Enhancement to HMRC’s Check Employment Status for Tax (CEST): Test Plan and Approach

HMRC’s Check Employment Status for Tax (CEST) was developed under Agile.

In Agile, all code is fully integrated and tested at all times. Whenever there are new features, all the existing and new tests are run automatically and have to be passed before the new feature is merged. If the test fails, the development team is notified and the issue will be fixed immediately, preventing any bug from growing into a fatal error.

Sprint Reviews

Sprint Reviews are done fortnightly. The sprint review is an informal meeting which the development team, the scrum master, the product owner and the external stakeholders will attend.

Sprint Review	Regular Attendees	Ad hoc attendees
CEST (Check Employment Status for Tax)	Status Policy team Project team Solicitors Office Policy Technical Team Large Business Compliance Employer Helpline team Live Service DSM Wealthy & Mid Size Compliance team	Operational Excellence Standards & Assurance Lead Project manager

The team gives a demo on the product and will determine what has been finished and what hasn’t.

The purpose of the Sprint Review meeting is for the team to show the stakeholders the work they have accomplished over the sprint, in particular the outcomes of the user testing and compare it to the commitment given at the beginning of the sprint.

It is also used to collaborate on the next things that could be done to optimise value.

It is an extremely valuable way of ensuring that the development team and the stakeholders remain aligned. It enables concerns to be aired early.

We have been very fortunate to have had well attended sprint reviews. Views are exchanged and feedback given.

Transparency

- A wide group of stakeholders could access our prototype environment
- All stakeholders can access our staging environment (working software)
- All designs are accessible for all users for comment

Sign off for Tax Technical Accuracy

The Technical Policy team were embedded into the project to ensure the design was technically accurate.

User Research Usability Testing

The aim of the user research was to:

- test the developing service with likely users to make sure it meets their needs
- understand and resolve usability issues

Research was done with a broad range of users, including:

- those with limited digital access and confidence
- people with a range of visual, hearing, motor and cognitive impairments
- people who use assistive technologies like screen readers or speech recognition software

During our research we:

- did face-to-face and remote usability tests to find usability and accessibility issues
- commissioning an accessibility audit to uncover accessibility issues and get recommended fixes
- reviewed web analytics and back-office data to measure service performance
- analysed support tickets to identify problems users have with your service
- using surveys or follow-up interviews to collect detailed feedback from service users

From these activities we learned:

- more about how different kinds of users experience our services
- the usability and accessibility issues we needed to fix
- ways to improve our service
- A key findings report was produced on a fortnightly basis. This was shared in the sprint reviews and published on the MS Teams site.

The whole team is engaged in the research. The team have carried out a significant amount of user research, and a total of 64 users took part.

Research, Accessibility & Design Assurance

Each digital service is built to the WCAG 2.1 AA standard. The accessibility statement is part of the CEST, and the link is at the bottom of every CEST screen.

CEST is not a transactional service and for such services the Cabinet office/Gov Digital office devolves the assurance against the gov. Digital standards to the individual departments.

HMRC provides additional levels of assurance using Research, Accessibility and Design reviews conducted by experts in these areas. The aim of these reviews is to share best practice, build a picture of what is happening in a given service, as well as understand how our inclusive research, and design approaches are developing accessible services across HMRC.

Experts in each field review the research, accessibility and design decisions that have been made in that product. They check the quality and give constructive feedback to the teams. They want to understand how research has been used to inform the design of the service.

They consider the following:

- who the users are and their needs, and what research has been done to understand this
- accessibility needs and accessible design response
- what the service is, for example where it starts and ends
- content, interaction design, use of design patterns and user journeys
- design and research process

Each page of CEST links to an accessibility statement confirming that the service is fully compliant with the Web Content Accessibility Guidelines version 2.1 AA standard.

Off Payroll Technical IT Test Plan and Approach

Introduction

This document is owned by the Off payroll / CEST project team and outlines the Test plan, and approach taken by the team in the testing of the CEST product.

The CEST (Check employment status for tax) tool allows users (the engager, the worker or an agency) to determine if an engagement is an employment (and if an intermediary is present inside the Off Payroll Working Rules) or is a self-employment (and if an intermediary is present outside the Off Payroll Working Rules).

Employment status can be complex and where there is insufficient information provided by the user an Undetermined result is given. The user is provided with guidance and a support help line to progress the determination.

When a result is recorded, this can be printed or downloaded in a PDF format.

Scope

The scope of testing is the 2 micro-services that have been created:

- The frontend micro-service that the user uses to provide answers
- The backend micro-service (the decision service that contains the business rules) which provides the result

Not in scope is the external services of PDF generator and assets frontend.

