# Initial post

**Initial post**

by Ruben Marques - Monday, 27 October 2025, 9:32 PM

**Collaborative Discussion 1: Project Failures Study – Enhanced Response**

Leaning on Agrawal's paper (2020):

**Question 1: What do you believe are the three most common reasons for project failure?**

Reading through the study made by Agrawal (2020), we can assess that the three main denominators for failures in software projects tend to be (ranked by per cent):

#1 Knowledge based errors

This type of error is most common during projects, being the main culprit approximately 31% of the time, mainly caused due to ***Lack of domain knowledge***, which happens when a software designer doesn't understand the entire scope of the business processes, consequently leading to design errors and poorly modelled architecture, ***insufficient technical knowledge*** and also making the wrong assumptions from stakeholder inputs.

#2  Organisational Influence Errors

This is mainly due to the constraints made by the organisation, especially time constraints (promising optimistic completion time to stakeholders with low flexibility), lack of budget and one that tends to happen mostly, miscommunication between involved parties.

#3 Cognitive-Load Errors

Mostly caused by stress, fatigue or lack of attention when they act.

**Question 2** – Two Concrete Failure Examples

Chat to us!

**1st Example from TSB Bank IT Migration Disaster (2018) (Knowledge + Organisational)**

Context: Migration of 5.2 million customer accounts from legacy Lloyds Banking Group system to new TSB platform.

Root causes of the failure:

- Lack of domain knowledge – developers didn't fully understand the complexity of legacy data structures and account relationships built over decade

- Insufficient technical knowledge – migration team underestimated data transformation requirements and failed to architect proper rollback mechanisms

- Time constraints with low flexibility – management set an immovable weekend migration window despite warnings from technical teams

- Miscommunication between parties – offshore development team, UK IT staff, and business stakeholders had misaligned understanding of migration readiness

Outcome: Led to 1.9 million customers being locked out of accounts, incorrect balances displayed fraud vulnerabilities exposed and £330+ million in costs

**2nd Example from NHS Contact Tracing App Failure (2020)**

Context: COVID-19 contact tracing app with a centralised data model.

Root causes of failure:

- Wrong assumptions – They believed Apple/Google would weaken privacy protections despite explicit policies blocking constant Bluetooth scanning

- Insufficient technical knowledge – disregarded Bluetooth performance challenges and difficulty of building workarounds

- Organisational chaos – management reorganisations caused script mismatches and the political pressure for a "world-beating" app overrode feasibility concerns

- Cognitive-load under stress – pandemic urgency caused team to ignore external warnings and focus only on potential upsides

Outcome: Complete abandonment, overspending, wasted time during critical pandemic response, forced switch to decentralized backup solution

**References**

Agrawal, T., Walia, G.S. and Anu, V.K. (2024) ‘Development of a Software Design Error Taxonomy: A Systematic Literature Review’, SN Computer Science, 5(7), p. 467. Springer. A
https://doi.org/10.1007/s42979-024-02797-2. [Accessed on: October 27th, 2025]