



# SERVICES & PROJECT MANAGEMENT OVERVIEW

## COMMONTIME PROFESSIONAL SERVICES

*We offer an integrated complement of product and services that enable the development, deployment and management of mobile business applications. Our capability is based upon the CommonTime Infinity framework which provides for the development of cross-platform/device mobile applications capable of both online and offline working.*

*In addition, we also invest significantly in our staff. This ensures that we have in-depth technical expertise in all aspects of mobile computing, and extensive experience delivering solutions across multiple verticals. This document outlines our approach in more detail.*

### COMMONTIME

15 St Christopher's Way  
Derby, DE24 8JY

### CONTACT

T: +44 (0)845 009 0028  
E: [info@commontime.com](mailto:info@commontime.com)

### WEB

[www.commontime.com](http://www.commontime.com)  
[support.commontime.com](mailto:support.commontime.com)

# An Introduction to Our Professional Services

The CommonTime professional services team offer a range of options for delivering projects. We regularly provide services to clients at the start of a mobilisation journey, as well as the seasoned users and developers of mobile technology.

In all cases, it is imperative that we review the requirements and deliverables for complexity and risk, delivering these as early as possible to ensure success. Our PMO team co-ordinate technical and the client teams to ensure effective collaboration is achieved and maintained.

**Assisted Development:** Extended training, knowledge transfer and delivery of best practise in a way that allows the project to be delivered at an agreed upon pace. Typically, this starts with mDesign training and then moves on- site for extended assistance on either an ad-hoc or scheduled basis.

**Waterfall Development:** A classic methodology that allows CommonTime to really understand the business requirements which are delivered into a testing environment. This is ideal for projects where the client does not have resource capacity for extended collaboration. Early testing, demonstrations and short development cycles ensure client confidence.

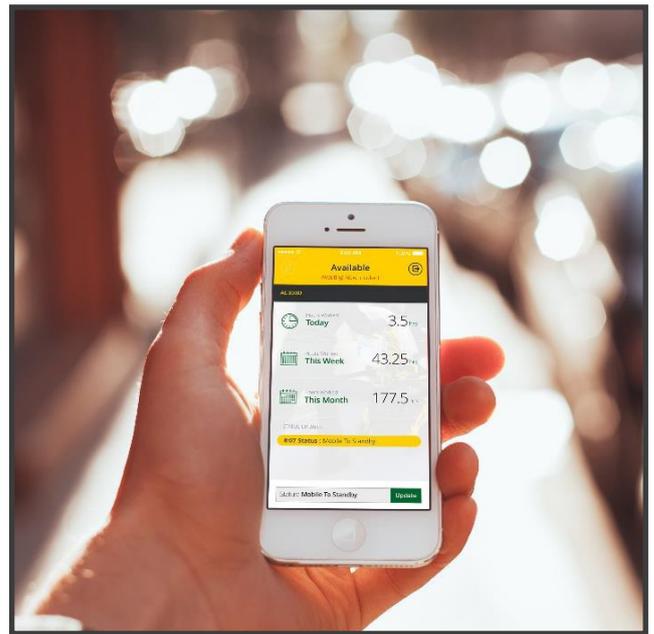
**Iterative Development:** Our agile methodology requires constant communication with client teams so the whole process is visible throughout. Once a level of specification, design and architecture is confirmed, development cycles are co-ordinated by our technical manager and delivered into a test state. This enables our team to deliver a MVP within tight timescales.

## Tailored Delivery Methods

Additionally, we are capable of tailoring any of these methods to client needs. Previous examples of how this has been achieved include:

**Off-shoring technical delivery:** We use the outsourced development pools of one of our corporate clients to develop technical deliverables, whilst we manage the project.

**Delivering complete applications:** A number of our health customers have specific local integration knowledge (using NHS data sets and supporting



frameworks) so we deliver the application front end with an integration layer exposed for clients to work with.

**Proof of Concept delivery:** Development of the minimum product to an outline spec rapidly allows the client to take something tangible into a board for sign off.

## Agile vs Waterfall

In addition to delivery service, we know that project management style is an important consideration. That's why our team are able to work in an agile, waterfall or hybrid manner depending on the needs of your particular project.

The agile approach, delivered on-site, offers lower costs and reductions in both development & deployment timescales. For this approach to be successful, clients are required to dedicate the necessary levels of management resource to facilitate rapid decision making.

