MASIN

IT Development Disputes: What Goes Wrong And How Experts Prove It

BY RAJAT SINGLA

Contents

Foreword

IT development disputes have become one of the most complex areas in modern commercial litigation and arbitration. Software and system delivery does not leave behind physical evidence like traditional projects. The evidence that remains is a mix of code, documentation, emails, system logs, and human interpretation.

The question is rarely just whether something went wrong when disputes arise. The real challenge is to prove how it went wrong, when it went wrong, and what part of the failure is legally and technically relevant.
This eBook explains how IT disputes are formed, why evidence becomes fragile, and how experts reconstruct the story of a project in a way that can withstand cross-examination in a tribunal.

CHAPTER 1: What Are IT Development Disputes and Why They Happen

IT development disputes arise when a project designed to deliver software, systems, or digital platforms does not meet the expectations of the parties.

These projects look structured on paper - scope defined, timelines agreed, milestones set, and payment linked to delivery. But the reality changes quickly once execution begins.

Most disputes begin with a mismatch in expectations.

One side believes that they are building a working system based on agreed requirements. The other believes those requirements were never fully met.

What makes IT disputes complex is that they are rarely about one single issue. They usually involve a combination of:

- Delays in delivery
- Missed milestones
- Systems not performing as expected
- Disagreements over scope
- Payment withheld or delayed
- Technical disagreement on what working means

At the centre of every dispute are three core questions:

What was promised?
What was delivered?
How do we prove the difference?

Where do disputes begin?

Most IT disputes do not begin when all things go wrong, and everything starts to fail. They begin and these are the small warning signs that are ignored:

- Reports that do not clearly reflect progress
- Informal instructions replacing formal approvals
- Changes in requirements without documentation
- Increasing reliance on emails or verbal agreements
- Pressure to deliver without updated scope clarity

When you realize that the project is in trouble, it is already too late, as the disagreements have already existed for months.

What makes these disputes complex?

What makes IT disputes complex is the tangibility factor. Whenever something goes wrong in a dispute, you can see the damage yourself. But when it comes to IT disputes, you cannot really figure out as the truth is technical.
You cannot always see failure. You must interpret it through:

- System behaviour
- Logs and records
- Code versions
- Testing outputs
- Human interpretation of technical work

This is where experts become critical. They translate technical reality so that the tribunal can understand.

CHAPTER 2: IT vs Construction Disputes - Same Problems, Different Evidence

One of the strongest comparisons in IT disputes is with construction projects. When you shall look at it at the first glance, both industries shall look very similar.

Both involve:

- Large teams
- Fixed timelines
- Complex delivery stages
- High-cost overruns when things go wrong

Both suffer from the same root causes:

- Delay
- Defects
- Miscommunication
- Scope changes

The key similarities:

IT and construction projects rely on specialists working under time pressure. The expectations between stakeholders start to shift when execution begins. This is the shift where disputes begin.

Let's go through the differences:

Evidence

In construction, the evidence is physical.

You can see a cracked wall, delayed structure, or a faulty installation. You can photograph it, measure it, and inspect it later.

But in IT, the evidence is invisible.

You deal with code, systems environments that may no longer exist, supplier-owned infrastructure.

If systems change or update, then it is even more difficult to find evidence as the original evidence may be gone.

Why do IT evidence disappears faster?

The main reasons are:

Missing documentation

When the clients ask the service provider to make any changes to the software, then approval remains informal, even though the work may have been accomplished. This creates problems later as there is no record.

Scope confusion

As the project progresses, it is naturally to have scope changes. But the dispute arises when they are not formally tracked.

Technical complexity

It is difficult to interpret data without expert analysis.

Key Takeaway

IT and construction projects share similar problems, but they differ in one key aspect - evidence. Construction evidence is visible and physical, while IT evidence is digital and often disappears over time. This makes IT disputes more about reconstruction than observation.

CHAPTER 3: Why Evidence Disappears Before the Dispute Even Begins

One of the most important realities in IT disputes is this that by the time lawyers or experts are involved, the evidence is already gone or fragmented. It happens during project delivery.

Documentation gaps

Most IT projects move fast. The teams are more focused on delivery, rather than on documentation. There are two common issues that appear:

Miscommunication in variations

The changes are discussed informally, implemented quickly, but are never documented properly. This leaves no room for proof of approval.

Weak progress reporting

The reports don't reflect a true picture of reality. The delays are not clearly recorded which makes it difficult to identify when the problems began.

Scope drift

Scope is one of the biggest causes of dispute.

Projects evolve:

- New technologies are added
- Requirements shift
- Resources change
- Priorities are adjusted

These changes are not made in formal contracts. This creates a gap between what was agreed and what was done. This later becomes a cause of dispute.

Vagueness in requirements

Many IT disputes come down to one issue - unclear expectations.

Testing is a key example. Different stakeholders may have different views on whether the system works or not. When there is no clarity on the acceptance criteria, disagreement becomes inevitable.

What are early warning signs?

