

Try Before You Buy: Acceptance testing in technology contracts

Published 9 September 2019

User acceptance testing (UAT) is defined by the International Software Testing Qualifications Board as formal testing with respect to user needs, requirements, and business processes, which is conducted to determine whether or not a system satisfies certain acceptance criteria. It is an essential aspect of many types of technology contracts, but is often overlooked.

On the one hand, those without technical backgrounds may gloss over specification or testing details. On the other, software engineers and other technology specialists may fail to consider the commercial or legal implications of how the contract is worded. This note aims to break down the key aspects of the UAT process, and provides a checklist of things to consider from both a customer and a supplier's perspective.

WHY CARRY OUT USER ACCEPTANCE TESTING?

UAT enables the customer to determine whether or not to accept the particular software, hardware, infrastructure, or other technology in question. This is particularly important because after formally accepting the supplier's product, the customer will lose its ability to reject the system and receive a refund, absent a contractual breach or other course of action. The only remedy typically remaining for the customer after acceptance will be damages, which can be both difficult to ascertain and even more difficult to obtain.

Given the variety of things which can go wrong when a new product is deployed or integrated, customers will normally seek to carry out some type of user acceptance testing (UAT) in the final phase of development, before the application or system is moved into the final market or production environment. In plain English, this allows a customer to "try before they buy". In most cases, the customer may either accept the system as delivered, accept subject to requested modifications, or reject the system entirely.

COMMON COMPLICATIONS

The idea behind UAT sounds both straightforward and intuitive. However, in practice it can be complicated by a variety of factors. Each of the following must be considered on a case-by-case basis, in light of the relative bargaining power and informational asymmetry between the parties.

- **Back to basics.** Both parties will benefit from having clearly defined specifications within the contract itself. Ensuring that the supplier is fully aware of the customer's requirements from the outset will help to prevent complications from arising during the testing phase. It is perfectly acceptable to append technical drawings, diagrams or other helpful information to the contract: the parties should not feel pressured to convert these details into "legalese."
- **Pay or delay?** It may be worthwhile for the customer to condition payment on the successful completion of testing. Whereas it is not uncommon for customers to want upwards of four to six weeks in order to comprehensively test the product, the supplier will want to limit the amount of time available for testing, or otherwise ensure some form of payment before acceptance. This could be negotiated by way of licence fees, one-off charges for the testing itself, or otherwise limiting payment retention.
- **Have an internal strategy in place.** Testing is of no value without an internal strategy detailing how the product will be deemed to have met its business requirements. Typical deliverables for UAT testing will include a comprehensive testing plan, separate environment specifications, scenarios and test cases, test results and defect log. If other systems will be exposed during testing, it may also be worthwhile to revisit any disaster recovery or business continuity plans.
- **Failure vs Defect?** Because the supplier is the one with intimate knowledge of the product, it will normally take ownership of testing. However, it may be preferable to have the customer take the lead, especially if the new product is to be integrated into an existing ecosystem. The customer may also wish to carry out its own testing if the testing environment is under heightened security, or simply because it wishes to exercise more direct control over the process. In any event, the supplier should establish clear instructions and parameters. What constitutes a "failure" will need to be distinguished from a minor "defect", for example. The supplier must also consider the costs associated with fixes, patches, and other modifications, and whether any additional charges should apply in such instances.
- **Keep an eye on the clock.** All of the activities - ranging from installation, configuration, testing, analysis and modification - should adhere to an agreed timetable. The supplier will need to think carefully about what is realistic to

accomplish with the time and resources available. Likewise, if timing is critical, the customer should seek to include remedies if the product fails its testing, or delivery does not materialise on time. Most testing clauses will also include language to the effect that the customer will be deemed to have accepted the product, unless notification is received by the supplier by a certain date.

- **Maintain open lines of communication.** It is important to remember that technology contracts are collaborative endeavours. Development and modification can be the most difficult and nuanced part of an agreement, and can easily lead to disputes between customers and their suppliers if not handled appropriately. By having a dedicated point of contact and open lines of communication throughout the process, you may be able to avoid unnecessary headaches.

To discuss anything covered in this update, or technology contracts more generally, please contact Tim Ryan or your usual DACB contact.

Authors



Tim Ryan
London - Walbrook
+44(0)20 7894 6978
tryan@dacbeachcroft.com



Kelsey Farish
London - Walbrook
+44 (0) 20 7894 6320
kfarish@dacbeachcroft.com