The PDF generator and assets frontend will already be tested by the relevant development team.

Assumptions

The following is a list of assumptions which are specific to this project and may be revised as information becomes available:

- There will be test scenarios provided that align with the live cases as mentioned by HMRC FOI request on what scenarios was tested
- There is no E2E Scenarios with external Services. All Tests will be contained within the developed service
- The business rules will be designed and agreed by HMRC tax specialists

Risks

The following risks have been identified and have been impact assessed and appropriate mitigation put in place.

	Risk	**Impact**	**Mitigation**
1	PDF generator will not be able to provide the pdf	Users of the tool will not be able to store the outcome of the CEST tool	The PDF generator frontend can be started in local and staging environments, that can be manually tested to ensure the PDF is created.
2	The business rules weighting will not be correct	Users of the tool will get a determination that HMRC tax specialists do not agree with.	The business rules will go through several iterations of internal testing. They will be signed off by HMRC tax specialists. This is controlled by IPD Technical Policy team.

Test Approach

Test in Development

Unit Test -

Unit testing is a process of testing the smallest testable code unit possible e.g. if a webpage is developed, there would be a test on the page (and its contents) its view, and its controller.

These are created within the Microservice. When the microservice is sent for a build all the unit tests are run each time. This includes the tests for the code that has changed and for the ones that have not changed. Therefore, the regression testing is built into the run.

The process of code development is that the developer creates the test first, then creates the code to make the test pass.

This is called test driven development. These tests don't depend on any external sources.

Integration Test -

Integration Tests (IT) are also created within the Microservice test pack. These test the in and out routes of the service:

- For the frontend, this will test the web pages are correctly rendered
- For the backend (Decision Service), this will test for certain combination of answers the correct result is given (as per the business rules)

For each build these tests are also ran independently of other services, if they require external services they will be mocked out.

Acceptance Test -

Acceptance tests are created to test the entire service under scope i.e. across the Frontend and Decision service.

These tests ensure that any areas with the Unit Test and Integration Test that have not been tested are tested. Focusing on the flow through the user interface and achieving the correct outcome for the selected answers.

The tests are executed in its own test pack, external to the Microservices. They are written in Behaviour Driven Development Language Cucumber (a software tool used by computer programmers that supports behaviour-driven development). This allows the tests to be easily read and maintained.

Selenium (web automation framework that allows you to execute your tests against different browsers, not just Firefox, Chrome) is used within the step definitions of the underlying code base of the tests. This provides a framework for the tests to interact with Web browsers.

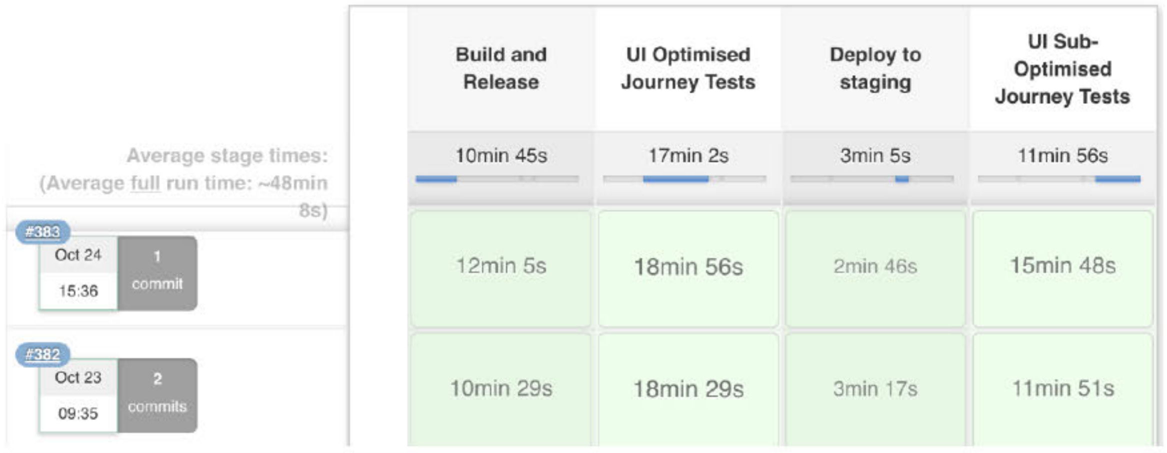
Build/Pipeline

Each time a build is created it is run into the build pipeline for CEST, following the continuous integration process.

The build and release stage build the service, executing the Unit Testing and Integration Testing. Only once all are passed is it released.

