

Information technology and software development have delivered enormous benefits to humankind. They have improved living standards for most of us and made many of our lives more enjoyable and rewarding. However, far too many software development projects have either failed outright or not delivered the expected benefits.

Contrary to popular belief, the main reasons for failed projects are not technical but instead are human factors. Until we recognise the importance of human factors and address the problems they cause, our industry will continue to experience failed projects.

Andrew Brown, Principal Consultant at Expleo, discusses the top five ways that human factors hold back software development.

1. Poorly estimated projects in planning stage

The main reason projects fail or are delivered late or over budget is that they are **poorly estimated**. Our poor estimations are driven by four human biases; anchoring, overconfidence effect, optimism bias, and the show-off hypothesis.

- **Anchoring** is where our estimate is anchored to an initial value and we fail to adjust far enough away from that value. Anchoring causes low estimation as follows. The business asks for a cost and time estimate, but whilst doing this it also asks if the software can be delivered by, say, the end of the year. Naturally, this date is far too early. When we produce an estimate, we adjust upwards. However, we do not adjust far enough.
- **Overconfidence effect** is where we have an excessive confidence in our own judgement that is not warranted by objective accuracy. This causes us to be overly confident in our own skill, our ability to influence others, and our ability to deliver, hence leading us to believe that we can deliver far earlier and at lower cost than is realistic.
- **Optimism bias** is where we have an unwarranted optimism. When estimating a project, this bias causes us to believe that we will be one of the lucky few not hit by an adverse event, despite ample evidence that most projects encounter several such events.



- **Show-off hypothesis** is where we engage in unnecessarily risky activities. In our evolutionary past, we engaged in such activities to demonstrate the quality of our genes to potential mates. On a modern-day project, this leads us to take on more risk than is prudent.

2. Irrational risk-taking

Once a project has been underestimated, sooner or later it will become overdue or over budget. When this occurs, the project tends to take crazy risks (risks where the expected costs exceed the benefits, or there is a high chance of catastrophe). In addition, overdue projects may behave irrationally in other ways, such as discarding partially developed code or features that have value.

We take crazy risks on projects because when that project becomes overdue, our perception of the situation changes from one of the gains to one of the losses. This in turn causes us to **switch our risk appetite from risk-averse to risk-seeking**, encouraging us to take crazy risks.

3. Organisations have too much technical debt

Most organisations complain that they have too much technical debt. However, at the same time as they complain, they are probably running a project which will add to that level of technical debt.

Technical debt is not a technical problem, although it sounds as if it should be. Technical debt is a problem of trade-off, between present and future, between costs and benefits. It is a classic vice versus virtue dilemma. Technical debt is a consequence of how we make decisions, which is by using a process that involves our emotions; the **affect heuristic**.

Organisations have excessive levels of technical debt because when they make the trade-off between technical debt and new features, technical debt does not weigh heavily in that

decision, using the affect heuristic. Consequently, organisations choose new features over technical debt and then complain about the technical debt they suffer.

4. Adding quality processes does not improve quality

Many organisations are unhappy with the quality of their finished software product, so they add in additional quality processes to try and catch more bugs. However, they are often disappointed to find that quality does not improve as much as is hoped.

This is because of **risk compensation**, sometimes known as the **Peltzman Effect**. This effect causes us to take more risks when we perceive safety has improved. It is why we drive faster when wearing a seatbelt. On a software project, it causes us to become a little less careful when we know that the project has put in some additional quality and test processes, to catch any mistakes we make.

5. We continue to fund zombie projects

Irrational escalation, sometimes known as **sunk cost effect**, is a bias that causes us to continue investing in a decision, due to the past investment that we have 'sunk' into it, despite evidence indicating that continued investment will never be repaid.

This bias causes organisations to continue to invest in zombie development projects that they would be better off killing and investing resources elsewhere.

Conclusion

All these problems, plus others, are important issues that hold back software development for most organisations. Although these problems can sometimes be caused by technical factors, human factors play a far more important role in driving these problems.

We have had these problems within software development for many decades and will continue to have these problems until we begin to recognise and address the human factors underlying these problems.

Other industries, such as aviation, medicine, and government policy, are already addressing similar problems caused by human factors. These problems are difficult, but by no means insurmountable. The rewards to organisations that successfully address these human factors and overcome these problems will be enormous.