ECE444: Software Engineering
AI for SE

Shurui Zhou
Learning Goals

Understand the AI-enhanced software development processes in practice
The Emerging Role of Data Scientists on Software Development Teams

Miryung Kim  
UCLA  
Los Angeles, CA, USA  
miryung@cs.ucla.edu

Thomas Zimmermann  
Microsoft Research  
Redmond, WA, USA

Robert DeLine  

Andrew Begel  
{tzimmer, rdeline, andrew.begel}@microsoft.com
Data Scientist: The Sexiest Job of the 21st Century

by Thomas H. Davenport and D.J. Patil
FROM THE OCTOBER 2012 ISSUE

When Jonathan Goldman arrived for work in June 2006 at LinkedIn, the business networking site, the place still felt like a start-up. The company had just under 8 million accounts, and the number was growing quickly as existing members invited their friends and colleagues to join. But users weren’t seeking out connections with the people who were already on the site at the rate executives had expected. Something was apparently missing in the social experience. As one LinkedIn manager put it, “It was like arriving at a conference reception and discovering there was no one there.”

To fix this problem, the team created an algorithm that identifies and suggests potential connections. The algorithm finds people who, based on shared friends or industry membership, might be good candidates for a friendship. This feature, which is now called “Second-degree Connections,” has since become LinkedIn’s most popular feature. But it took a lot of work to get there. And that work was just the beginning of something that would become a hallmark of LinkedIn’s success—its ability to use data to create new products and services.
Typical data science workflow

Acquire data
- Reformat and clean data
- Explore alternatives

Preparation

Analysis
- Edit analysis scripts
- Debug
- Execute scripts
- Inspect outputs

Dissemination
- Make comparisons
- Take notes
- Hold meetings
- Write reports
- Deploy online
- Archive experiment
- Share experiment

Reflection

Why are Data Scientists Needed on SW Teams?

Software companies want to experiment with real users, e.g., A/B testing, flighting, games and rewards.

“Instead of having an army of testers to go off and generate a bunch of data, that data's already here. It's more authentic because it's real customers on real machines, real networks. You no longer have to simulate and anticipate what the customer's gonna do.” [P10]

Demand for Data Collection Rigor

What about storage, what about speed? What about legal, what about privacy? There is an entire gamut of things that you need to jump through hoops to collect the instrumentation. [P1]
What Do Data Scientists Work on?

**Performance Regression**
Are we getting better in terms of crashes or worse? [P3]

**Requirements Identification**
If you see the repetitive pattern where people don’t recognize, the feature is there. [P3]

**Root Cause Analysis**
What areas of the product are failing and why? [P3]

**Bug Prioritization**
Oh, cool. Now we know which bugs we should fix first. Then how can we reproduce this error? [P5]

**Server Anomaly Detection**
Is this application log abnormal w.r.t. the rest of the data? [P12]

**Failure Rate Estimation**
Is the beta ready to ship? [P8]

**Customer Understanding**
How long do our users use the app? [P1]
What are the most popular features? [P4]

**Cost Benefit Analysis**
How many customer service calls can we prevent if we detect this type of anomaly? [P9]
### Activities of Data Scientists

<table>
<thead>
<tr>
<th>Collecting</th>
<th>Building the data collection platform</th>
</tr>
</thead>
<tbody>
<tr>
<td>Building the experimentation platform</td>
<td></td>
</tr>
<tr>
<td>Injecting telemetry</td>
<td></td>
</tr>
<tr>
<td>Analyzing</td>
<td>Data merging and cleaning</td>
</tr>
<tr>
<td>Sampling</td>
<td></td>
</tr>
<tr>
<td>Shaping, feature selection</td>
<td></td>
</tr>
<tr>
<td>Defining sensible metrics</td>
<td></td>
</tr>
<tr>
<td>Building predictive models</td>
<td></td>
</tr>
<tr>
<td>Defining ground truth</td>
<td></td>
</tr>
<tr>
<td>Hypothesis testing</td>
<td></td>
</tr>
<tr>
<td>Using and Disseminating</td>
<td>Operationalizing models</td>
</tr>
<tr>
<td>Operationalizing models</td>
<td></td>
</tr>
<tr>
<td>Defining actions and triggers</td>
<td></td>
</tr>
<tr>
<td>Applying insights/models to business</td>
<td></td>
</tr>
</tbody>
</table>
Data Scientist Working Styles

• Insight Providers
• Modeling Specialists
• Platform Builders
• Polymaths
• Team Leaders
“Please list up to five questions you would like [a team of data scientists who specialize in studying how software is developed] to answer.”
What metrics are the best predictors of failures?

If I increase test coverage, will that actually increase software quality?

What is the data quality level used in empirical studies and how much does it actually matter?

Are there any metrics that are indicators of failures in both Open Source and Commercial domains?

I just submitted a bug report. Will it be fixed?

How can I tell if a piece of software will have vulnerabilities?

Should I be writing unit tests in my software project?

Do cross-cutting concerns cause defects?

Is strong code ownership good or bad for software quality?

Does Test Driven Development (TDD) produce better code in shorter time?

Does Distributed/Global software development affect quality?
How would you approach these questions with data?