Our traditional, off-site approach is suited to clients who prefer formal project management or where a fixed price is required. Our in-house specialist professional service teams work from a Project Information Document that specifies the scope of the chosen application. The delivery schedule may consist of a pilot followed by a full production roll-out.

# The Project Lifecycle

We split projects into three distinct stages. These help us effectively manage the entire lifecycle, from initial scoping to management & review. By outlining distinct end goals for stage, we ensure that deliverables can be produced to the highest possible standard and on-time. Here, we outline each phase in detail.

## 1. Discovery

Upon appointment CommonTime appoint a ring- fenced team for discovery. We engage in an in-depth exploration that looks at requirements for the overall delivery of the project.

## 2. Project

Following the discovery phase, the project and team adopt the strategy as defined in the discovery phase report. This is an iterative and collaborative process, ensuring client access.

## 3. Handover

The principle aim of the project is on-schedule delivery, within the defined parameters. To ensure this, the steering team manage change & risk whilst retaining clear visibility of progress.

## Discovery Phase Details

CommonTime would engage in a full discovery process that would be required for the overall delivery of the project. The discovery process would allow CommonTime to collaboratively draw up a series of functional and technical specifications with you. This includes the works required to integrate the various systems and draw up any supporting business processes that would be required.

We generally estimate that the discovery phase lasts up to 3-4 weeks and requires specific subject matter experts for the duration of appropriate workshops, output review and subsequent follow up sessions. Once agreed, the requirements are used to create the following plans & specifications: Development plan, Quality plan, Launch plan, Roll out phase plan, Business process document, Functional specification, Project timeline/plan.

## Stages of Discovery

In a typical project, we would expect the discovery process to comprise of:

An initial introduction workshop/ requirements scope briefing (output being a high level common scope and skeleton timeline for discovery).

A jointly owned timeline for a series of functional workshops:

1. Data sources and repositories
2. Data flow / integrations
3. Business processes workshops
4. Data sources and repositories 2
5. UI/UX requirements
6. Style guidance
7. Reporting

## Expected Results

During the discovery phase, CommonTime appoint a ring- fenced team that is tasked with producing:

1. Detailed and aligned plan
2. Functional, integration & data specs
3. Use Cases (RUP Format)
4. Wireframes (where appropriate)
5. Business specification (or a timeline)
6. Day to day business rules
7. Business scenarios
8. Risk, & change log
9. Approach strategy
10. Straw Man/ personas for test cases
  - a) Failure processes
  - b) Governance process
  - c) Communication document

## 1. Discovery

Upon appointment CommonTime appoint a ring- fenced team for discovery. We engage in an in-depth exploration that looks at requirements for the overall delivery of the project.

## 2. Project

Following the discovery phase, the project and team adopt the strategy as defined in the discovery phase report. This is an iterative and collaborative process, ensuring client access.

## 3. Handover

The principle aim of the project is on-schedule delivery, within the defined parameters. To ensure this, the steering team manage change & risk whilst retaining clear visibility of progress.

## Project Phase Details

During this stage, business use cases and technical documents are prepared and shared prior to sign off and development proceeding. Use cases and Straw men can be turned into test cases for the development and QA functions. The test cases are then circulated to all areas of the businesses for review and sign off which would allow the UAT to be given a full scope.

### Strategy

Following the discovery phase, the project and team will adopt the strategy as defined in the discovery phase report. This strategy generally includes:

- Steering group frequency
- Project team call frequency
- Working to & reporting against the plan
- Development process
- Handover planning to support functions

### Example Outcomes

The development process proposed would be iterative and collaborative. This would ensure:

- An agile approach to the development
- Development to continue in line with analysis and requirements gathering
- Earlier testing and demonstration
- A feeling of inclusion for all parties
- Possibility for phased launch

## 1. Discovery

Upon appointment CommonTime appoint a ring- fenced team for discovery. We engage in an in-depth exploration that looks at requirements for the overall delivery of the project.

## 2. Project

Following the discovery phase, the project and team adopt the strategy as defined in the discovery phase report. This is an iterative and collaborative process, ensuring client access.

## 3. Handover

The principle aim of the project is on-schedule delivery, within the defined parameters. To ensure this, the steering team manage change & risk whilst retaining clear visibility of progress.

