Attachment 03

Rules-as-Code Service Delivery Methodology

Table of Contents

RaC Requirements discovery

During the discovery phase, our business analyst will work with the relevant agency to fully understand the requirements and how the RaC approach can be adapted for the personalisation user journey.

Workshops will ensure all necessary requirements are captured, understood and documented. We'll start with ensuring a shared agreement on what policy, legislation, rules or benefits are to be converted to rules as code. We'll then move onto the actual rules analysis.

Of course, there are many different methods you could use to turn existing rules/legislation into code. Salsa is currently working with Pia Andrews and a team of volunteers on the rules as code project for GovZero Aotearoa, and our approach for GovCMS clients will track the established process we've been following for the GovZero project.

GovZero Aotearoa is an open collaboration initiative with three projects underway for 2022. Salsa is contributing to the project by supplying three team members to work on the rules as code (RaC) project. We're codifying New Zealand's Social Security Act 2018 to make it easier for people in NZ to see if they're eligible for social security benefits.

Salsa is currently providing three team members to the RaC GovZero project, 8 hours per person, per week for 2022. This is part of our commitment to open government and innovative ways to deliver better services for citizens.

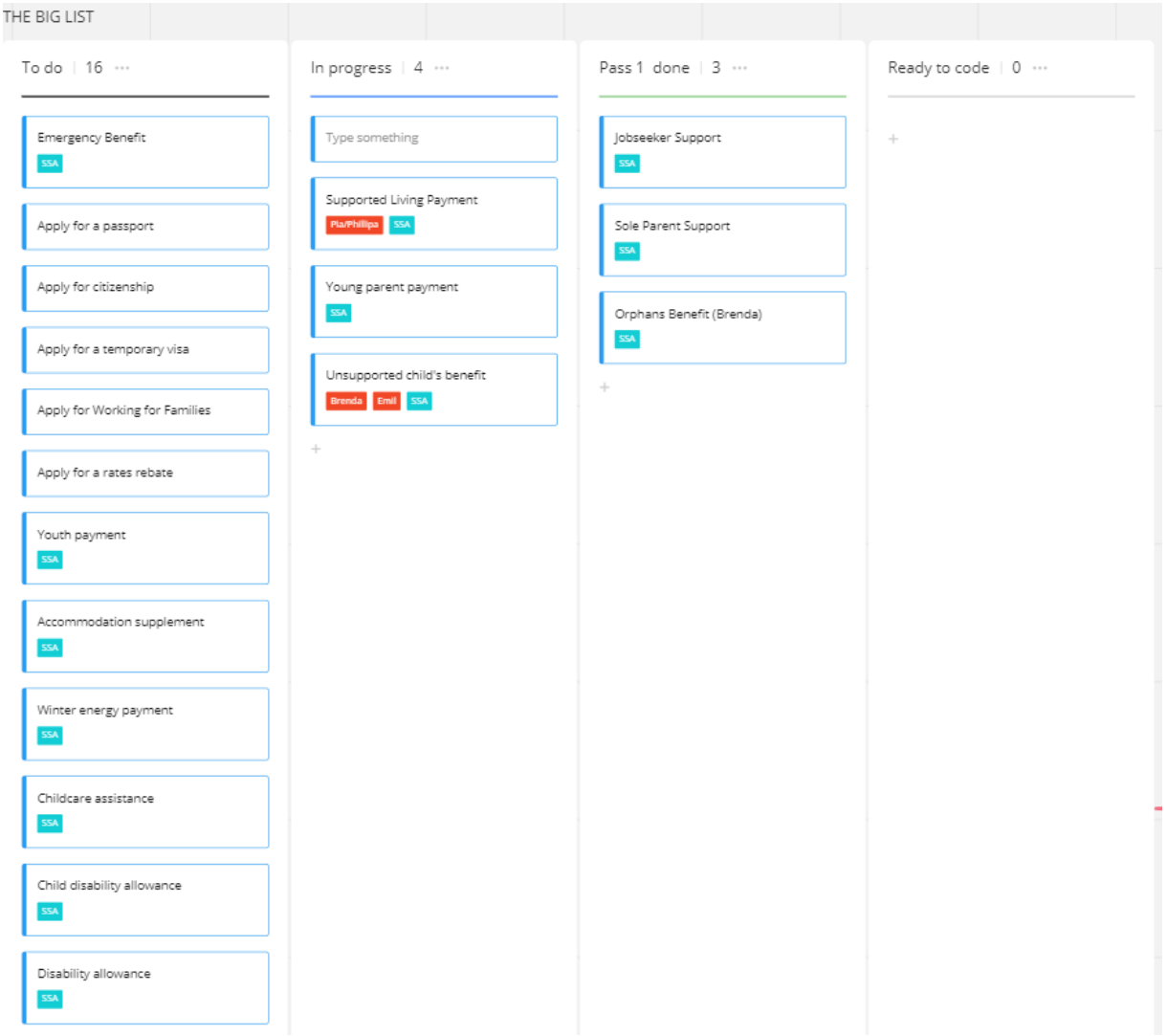
Starting with Miro

Pia and her team have set up a great system using a Miro board to analyse the relevant rules/legislation. The Miro board is the source of truth during the planning stages. Miro is also an excellent collaboration tool that allows multiple people to work and view the legislative/rules analysis.

For GovCMS projects using RaC for content personalisation we'd follow the same approach, setting up a Miro board for each project. The Board is then divided into sections. The sections are:

