Singh
Tools

- Git, Github, JIRA, Stack
- PSPP/SPSS, Tabular, SAS, etc.
- Apach WeKa
- Google Tensorflow
- Facebook PyTorch
- MongoDB
- Jupyter Notebook, Framework Pandas, TF Learn
- ......
Datasets v.s. Program Libraries

- Kaggle
- ImageNet
- NIST
- Government agencies

Integrate Software Engineering Process and Practices into Data Science Project Development

Evaluate it before use it.

- Fit for purpose: Fidelity, Variety, Veracity?
- Intrinsic: Completeness, Correctness, ...?