Once released the Acceptance test pack is executed to ensure the new build still passes the acceptance tests, once passed it deploys to staging so performance test and adhoc tests can be carried out.

The last stage is the acceptance test pack but against the current production screens to ensure that the build has not changed anything on the existing service.



Average stage times: (Average full run time: ~48min 8s)	Build and Release	UI Optimised Journey Tests	Deploy to staging	UI Sub-Optimised Journey Tests
	10min 45s	17min 2s	3min 5s	11min 56s
#383 Oct 24 15:36 1 commit	12min 5s	18min 56s	2min 46s	15min 48s
#382 Oct 23 09:35 2 commits	10min 29s	18min 29s	3min 17s	11min 51s

Performance Testing

- Performance testing is to ensure the service is able to work correctly while under load
- Gatling is the tool is being used to for performance Testing
- CEST Performance Test Results holds more detail of the testing carried out for the service (available on request)

The latest build was run at 300%. This is currently at 250 request per second, which the Service handles without errors. There is a note about the CPU usage, therefore if we started to see these levels, we can increase the normal of instances to decrease the load.

Other Test Types

Content Testing

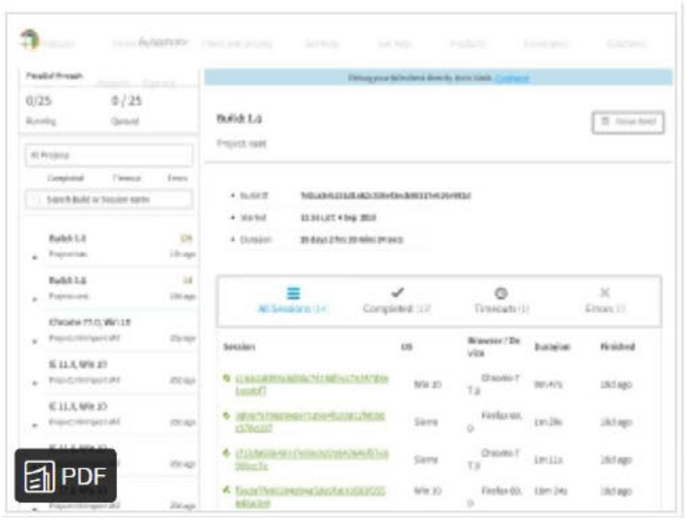
- Content of the whole website will be confirmed with the CD (content designers) and compared with the Prototype
- Content changes in the frontend will be auto compared using Unit Tests

Accessibility Testing

- Web accessibility testing is about ensuring the service is accessible by all users
- For the automation of accessibility testing we use the Wave app and aXe tool
- We can also use relevant software (screen reader, non-mouse navigation, screen magnifier) to test the service, as a user might
- We will also involve the internal accessibility advocate team, who will pre check the service before going to external testers
- It will also go to DAC (disability assessment centre) for external accessibility testing

The two issues identified were fixed on the 18 November and signed off by the accessibility team.

Emulation Testing



- Emulation Testing ensures the service works with the different Operating services and Browsers, through the use of emulation software
- Browserstack is used to the Emulation Testing

Device Testing

- Device testing ensures the service works on the different devices and looks correct on the screen of the device
- Android and iPhone/iPad are the main devices that will be tested against. These tests will be carried out using emulators and physical devices

Security vulnerability Testing

- Security Vulnerability testing ensures the services has no vulnerabilities that can be taken advantage of by malicious intent
- Internally the testing will be carried out automated using the OWASP ZAP tool
- Once ready the tool will be tested by security vulnerability experts of external testers

The report is available to those with a Secret security clearance.

Multi-Lingual Testing

- Multi-Lingual testing ensures the service has several Languages and the content is correct
- The languages under test are English and Welsh
- There will be Unit Tests that ensure there is no missing Welsh content
- Screenshots will be created in both English and Welsh. These are sent to the Welsh Language Unit to quality assure

Exploratory Testing

- Any Kind of Exploratory Testing is conducted in Manual Testing fashion
- Adhoc Testing: If we do not have sufficient detailed software specification, we can go for adhoc testing

Adhoc testing is a term commonly used for the tests carried out without planning the software and documentation. The tests are supposed to be executed only once, unless a defect is discovered.

The benefits at a glance:

1. No planning and documentation needed
2. The important bugs are found quickly
3. Easy to start and implement

Checking the browser compatibility (look and feel) by logging to the different in scope browser versions.

User Needs and Usability Testing

- User research testing will be carried out using user research sessions and the user's input
- Simulate the real time users experience and test the real time scenarios and Test the application for end to end Scenarios.
- We will take/provide input from/to the user research sessions and make/learn the real time ways.