# Coding interviews:
# What to expect and how to prepare

# whoami

- Software engineer by trade
- Interested in making IT-recruiting suck less

# Outline

1. Resume

2. Coding questions

3. How to interview your interviewers

# Software engineering resume

- People read resumes on autopilot.
- Don’t list every project you’ve worked on (page length 1-2)
- Contribution >> technology/frameworks.
- Explain in simple but detailed language.

# Good or bad?

1. *“Designed software application including: data modeling, software architecture design, software- hardware integration, user interface design, and database management“*

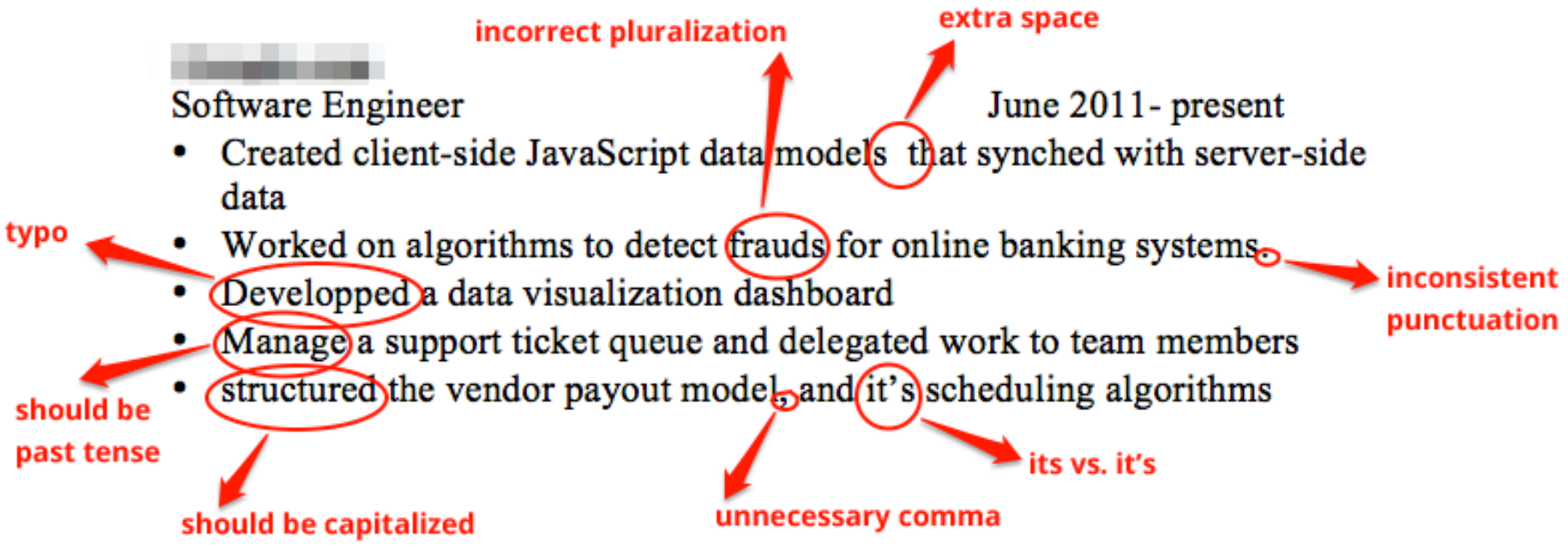
2. *“Created and launched a service that collects product opinions and recommendations from Twitter. The service finds related tweets, removes spam, analyzes sentiment and creates a structured database of everything that was said about particular products [link to demo]. The service is exposed as a consumer website and as widgets that can be embedded in online retail websites.“*

3. *“Developed [product name], using C# in .NET framework, for marketing and allowing end-users to experience [another product name]“*