- Project planning (see screenshot below) - a section that tracks the progress for each section of the rules/legislation so we can easily see project progress and when specific areas of legislation/rules are ready for coding
- A section for global variables and definitions, with all global variables and definitions added in this section as the legislation is explored and documented.
- A section for the section of the act/legislation that pulls out the key information/rules
- Sections for payment schedules for each area of the act/legislation (if applicable)
- Sections with test cases

Below is a screenshot for the planning part of the Miro board.



During discovery, we'd follow a two-stage process with two 'passes' for the legislation/rules. Again, this is mirroring the process we've been following on GovZero.

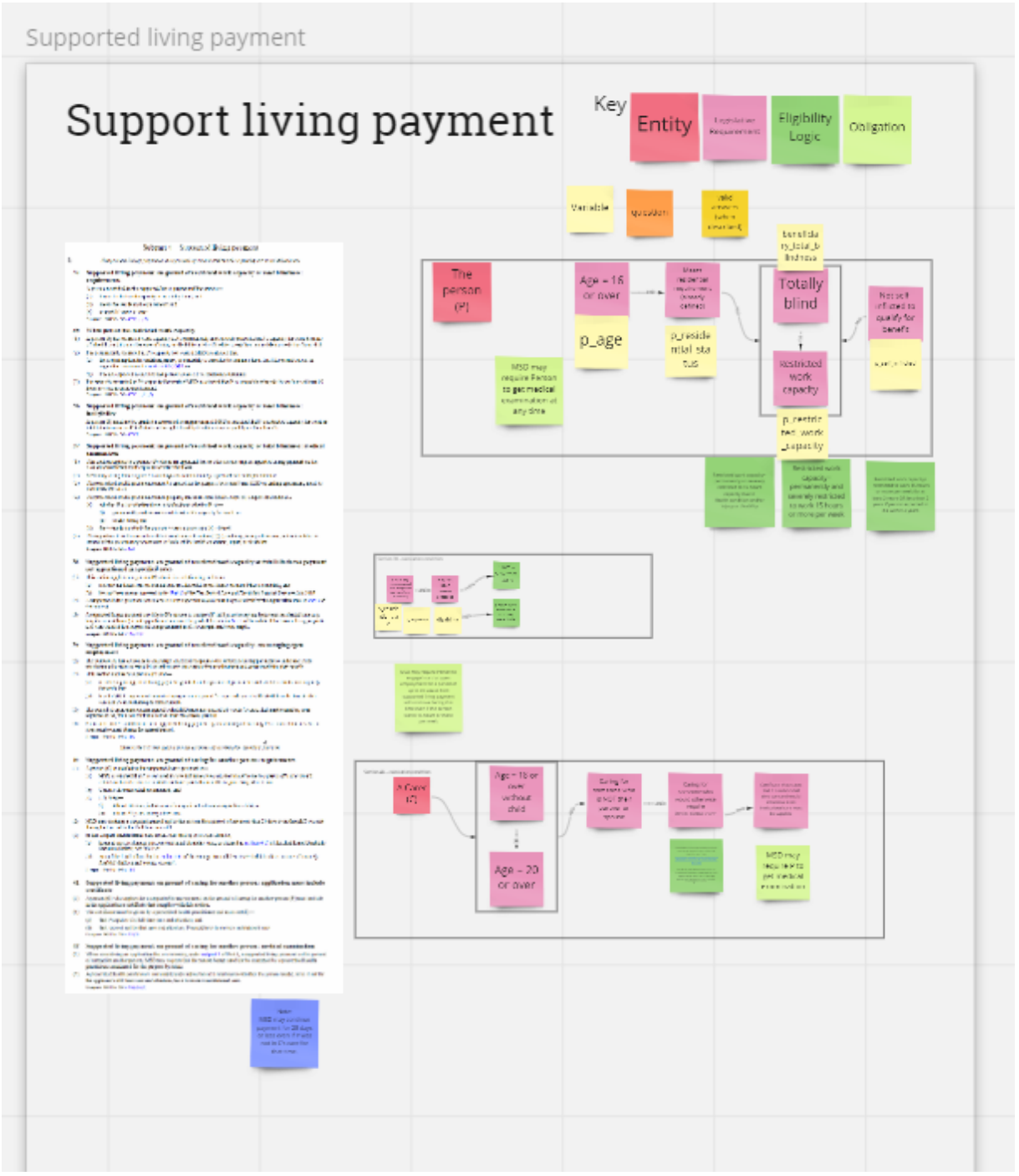
The first pass — analysing the legislation/rules