• Where to focus testing effort?
• Is our review practice effective?
• Is the expensive static analysis tool paying off?
• Should we invest in security training?
• What is a good team size?
Evaluate Effectiveness of an Intervention

- Controlled experiments
  - Compare group with intervention against control group without,
  - Randomized controlled trials, AB testing, ...
  - Ideally blinded

- Natural experiments, Quasi experiments
  - Compare similar groups that naturally only differ in the intervention
  - No randomized assignment of treatment condition

- Time series analyses
  - Compare measures before and after intervention, preferably across groups with the intervention at different times
Requirement
Engineering breakthrough: IBM introduces Watson AI for RQA

By Maggie Mae Armstrong | 2 minute read | February 28, 2019

Watson AI uses natural language processing and understanding to analyze a requirement’s text, suggesting improvements that leverage industry best practices for writing high quality requirements, based on the INCOSE Guidelines for Writing Good Requirements.

Meet Alice: Your Cognitive Assistant for Business Analysis

Your requirements gathering is about to get easier, better and faster. How? Artificial intelligence (AI) and machine learning. The most tedious part of the requirements process can often be gathering and elicitation. Yet that part of the process is well-suited for AI’s capabilities.
Requirement Analysis

• Detection of Hidden Feature Requests from Massive Chat Messages via Deep Siamese Network. Shi et al. (ICSE), 2020
Design
Meet AIDA: Your Artificial Intelligence Design Assistant

One of the big topics in design right now is artificial intelligence. Can a computer program actually design a website? Can it help a person speed up or improve the process?

Bookmark is taking the theory to a whole new level with its Artificial Intelligence Design Assistant, or AIDA for short. AIDA learns your needs and desires and uses this knowledge to create the perfect website for you. Today we’re taking a look at how it works!

The Ultimate Designer Toolkit: 2 Million+ Assets
Coding
Give your development team AI superpowers

Codota automatically learns the patterns and rules in your company's proprietary code and makes sure your developers have the best code insights, whenever and wherever they need them.

Kite adds AI powered code completions to your code editor, giving developers superpowers.

```
import os
import sys

def count_py_files_in_repo(dirname):
    
```
Use machine-learning-assisted code completion

You can utilize machine learning models to rank most suitable items higher in the suggestions list.

To do this, in the Settings/Preferences dialog \(\text{Preferences}\), go to Editor | General | Code Completion and enable the Rank completion suggestions based on Machine Learning option under Machine Learning-Assisted Completion.

ℹ️ This feature is experimental, so ranking may not change noticeably.
loss = tf.reduce_sum(tf.square(linear_model - y))

optimizer = tf.train.GradientDescentOptimizer(0.01)

train = optimizer
Deep TabNine: A Powerful Code Autocompleter For Developers

Amazing!! Deep Learning-based NLP techniques are going to revolutionize the way we write software. Here's Deep TabNine, a GPT-2 model trained on around 2 million files from GitHub. Details at tabnine.com/blog/deep nlproc

https://medium.com/syncedreview/dep-tabnine-a-powerful-ai-code-autocompleter-for-developers-70454a5953fe
With GPT-3, I built a layout generator where you just describe any layout you want, and it generates the JSX code for you.

**WHAT**

**Describe a layout.**

Just describe any layout you want, and it’ll try to render below!

```jsx
<h1 style={{fontSize: 50, color: 'white'}}>WELCOME TO MY NEWSLETTER</h1>
<button style={{color: 'white', backgroundColor: 'blue'}}><b>Subscribe</b></button>
```

Welcome to my Newsletter
Aroma: Using machine learning for code recommendation
Debugging

https://deepsource.io/blog/exponential-cost-of-fixing-bugs/
Ubisoft: ML catches 70% of bugs prior to testing

“The statistical nature of machine learning involves us changing the way we work,” he says. Unlike traditional software, in which developers write out rules for the application to follow, machine-learning algorithms use data to guide how the software should act.

-- Yves Jacquier, executive director, production studio services, Ubisoft Montreal

https://www.pmi.org/learning/library/ai-debug-code-11523
Debugging

Getafix: How Facebook tools learn to fix bugs automatically

Case study: Functionize Eliminates Agvance Test Maintenance with Machine Learning

September 11, 2018 - Ankur Verma
DeepCode

STEP 1: PARSING
This is the only language-specific part of our platform and enables us to add support for any custom language in a matter of weeks.

STEP 2: SOLVERS
Our custom language-independent linear-complexity Datalog solvers allow us to analyze huge repositories in a matter of seconds.

STEP 3: ML ALGORITHMS
Our custom Semantic Facts representations allow us to run powerful ML algorithms to understand the structure, function, and intent of the code.

AI KNOWLEDGE BASE

AI CODE REVIEW

AI QA AUDIT
CodeQL helps you explore code quickly to find and eradicate all variants of vulnerabilities before they become a problem.

By automating variant analysis, CodeQL enables product security teams to find zero-days and variants of critical vulnerabilities.
Deployment
Continuous app delivery firm ™

BY MIKE WHEATLEY

https://siliconangle.com/2019/04/23/continuous-app-delivery-firm-harness-raises-60m/
Project Management
How CraneAi uses Artificial Intelligence to help teams build apps faster

Behind the scenes look at how CraneAi’s uses artificial intelligence to empower teams
“Tara’s mission is to help teams develop their plans with visibility and predictability.”

COMPANY NEWS
Ford and Cisco are turning to an AI company to find the best freelance programmers

EDITOR
January 18, 2018
Trade-off?
Software 2.0

I sometimes see people refer to neural networks as just “another tool in your machine learning toolbox”. They have some pros and cons, they work here or there, and sometimes you can use them to win Kaggle competitions. Unfortunately, this interpretation completely misses the forest for the trees. Neural networks are not just another classifier, they represent the beginning of a fundamental shift in how we write software. They are Software 2.0.

https://medium.com/@karpathy/software-2-0-a64152b37c35
I got myself a cool AI T-Shirt - then the sticker began to peel off.