Most disputes show early signals:

- Increasing email discussions instead of formal change requests
- Delays not clearly explained
- Growing disagreement on what is done
- Testing disagreements between teams
- Pressure to proceed without clarity

You must watch out for these signs, as once these signs appear, the project has already in a fragile state.

What are the challenges faced by tribunals?

The tribunals often face one core challenge when disputes reach arbitration or litigation.
They are not deciding what the contract says. They are trying to understand what happened. This depends entirely on the quality of evidence that survived the project.

Key takeaway

The real problem in IT disputes is not the dispute itself. It is the fact that key evidence is lost during project delivery, before the dispute even starts.

CHAPTER 4: Scope, Vagueness and the Start of Most Disputes

Scope is that one point where most of the IT disputes are quietly born. It defines what the system will do, what will be delivered, and what success looks like.

When it comes to IT projects, scope is rarely as fixed as it appears on papers.

Most of the disputes do not begin with disagreements at the end. They start with uncertainty at the very beginning.

How do scope problems begin?
Scope issues usually start when:

- A requirement is discussed but not clearly written down
- A feature is assumed rather than confirmed
- A change is agreed verbally and never formally recorded
- Teams move ahead under delivery pressure without updating documentation

When the project is moving and delivery is the priority, it doesn't feel like a problem. But these missing definitions become critical when things go wrong,

The core issue: mismatch of understanding

There might be varying interpretations of the same scope between different stakeholders in IT disputes.

- Client believes certain features were included
- Vendor believes those features were out of scope or optional
- Technical teams interpret requirements differently from business teams

This mismatch of understanding leads to dispute.

What is the difference between scope in IT and construction?

In construction, the scope is relatively fixed. You cannot build the third floor before the second floor already exists.

This sequencing is flexible in IT. Modules can be developed in parallel, features can be re-prioritised, architecture can change mid-way. This flexibility is powerful but it also creates ambiguity.

When scope reaches a tribunal?

Once a dispute reaches arbitration or litigation, the tribunal usually asks a simple question:
What was agreed in the beginning? Yet, answering this is not simple because contracts are high-level, requirements may be spread across multiple documents, emails and informal discussions fill the gaps. This is where interpretation becomes central.

What is the real risk of a vague scope?
A vague scope does not merely create confusion. It shifts risk. When scope is unclear then each side interprets responsibility differently, delays become harder to assign, payment disputes increase, and trust between parties reduces.
The real cause of dispute is not the absence of work, but disagreement about what the work was supposed to be done.

Key takeaway

Most IT disputes are not a result from problems in execution. They begin with vague or incomplete agreements made at the beginning of the project.

CHAPTER 5: Milestones & Progress Analysis

Milestones are meant to bring structure to IT projects. They break large systems into smaller, manageable stages and link delivery to payments.

However, in practice, milestones become one of the most disputed parts of a project. This is because tracking progress in IT is not always straightforward.

What does a milestone mean?

A milestone represents a defined stage of functionality and agreed outputs used to measure progress. It is not only the completion of work. Yet, milestones are open to different interpretations.

Why do milestone disputes happen?

Milestone disputes usually arise for a few common reasons:

- **Changing expectations**

The expectations during execution of the project may differ from what was agreed at the beginning.

- **Lack of clear criteria**

Each party applies its own standard when acceptance criteria is not defined clearly.

- **Gap between progress and perception**

A project may appear complete in reports but still fail important functional checks.

What is the core question in disputes?

When a milestone is disputed, tribunals ask whether it was achieved according to the contract. The final decision depends on evidence rather than opinion.

The challenge of reconstructing progress

When disputes arise, progress is often reconstructed using:

- Emails
- Project trackers
- Status reports
- Internal notes

Key Takeaway

Milestones are only useful when they are clearly defined and can be measured in an objective way. A lack of clarity often leads to disagreements about progress.

CHAPTER 6: Acceptance Testing and the Blurry Line of Completion

Acceptance testing is the final step in IT delivery. It is when the client checks if the system works as expected. But in real projects, this stage is often disputed.

What is acceptance testing supposed to do?

Acceptance testing is designed to check whether the agreed requirements have been met, confirm that the system is ready for use, and make it available for business operations. It is the point where work in progress becomes a delivered system.

Why does acceptance become disputed?

The reasons behind acceptance disputes are:

- Testing begins before the system is fully stable
- Different stakeholders interpret test results differently
- Partial functionality is accepted informally during use
- Issues are identified after the system goes live

As a result, uncertainty often arises over whether acceptance has actually taken place.

What are the types of acceptance?

Acceptance may take several forms in IT projects:

- Partial acceptance
- Conditional acceptance
- Phase-based acceptance
- Informal operational acceptance

Each form carries different contractual and commercial implications.

Why is acceptance testing difficult in IT?

Acceptance testing is difficult in IT because systems do not always behave in a fixed way. They may work partially, fail under certain conditions, or depend on other systems. The performance depends on the context in which the system is used.