In the first pass the goal is to identify:

- Eligibility criteria
- Relevant entities
- Eligibility logic
- Calculation logic
- Other key variables relevant to each specific project/legislation

The Miro board ‘frames’ for each benefit/area include screenshots of the relevant section from the legislation/rules on the left for easy access to the relevant section of legislation. The typically dense legal language is then broken down into simpler rules/statements using Miro’s Notes, which are colour-coded to help visually differentiate different parameters.

Below is an example from one of the benefits we worked on for GovZero, the supported living payment. This payment is designed for people who are totally blind or have restricted work capacity. It also covers carers looking after someone who would otherwise need institutional care.



The process involved reading through the legislation/rules and pulling out eligibility criteria, including definitions if necessary.

For example, section 34:

Supported living payment: on ground of restricted work capacity or total blindness: requirements

A person is entitled to the supported living payment if the person—

(a) has restricted work capacity or is totally blind; and

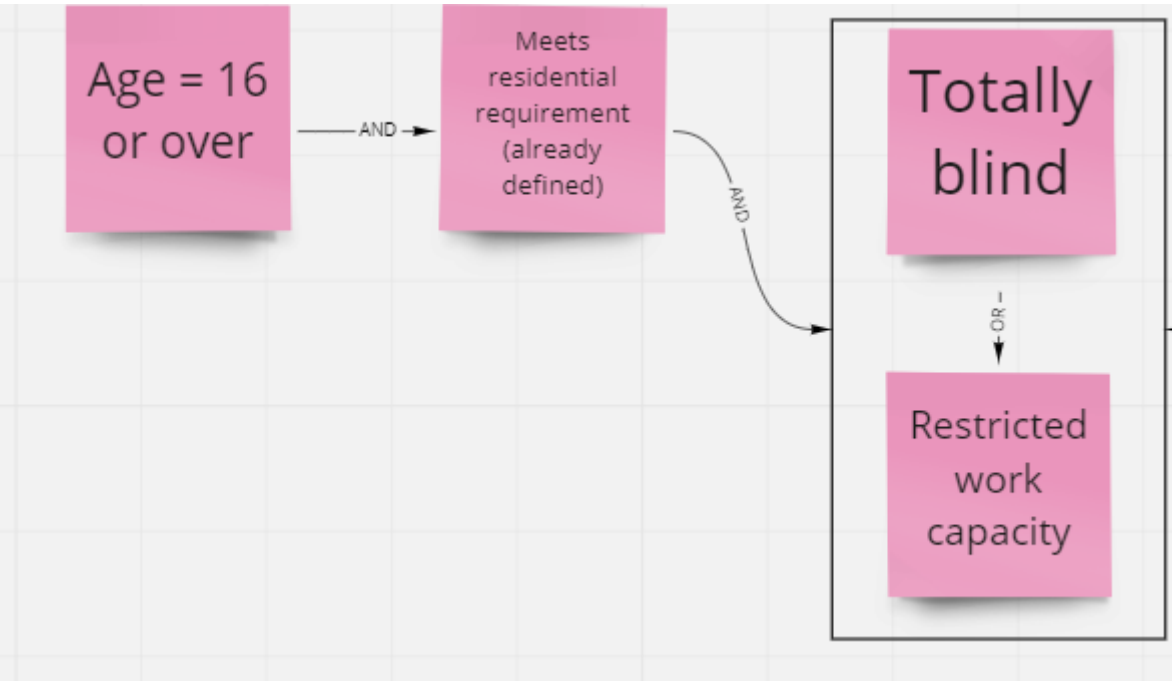
(b) meets the residential requirement; and

(c) is aged 16 years or over.

In our first pass, this translates to the following sticky notes on the Miro board:

- The person (P)
- Age = 16 or over
- Meets residential requirement
- Totally blind OR Restricted work capacity

We use AND/OR connectors as required.



The frame also includes any relevant definitions (e.g. the definition of 'restricted work capacity' and 'residential requirement'). If definitions are across more than one section of the legislation/rules, they can be set up in the global variables and definitions section and then copied and pasted into the relevant sections of the Miro board.

Calculation logic

The first pass also looks at calculation logic and payment rates (if applicable). To track the different payment types, we used a spreadsheet. As you can see from the screenshot below, the supported living payment is quite complex, with many variables determining the rate, plus the application of two different income tests.

supported_living_a	single (y), age (<18), dependent_child (n)	281.15	income_test_1
supported_living_b	single (y), age (>=18), dependent_child (n)	336.65	income_test_1
supported_living_c	single (y), dependent_child (y)	455.58	income_test_1
supported_living_d_i	single (n), partner_on_main_benefit (y), dependent_child (n)	275.27	income_test_2
supported_living_d_ii	single (n), partner_on_main_benefit (y),dependent_child (y)	289.12	income_test_2
supported_living_e_i	single (n), partner_on_super (y) OR partner_on_vets (y), dependent_child (n)	275.27	income_test_2
supported_living_e_ii	single (n), partner_on_super (y) OR partner_on_vets (y), dependent_child (y)	302.96	income_test_2
supported_living_g_i	single (n), partner_on_super (N) AND partner_on_vets (n) AND partner_on_main_benefit (n), restricted_wor	550.54	income_test_1
supported_living_g_ii	single (n), partner_on_super (N) AND partner_on_vets (n) AND partner_on_main_benefit (n), restricted_wor	578.24	income_test_1
supported_living_h_i	carer (y), single (n), partner_on_super (N) AND partner_on_vets (n) AND partner_on_main_benefit (n), restr	275.27	income_test_1
supported_living_h_ii	carer (y), single (n), partner_on_super (N) AND partner_on_vets (n) AND partner_on_main_benefit (n), restr	302.96	income_test_1

Income test calculations were set up on the Miro board, for example:

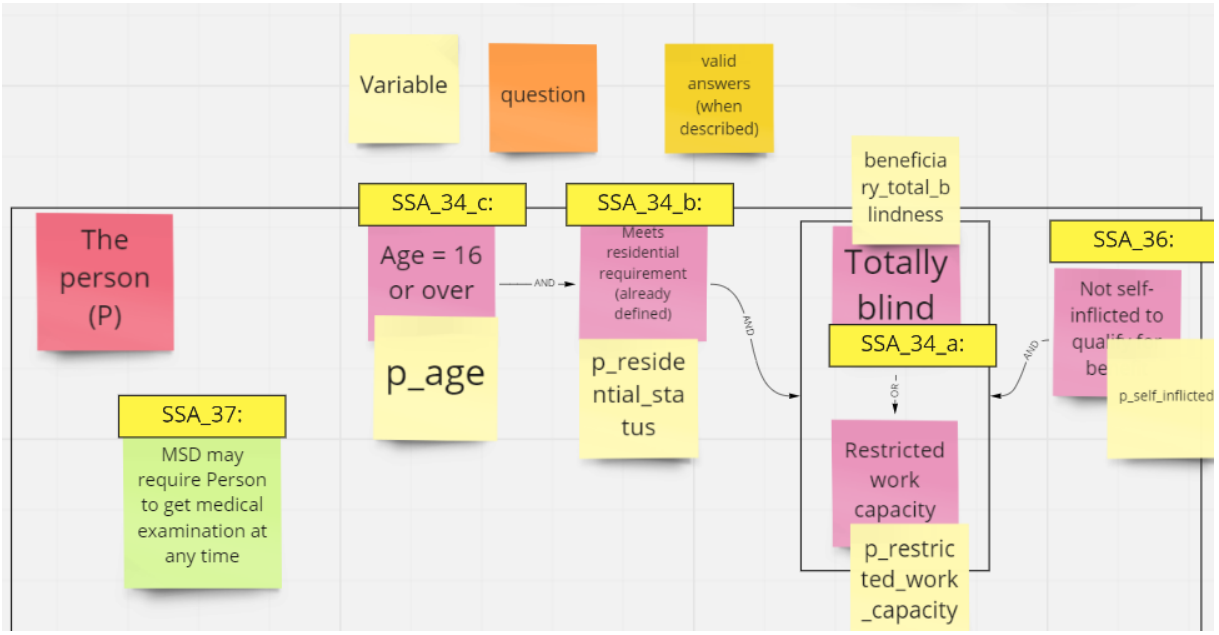
- If income_total_weekly<160 then benefit = <rate> (regardless of income test)
- if 250 > income_total_weekly > 160 then if income_test_1, benefit = <rate> - ((income_total_weekly - 160) * 0.15)
- if income_total_weekly > 250 then if income_test_1, benefit = <rate> - ((income_total_weekly - 250) * 0.35) - ((250-160) * 0.15)

The second pass — user variables and test cases

The second pass looks at translating entities, legislative requirements and eligibility criteria into user variables. For example:

- p_age
- p_residential_status
- beneficiary_total_blindness
- P_restricted_work_capacity

Below is a screenshot of Miro mapping after the second pass.



Note we've also added in the relevant sections from the Act, so that specific variables can be quickly and easily traced back to the section of the Act.

The **test cases** look at both eligibility criteria and rate calculations. For example:

- P=17yro
- Meets residential requirement
- Is totally blind
- And not self-inflicted
- Is employed
- Earns $20K/year (384.61 per week)
- Is single

