

ACM Code of Ethics and Professional Conduct

[Home](#) > [Code Of Ethics](#) > [Case Study: Dark UX Patterns](#)

Case Study: Dark UX Patterns

The change request Stewart received was simple enough: replace the web site's rounded rectangle buttons with arrows and adjust the color palette to one that mixes red and green text. But when Stewart looked at the prototype, he found it confusing. The left arrow suggested that the web site would go back to a previous page or cancel some action; instead, this arrow replaced the button for accepting the company's default product. The right arrow, on the other hand, upgraded the user to the more expensive category; it also silently added a protection warranty without asking for confirmation. Stewart suggested to his manager that this confusing design would probably trick users into more expensive options that they didn't want. The response was that these were the changes requested by the client.

Shortly after the updates were released into their production system, Stewart's team was invited to a celebration. As a result of these changes, revenues at their client had increased significantly over the previous quarter. At the celebration, Stewart overheard some of the client's managers discussing the small increase for refunds by users who claimed that they didn't want the protection plan, but there weren't many. One manager noted several complaints from visually impaired users, who noted that the mixture of red and green text obscured important disclaimers about the product. "So what you're saying, then, is that the changes worked as planned," quipped one of the managers.

Analysis

Dark user experience (UX) patterns, which are designs that intend to trick users toward unintended (and often more expensive) options, cause harm. They can make users feel duped (Principle 1.2), provide deliberately misleading information (Principle 1.3), or discriminate against those with disabilities (Principle 1.4). Computing professionals have a moral obligation to use their skills to benefit the members of society (Principle 1.1), not to deceive them. Furthermore, the use of dark UX patterns is an affront to the dignity of users, violating Principle 2.1. Consequently, dark UX patterns violate several of the core principles of the Code.

The managers at Stewart's clients knew that these design changes would disrespect users' autonomy so as to increase revenues, thus failing to adhere to Principle 3.1. While the moral offense rests primarily on these individuals, Stewart and his manager also failed to exemplify several principles of the Code.



Specifically, their development and testing infrastructure should provide a more thorough evaluation of changes to their interface (Principle 2.5). The leadership of Stewart's company should articulate policies and processes (Principle 3.4) that ensure their clients' designs are not deceptive and harmful to users. Taking a stronger position in this regard would guide Stewart and his company toward greater compliance with the principles of the Code.

These case studies are designed for educational purposes to illustrate how to apply the Code to analyze complex situations. All names, businesses, places, events, and incidents are fictitious and are not intended to refer to actual entities.

Using the Code: Case Studies

Demonstrating how the principles of the Code can be applied to specific ethical challenges:

- **Malware Disruption**: Security vendors and government organizations collaborate to disrupt the operation of an ISP that hosts malware.
- **Medical Implant Risk Analysis**: A medical implant device maker creates a smart phone application to monitor and control the device.
- **Abusive Workplace Behavior**: A manager fails to address abusive behavior by a technical team leader.
- **Autonomous Active Response Weapons**: A defense contractor that specializes in autonomous vehicles begins to integrate automated weaponry.
- **Dark UX Patterns**: A web developer realizes that their client's requests are intended to trick users into making accidental and expensive purchases.
- **Malicious Inputs to Content Filters**: An Internet content filtering service deploys machine learning techniques to automate the classification of blocked content.
- **Accessibility in Software Development**: A web-based collaboration tool deploys a new inline feature that has significant accessibility issues.