## Handover Phase Details

As the aim of the project is successful completion on schedule and within defined parameters, the steering team must also manage & and risk - with clear visibility over the process. Therefore, it is important that a launch strategy is co-ordinated with CommonTime to ensure maximum impact whilst minimising disruption to end users.

A handover phase can include various support elements, such as; technical support, monitoring services, design, ongoing updates and additional resource.

# The CommonTime Team

When utilising CommonTime's Professional Services, we assign an on-shore team for the duration of the project with along with key internal stakeholders. The project team are primarily be based in Derby, with co-location options available during key points in the implementation.

The technical team has decades of experience in mobile and comprises several multi-disciplined developers, experienced in: C#, .NET, C++/C, Java, ObjectiveC, XML/XSLT, JavaScript, HTML, CSS, iOS, Android, WP8 Windows 8, Linux, AWS and Microsoft Azure.



Paul Lawrie | Project Manager

Paul delivers multi-party projects on time and budget. Paul has worked with Primark, Molson Coors and IBM - leading projects with complex integrations and aggressive timescales. Paul also brings a wealth eCommerce experience and co-ordinated Lockheed Martin, Cable & Wireless, Steria, Royal Mail, Polestar, BSS and Logica.



Ben Burt | PSO Development Manager

Ben joined CommonTime as manager of the development team - to oversee, manage and drive the quality of output. Ben originally comes from a healthcare background with Serverlec and has run large development teams for both internal and external projects working with both on and off shore resources.



Natalie Newall | Business Analyst

Natalie recently joined CommonTime from iTrade Network and has worked with several of the team before. Natalie brings a wealth of experience both in analysis and product management. We believe that this product management experience allied with the needs of any application development project will ensure success.



Nick Kipling | Senior PSO Solutions Architect

Nick has worked on many key projects in his role as architect and has been instrumental in the continuing development of our Infinity Server. Nick has helped to develop the architecture to support pager replacement applications, and recently trained a number of developers in India for a 3rd party (WIPRO) who are working with Carillion.



Richard Bourne | Senior Developer

Richard has worked with several our key clients including Carillion construction, Reconomy and London Southeastern Railway. Richard has a wealth of experience in the both the healthcare and private sector from his previous roles and is a respected expert in the mobile field.



Paul Winfield | Developer

Paul has 14 years of solutions development experience. He has a passion for learning and evaluating efficiency and performance. Paul spent 6 years with Drive Assist delivering vehicle delivery, tracking and management solutions. Paul leads the integrations teams on projects and is a specialist in Azure Mobile Services developments.

# Specialists in Complex Integration

In principle, we integrate into all leading back-end data architectures and are only bound by the limitations of technology. Some of our integration options include:

**Server Side Plug-in:** Designed and developed to consume a service deployed by a client. Can be used for server to server comms or as a data marshalling service when direct data integration is being used.

**Client Side Plug-In:** Typically used on a B2C application where we will need to integrate through to a cloud service. mDesign does not then feature in runtime as the integration capabilities are pushed to the client device.

**Service Bus:** We can integrate directly with ESB solutions such as Mule to deliver more complex integration where complex workflows are also required in the interface. These typically lend themselves to multi party integrations.

**PAAS Integration:** We have significant experience in delivering solutions using cloud based PAAS products and specifically Microsoft Azure Mobile services. This allows solutions to scale and offers significant savings compared to an on-premise equivalent.

We have also integrated successfully with a wide range of industry standard and proprietary systems. A selection of these is as follows:

- MS SQL DB / SharePoint / Salesforce / Oracle DB
- ODS (Arrow XL data system for manifest delivery)
- The NHS Message Information gateway
- AccuServ - Property Management
- TJX (TK MAXX) end systems
- Servelec - Care Solutions

## The Benefits of a Proprietary Studio

Apps created using CommonTime Infinity can choose to connect to systems and services in a variety of ways.

Apps can connect directly to existing APIs including cloud services, but can also make use of the Infinity Server to fulfil bespoke integration requirements or provide real-time messaging features. This can be installed either on-premise or in the cloud if required.

**REST APIs:** CommonTime Infinity can consume REST



APIs easily and has support for Swagger allowing endpoints to be imported into an app project. Apps can make use of the background REST connector to queue data while a device is offline and resume transmission when the user gets a mobile signal.