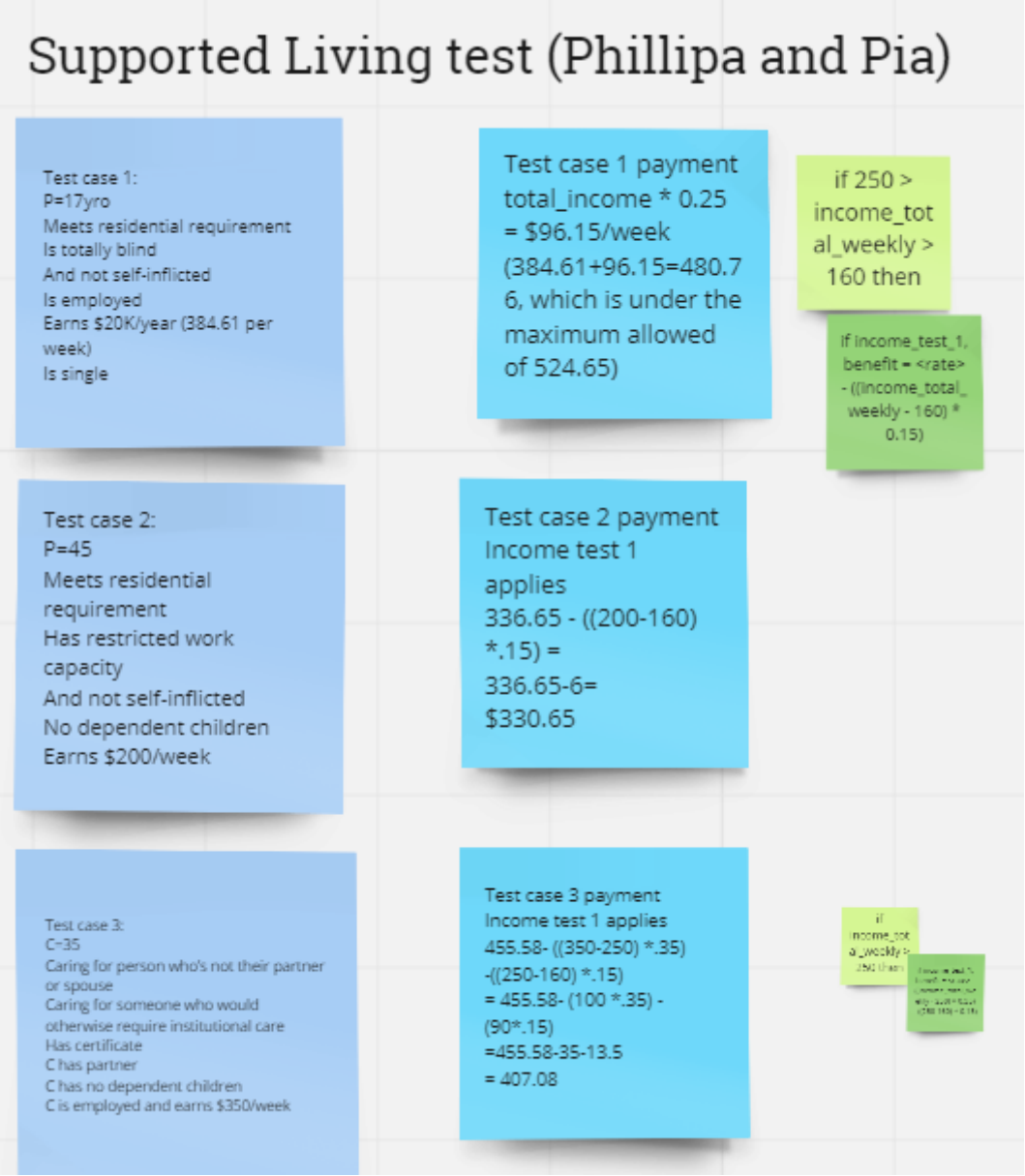
Using this test case, we then worked out the payment.

Payment:

- total_income * 0.25 = $96.15/week (384.61+96.15=480.76, which is under the maximum allowed of 524.65)

This scenario and payment will be used to set up the code, using test-driven development (TDD).

The screenshot below shows the test cases.



Global variables and definitions

Global variables and definitions can be coded once across multiple legislation or parts of legislation. For example, some global variables from the GovZero RaC project include:

- Definition of totally blind
- Definition of restricted work capacity
- Residency requirement
- Definition of single
- Definition of dependent children
- Definition of assets for the assets test