What is the role of experts in acceptance disputes?

· Experts assist tribunals in understanding:
· Whether the agreed requirements have been met from a technical perspective
· Whether identified failures are critical or minor in nature
· Whether system behaviour aligns with expected performance
· They translate technical testing outcomes into clear, understandable conclusions.

Key Takeaway

Acceptance is not merely a signature. It is both a technical and contractual determination of whether the agreed requirements have been met.

CHAPTER 7: Variation, Change Requests and Informal Work

Very few IT projects fail exactly as originally planned. Most evolve continuously.

It projects evolve - new features are added, priorities shift, technology choices change. In many of the cases, this variation in work continues without any formal documentation.

This is where variations begin to create disputes.

What is variation?

A variation is any change in scope, requirements, design, timeline or resources.
Whenever there is any change, it must go through a formal approval process. But many changes happen informally, as a discussion in a meeting, over email, verbal agreement between teams, or a request from business users during development.

Why are informal changes dangerous?

Informal changes may seem sufficient in the initial stages of the project. They allow work to progress quickly and can seem to address evolving requirements without disrupting momentum.

However, what feels efficient in the moment can create significant problems later. A gap in understanding can emerge between the parties involved when changes are not formally documented.

They lead to such questions:

- Was the work included within the original scope, or was it an additional request?
- Was the change formally approved, or was it merely discussed?
- Should the work be compensated separately?

The stakeholders may have different interpretations of what was agreed upon without clear documentation. As a result, informal changes become one of the most common causes of payment disputes, scope disagreements, and strained client relationships.

What is the difference between IT and Construction?

If there are any changes made in construction, then they are visible. If the designs must be formally revised, and structural impact is easier to trace.
Whereas in IT, the changes are not very evident, and are not visible until the code is modified, multiple variations may exist, and it more challenging to show variations.

What Is the Real Issue?

The real issue is not that changes were made. The real issue comes down to the absence of a clear record reflecting these changes, and whether these changes were authorised or not.
So, when a dispute arises, the real question is whether this change can be traced back to an approved instruction.

Even legitimate work can become difficult to justify, claim, or defend when it is not available. When there is no evidence of approval, parties may disagree about what was requested, what was authorised, and who is responsible for the associated costs.

Key Takeaway

Most variation disputes are not about the work itself. They arise from a lack of clear authorisation and approval before the work is carried out.

CHAPTER 8: Reconstructing the Project - How Experts Build the Truth

The challenge in dispute resolution is to determine the course of events that resulted in dispute.

A significant amount of time has already passed by the time dispute reaches formal proceedings - project teams may have changed, systems may have evolved, and the original project environment may no longer exist.

The experts cannot rely on memory or hindsight. They need credible evidence.

Why is reconstruction necessary?

When a dispute becomes formal, the project may have already:

- Changed direction
- Adopted new systems or processes
- Lost access to original environments
- Moved beyond its initial documentation

This makes reconstruction critical. Experts reconstruct events by analysing the records that were created during the project.

Key Takeaway

Experts do not recreate based on opinions or assumptions. They reconstruct events using the evidence that still exists and then, build an objective timeline of what occurred.

CHAPTER 9: Defects, Performance Issues and Technical Disputes

Many IT disputes arise from disagreements over whether the final product meets the client's agreed requirements.

These disputes involve allegations of defects, performance failures, security vulnerabilities, integration problems, or incomplete functionality. However, the existence of a technical issue alone does not determine the outcome of a dispute.

The central question is whether the system complies with the requirements agreed by the parties.

Why are defects difficult to prove?

A software defect is often more difficult to identify than a physical defect. A physical defect can be seen and inspected directly.

A software defect on the other hand, may only appear when certain conditions exist, when specific data is entered, or when certain functions are used.

A defect may also appear intermittently or disappear after later updates to the system.

The process of proving a defect therefore requires more than simply showing that a problem exists. The evidence must demonstrate when the problem occurred, how it affected the system, and whether the system failed to meet the requirements agreed by the parties.

Key Takeaway

Most defect disputes arise because the parties disagree about whether the behaviour of the system meets the agreed requirements.

CHAPTER 10: How Tribunals Decide - Evidence, Experts and Outcomes

When an IT dispute reaches arbitration or litigation, the focus shifts from project delivery to proof.

The tribunal is not concerned about what each party believes in. Its role is to establish the truth through reliable evidence.

A persuasive narrative alone rarely decides a case. Documents, records, technical analysis, and credible evidence ultimately determine the tribunal's decision.

What is the role of experts?

Technology projects deal with complex systems, specialised terminology, and technical issues that fall outside the expertise of judges and arbitrators.

This is where the role of the expert becomes important. The expert must analyse the technical evidence and present it in a clear and simple way before the tribunal.

The expert’s task is not to advocate for either party. It is to assist the tribunal in understanding what happened, why it happened, and what impact it had.

Key Takeaway

In IT disputes, evidence does not simply support the case. It effectively defines the case itself.