**Azure App Services:** The Azure App Services connector extends the REST API feature to include OAuth authentication with support for four identity providers: Azure Active Directory, Facebook, Google, Twitter. In addition, apps can also receive and transmit messages over the Azure Messaging Bus.

**Infinity Server:** CommonTime Infinity enables app creators to incorporate real-time chat functionality to any part of their app. This functionality can be used with any XMPP chat server. CommonTime can also configure a private server to run in the cloud or on-premise.

This chat functionality includes all the standard chat features such as: received receipts, read receipts, user rosters, queued delivery, end-to-end encryption, photo and compressed audio attachments.

Bots, written in any language (including JavaScript), can also be used for the purposes of enterprise data integration, accessing remote data on behalf of the app and informing users via push notifications when specific events occur.

# Environments & Quality Assurance

CommonTime regularly work within a multi-vendor environment, delivering business critical applications alongside a large number of stakeholders. We typically hold weekly teleconference sessions where all vendors and the customer are involved to ensure that activity is co-ordinated along the same timeline. Examples of this include:

**ArrowXL:** This was a business critical tool that required multiple integrations with existing systems and collaboration with new vendors. This project was run with regular scheduled and ad-hoc communication with third party vendors including TomTom, Motorola and Zebra.

**Don't Walk By:** This was developed alongside Carillion with third party vendors including K2, Wipro Ltd and Microsoft. K2 provided database structure and APIs for their own multi-tier platform, Wipro Ltd are a Project Management, Testing and Infrastructure Provision partner with Microsoft supporting the infrastructure and transition towards Azure-based AD.

## Ensuring Quality Delivery

CommonTime adopt a derivative of a SBTM (Session-Based Test Management), an established practice in RST (Rapid Software Testing), used frequently by contractors and in agile development environments. This has more test artefacts without introducing any onerous process. This process is described below:

Test planning begins with a review of available documentation and the identification of risks and questions. Once risks have been identified, our team are able to start creating Test Charters. These are areas of functionality that need to be tested, that should each take approximately 2 hours of testing.

Each session is recorded with separate notes detailing setup, tests and bugs; in addition to complex items such as additional questions for the project and possibilities for future investigation that are beyond the scope of the active charter. These notes can then be combined to generate a report.

Charters offer advantages such as being able to better estimate after-test planning, in addition to offering the project team an opportunity to reduce the testing time by placing individual charters out of scope for testing,

and accepting the risks therein. It also enables the test team to identify during testing areas of functionality missed from test planning, create additional charters flag changes to the project team, and add this as part of subsequent projects.

Where overruns or delays happen, we can also use SBTM to better inform our retrospectives - identifying problems in test planning, environments, code quality, resourcing, etc.

We are then able to raise issues in a system that allows us to monitor and complete the defect cycle with a known set of tests to validate closure. Beyond this we also perform rigorous tests of the application in general and are well placed to perform security and performance testing.

Every charter includes the following information:

**Charter:** The name of the charter, which is a sentence, describing what will be explored. The format for this should be the same as a bug name, terminated by a period.

**Areas:** What was covered, describe the test harness used under test (SUT) and the areas of the SUT that were actually tested. Each entry in this section should be on its own line.

**Start time & duration:** The start date and time, in addition to whether it is a short, medium or long test.

**Data Files:** Any data that was gathered during the test such as screen recordings or screen shots. The path to this is acceptable.

**Test Notes:** Step by step journal of the test. A sentence describing what actions were taken followed by observations and steps taken.

**Opportunity:** This section follows the same format as Test Notes, but covers unexpected opportunities that were uncovered during the test.

**Bugs:** What bugs were found during the exploratory test, followed by – Steps to Reproduce, Expected Results & Actual Results. Each bug is numbered.

**Issues:** Each numbered issue is described in detail.

## **About CommonTime**

We build intelligent messaging systems that create better communications for organisations around the world. Clients use our secure communication tools every day to make informed decisions in response to critical events.

We are passionate about empowering organisations with intelligent communications that help consistently deliver the best possible service to their customers.

W: [www.commontime.com](http://www.commontime.com) | E: [sales@commontime.com](mailto:sales@commontime.com) | T: 0845 009 0028