Software Engineering
Day in the Life

COMPSCI 308 – 2022-04-11
Luke Moffett, Duke CS
Big Picture
Every Job will be Unique

Everything in these slides is meant to give you a flavor of what to consider about places of employment.

Orange – Unlikely to have encountered in your projects
Blue – Likely to have encountered in your projects
“The Best Job in America”

- Work is intellectually stimulating
- Colleagues are largely competent
- You actually build things that run at the end of the day
- Each day and task is different
- Culture of trying new things
- Constant Learning
- **Excellent** Pay
- Flexible work schedules and situations
- Many sources of job security

[https://xkcd.com/303/](https://xkcd.com/303/)
Common Responsibilities

This in no way represents everything you will do, but represents the kinds of things you can reasonably expect to spend your time on in most jobs.

- Planning and Designing (Agile Processes)
- Asking/Answering Questions (Email, Chat, In Person)
- Coding and Reviewing Code, including Test Automation
- Reading and Writing Documentation
- Other Meetings
- Learning
Example Day

- Catch-Up on Emails/Chats
- Plan Day/Prep
- Stand-up Report
- Daily Stand-up
- Follow-up with Peers from Stand-up
- Code Your Stories
  (maybe Pair Programming)
  (Lots of interrupts to answer chats)
- Lunch
- Backlog Refinement or Sprint Planning
- Answer Questions for Peers
- Finish Daily Coding
  (Grab a peer for help)
- Meet with Manager
- Attend Training
Work Style features that Vary Significantly Between Places

- On-Call Responsibilities
- Degree of Independence/Explicitness of Rules
- Frequency of Releases and Degree of Automation
- Technical Sophistication of “Platforms”
- Technical sophistication of management and peers
- Remote vs. In-Person vs. Hybrid
- Pair vs. Independent Programming
- Synchronous vs. Asynchronous Communication
- “Culture”

[Link to XKCD Comic](https://xkcd.com/2562/)
Daily Challenges of Software Work

• Estimation is extremely difficult, which makes effective planning a constant challenge
• Very difficult to decide what’s “good enough” – scope is always in flux
• More features than you could ever get done
• Getting access to the knowledge you need (made worse with poor documentation)
• Trade-off between investing for the future and delivering now
  • Shortcuts lead to tech debt, which makes everything slower and harder
• Many ways to do things and standardization takes effort
• Software verification (testing) is an art and not a science

https://xkcd.com/619/
Daily Challenges of Software Work

• Rarely working on a project from scratch and need to account for “current state”

• Major decisions are made by higher-ups who don’t understand technical details

• Continuous Learning
  • Need facility in dozens of tools
  • Tools change too quickly to master of all of them

• Subject matter experts are frequently not technically knowledgeable

• Many decisions are made for non-technical reasons (ie, cost of contract)

• Engineering teams have to choose implementations within their ability

• Complexity builds up over time and has to be continually pruned

https://xkcd.com/979/
Other Random, Frequent Tasks

- Presenting/Demoing Your Work
- Team Building
- Formal Trainings
- Assessing 3rd Party Software
- Giving and Receiving Feedback

- Meeting with Management (1-on-1s)
- Automating Small Tasks
- Educating Management
- Work Overhead (Timesheets, Corporate Training, etc)
- Travelling and Attending Conferences

https://xkcd.com/1168/