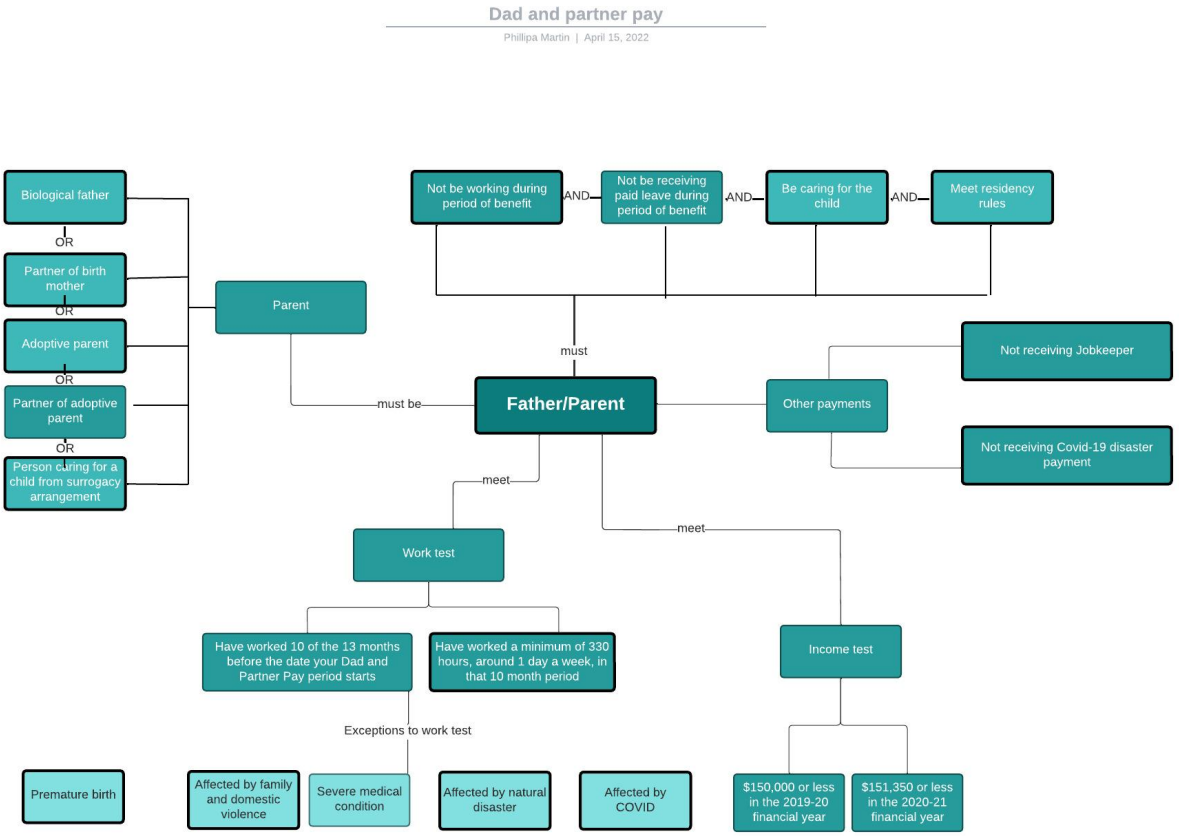
Applying the Miro process to GovCMS projects

The above methodology can be re-used for any project based on a RaC framework, including personalisation journeys. In the case of GovCMS projects, we'd create one Miro board per project, with all sections of the legislation/rules divided into frames for analysis. The Miro board would include screenshots of the relevant sections of the legislation/rules, with sticky notes used to map the legislation/rules.

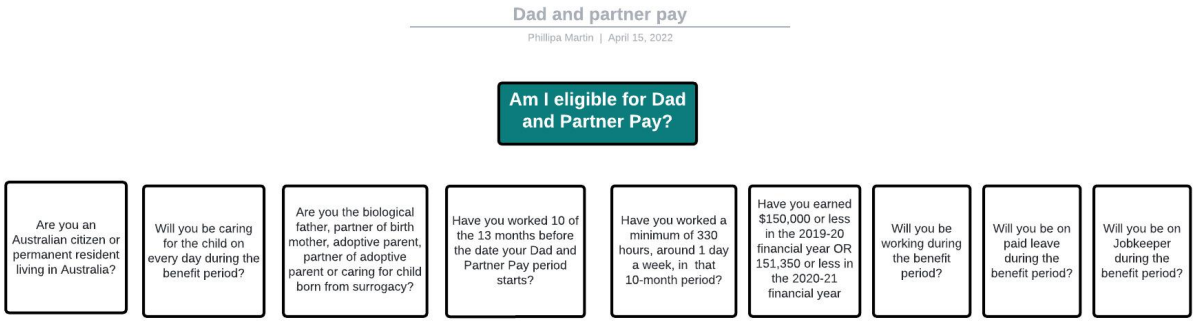
From Miro to concept models and decision models

The next stage in the process is to develop **concept models**. Concept models pull out the concepts and variables identified in the legislation/rules analysis (and notated in Miro) and turn them into a diagram.