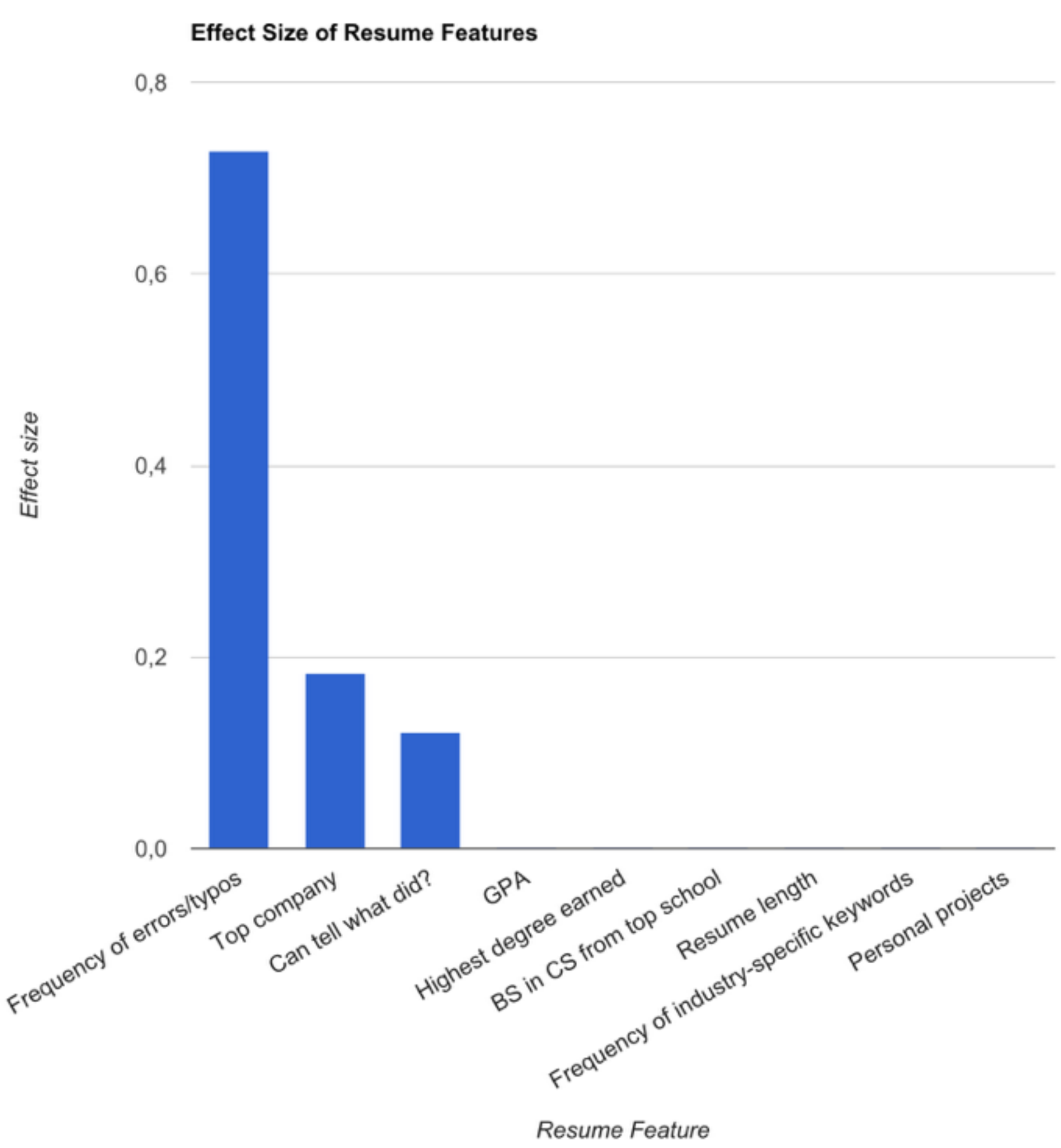
4. *“Evaluated and identified [OS name] network stack performance bottleneck in latency, per-packet processing overhead, and scalability of different network IO models through various system measurement and profiling techniques“*

http://blog.alinelerner.com/lessons-from-a-years-worth-of-hiring-data/

# Avoid typos




http://blog.alinelerner.com/lessons-from-a-years-worth-of-hiring-data/




http://blog.alinelerner.com/lessons-from-a-years-worth-of-hiring-data/

# Steps in a typical coding interview

- Phone interview (either depth or breath)
- Homework
- Show existing code
- Code smth small onsite (algorithms / data structure / practical)

# Coding interview prep

Tons of books:

- “Cracking the Coding Interview“ et al.

Tons of websites:

- interviewcake.com
- interviewing.io
- ….

# How to interview your interviewers: The Joel Test

1. Do you use source control?
2. Can you make a build in one step?
3. Do you make daily builds?
4. Do you have a bug database?
5. Do you fix bugs before writing new code?
6. Do you have an up-to-date schedule?
7. Do you have a spec?
8. Do programmers have quiet working conditions?
9. Do you use the best tools money can buy?
10. Do you have testers?
11. Do new candidates write code during their interview?
12. Do you do hallway usability testing?

# How to interview your interviewers

- If possible, ask for the opportunity to view the source code.
- If possible, ask for the opportunity to go with the guys for a beer.

**Bonus (if you feel comfortable):**

- "What is the most costly technical decision made early on that the company is living with now?"
- "Where do product / feature ideas generally come from?“

**Generally:**

- Don’t ask engineers about benefits/salary/vacations/process – you can get those answers later from HR or whoever.

# Contact

E-Mail: iwan@gulenko.ch

Twitter: @iwangulenko