Everyday Ethics

Examples for Software Teams

Brian Myers, PhD, CISSP

Author
Technical Writer
Program Manager
Software Developer
Lead Developer
Project Manager
Software Architect
Product Manager
VP of Engineering
Director, Software Development
Information Security Architect
Director, Information Security
Senior Application Security Architect

SYBEX

NETSCAPE®

Borland

WebMD®
health services

WorkBoard

Ethical Constraints and Guidelines in Business

Professional codes of ethics	(ISC)2, OWASP, CEH
Laws and regulations	SOX, HIPAA, GDPR, CCPA, EAR…
Company policies	Corporate Ethics, Code of Conduct, Acceptable Use, Employee Sanctions
Company goals	profitability, reputation, customer satisfaction
Personal ethos	morals, upbringing

Ethical Constraints and Guidelines in Business

Laws and regulations	Protect the public and the shareholders
Company policies	Don’t do anything that could get the company in trouble.
Professional codes of ethics	Act honorably, honestly, justly, responsibly, and legally. Provide diligent and competent service to the people paying you.
Company goals	Make money. Keep customers and shareholders happy. Help your team. Win.
Personal ethos	Be fair and reasonable. Earn your salary. Respect others. Foster a constructive work environment.

You build a reusable component for your company’s product.

You publish the same code in your own GitHub repository so other programmers can benefit.

You are writing code for work and you find an open source library that lets you finish the job two days earlier than scheduled.

Someone outside your company asks you to show them what you are working on.

Your company is designing a brand new product.

Someone you know has an account on a competing product.

The person offers to log in and show you what the competing product does.

You are a manager. You have a 1:1 with a team member at a beer hall after work.

The manager of one development team in your company often takes his team out for beers after work at a strip club.

You have a fulltime job building web page front ends.

Occasionally you take small jobs in the side doing similar work for other companies.

A vendor from whom your company might buy software offers to take you out to dinner.

Your team is brainstorming about requirements for a new product that will create online communities for people who are coping with cancer.

Your team is evaluating candidates to fill an opening and rejects one as being not a good cultural fit.

Someone in your company runs a BitTorrent server with music files and shares it internally so everyone can stream music to their headphones.

You’re building a web page that must be accessible.

The basic logic takes longer to implement than you expect.

To meet your deadline you settle for putting in only some of the accessibility code, and you don't have time to test even that.

An auditor asks you whether your company policy requires management approval for every change to the live product.

You know that your policy does.

You also know that many changes go through without approval. No one checks.

The same auditor asks you to provide a JIRA query showing change tickets in order to verify that each was tested before release.

Not all tickets show testing. What do you do?

Your company requires that every code change be reviewed by another developer.

Someone on your team makes three or four changes and sends them to you for review.

You are behind so you approve the changes but don't actually review the code.

The website you sell has an idle session timeout of 30 minutes.

It also integrates with Slack, which has no idle timeout.

Users complain that Slack keeps making them log in.

Your product manager asks you to change the timeout to 72 hours.

You get reports from several customers that when they log in to your site they see data belonging to another user instead of their own.

You report an incident to your manager.

Your manager tells you not to talk to anyone else about what you saw, and not to put any details in email.

You become aware that two members of your team are in a relationship.

Your manager asks you to pull some user data from your website’s database and send it to someone at another company.

At your annual review your manager gives you a raise and asks you not to tell other people what you are making.

You discover a vulnerability in your website that a hacker could easily exploit.

It could expose lots of sensitive documents.

As far as you know no one has exploited this vulnerability, but you can't prove it.

The fix will take three weeks.

Someone on your team shows up later than everyone else, often leaves early, and periodically laboriously pulls all the keys off his keyboard to clean it.

His work is good and he never misses deadlines, but other employees complain.

You are about to fly out to spend several days with a client helping them get started with your company’s software.

The sales person for this client instructs you not to tell the customer that they are using beta software. They don’t want to experiment. They want a tried and proven solution.

You are a manager.

Some of your company’s developers are contractors in India.

Someone on your team sends an email stating that Indian programmers are terrible.

Your boss drinks too much at a company party and becomes sexually aggressive with employees.

It happens again, twice more.

Then he posts in a Slack channel a photo of an employee along with a very lewd remark.

Your managers bring you into a phone call with a customer who wants to ask security questions.

The customer asks if SMS is still enabled for MFA (which they would clearly consider bad.)

You say, correctly, that it is still enabled.

Afterwards leadership and sales get mad at you: “Think at a higher level about what the customer is really asking. They want to know if we are secure, and we are.”

You are reviewing a contract from a customer.

The contract says all your encryption will be FIPS compliant.

You know your website is not FIPS compliant.

Only the weekly tape backups of your site are encrypted in a FIPS-compliant manner.

License and Attribution