To demonstrate this stage, let's consider the rules for our proposed PoC, Dad and Partner Pay (see https://www.servicesaustralia.gov.au/dad-and-partner-pay). Below is a simplified version that takes into account some of the eligibility parameters to provide an indicative view of a concept model.



With our concept models developed, we'd then move onto decision models. Decision models cover the key **questions** for eligibility. Again, below is a partial representation of the decision model for the Dad and Partner Pay.



Rules statements

The final stage before coding is creating rules statements. Below is an example of rules statements.

A claimant is eligible for Dad and Partner payment if all of the following are true:

- The claimant is an Australian citizen or permanent resident living in Australia.
- The claimant will be caring for the child on every day during the benefit period.
- The claimant is the biological father, partner of birth mother, adoptive parent, partner of adoptive parent or caring for a child born from surrogacy.
- The claimant has worked 10 of the 13 months before the date their Dad and Partner Pay period starts.
- The claimant has worked a minimum of 330 hours, around 1 day a week, in that 10-month period.
- The claimant has earned $150,000 or less in the 2019-20 financial year OR 151,350 or less in the 2020-21 financial year
- The claimant will NOT be working during the benefit period.
- The claimant will NOT be on paid leave during the benefit period
- The claimant will NOT be receiving Jobkeeper during the benefit period.
- The claimant will NOT be receiving the COVID-19 Disaster Payment during the benefit period.

Test-driven development

Test-driven development uses the test cases to develop code. Developers will code in OpenFisca based on the test cases outlined in the Miro board, and drawing on the decision models and rules statements. Code can be set up to 'pass' each test, and new test cases can be added to address different scenarios.

The role of user stories

In some cases, user stories can also be written to expose the decision models and the content flow within the Drupal webform. Below is an (incomplete) example based on GovZero Supported Living Payment. It provides an indicative, traditional user-story ticket that coders can use to help guide them when building the webforms and OpenFisca logic.

Title: Eligibility: Supported living payment for citizens

Acceptance criteria

GIVEN I am a site visitor

WHEN I visit the eligibility checker page

THEN I will be presented with the following initial questions:

- How old are you?
    - 15 and under (THEN ineligible for supported_living_payment)
    - 16 or 17 (p_age=16_17)
    - 18 and over (p_age=/>18)
- Are you a NZ citizen?
    - If yes (p_residential_requirement=yes)
    - IF no:
        - Do you have residence class visa AND ordinarily reside in NZ AND have resided in NZ for 2+ years since becoming a citizen or resident?
            - If yes (p_residential_requirement=yes)
            - If no:
            - Are you considered a refugee or protected person under the Immigration Act?
                - If yes (p_residential_requirement=yes)
                - If no (p_residential_requirement=no) THEN ineligible for supported_living_payment
- Are you unable to work more than 15 hours per week due to a disability, injury or health condition?
    - If yes, p_restricted_work_capacity=yes
    - If no p_restricted_work_capacity=no
- Are you totally blind?
    - If yes totally_blind=yes
    - If no totally_blind=no

- If p_restricted_work_capacity=no AND totally_blind=no THEN ineligible for supported_living_payment

If all eligibility requirements are met for the supported living payment (and/or other benefits), then the user will be presented with questions to determine the rate of payment they'd receive. Questions would include questions like:

- Are you single?
- If no:
    - Is your partner receiving a main benefit?
    - Is your partner receiving New Zealand superannuation or a veteran's pension?
- Do you have any dependent children?
- Are you currently employed?
    - If yes: What is your total weekly income? <Income ranges supplied, in line with income tests, e.g.>
        - Less than $160
        - Between $160 and $250
        - More than $250
        - More than $xxx (income when the person is ineligible due to income test threshold)