CSF100-104 Software Models

Contents

1 IDD – Insurance Distribution Directive - Introduction 2
1.1 Background 2
1.2 Description of the IDD project 2
1.2.1 What Changed 3
1.2.2 The Stakeholders 5
1.2.3 The Target Platform 6
1.2.4 The Development Team 6
2 First Chosen Model 7
2.1 The Agile Process 7
2.1.1 Description of the Agile Process 7
2.1.2 The Stages of the Agile Process 8
2.2 The Scrum Framework 10
2.2.1 Definition of Scrum 10
2.2.2 Scrum Phases and Practices 10
2.3 The Scrum Methodology applied to the IDD Project 11
2.3.1 Planning the IDD Project 11
2.3.2 The Sprint 12
3 The Second Chosen Model 13
3.1 The Waterfall Model 13
3.1.1 Description of the Waterfall Model 13
3.1.2 The Stages of the Waterfall Model 13
3.2 The Waterfall Model applied to the IDD project 15
3.2.1 Project Management 15
3.2.2 Development Team 15
3.2.3 The Design 15
4 Agile Model vs Waterfall Model 17
4.1 Advantages of the Agile Model 17
4.2 Disadvantages of the Agile Model 17
4.3 Advantages of the Waterfall Model 18
4.4 Disadvantages of the Waterfall Model 18
4.5 What is best for the IDD? 18
5 References 19

1 IDD – Insurance Distribution Directive - Introduction

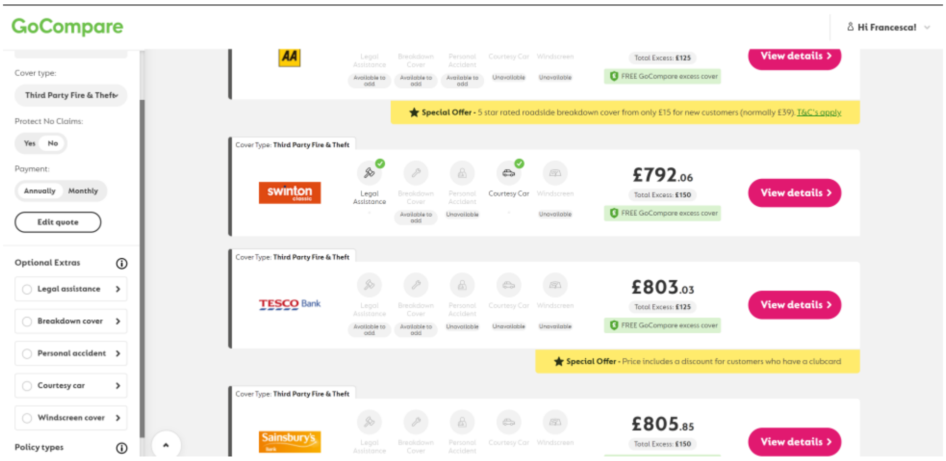
1.1 Background

The insurance Distribution Directive is an EU legislation which sets new regulatory requirements for firms designing and selling insurance products [1]. It has been active since October 2018 and it enhances the responsibility of the insurance industry to actively identify customer needs and only offer products which meet them. The implementation of features which ensure the compliance to the new legislation is an addition to the Treating the Customer Fairly mandate, which stimulates insurance companies and comparison websites to act in the customer's best interest at all times.

In May 2019, GoCompare has started a review of their website in order to guarantee the compliance to the new legislation and have highlighted areas in which architectural change was necessary. The project was successfully concluded in April 2020, although parts of the code were released within the range of one year (since the beginning of the project in 2019).

1.2 Description of the IDD project

The IDD project initially regarded the results page of the Car Insurance Journey on the Gocompare.com insurance comparison website (Figure 1). The IDD project consisted in the implementation of the Wizard, a tool which opens a series of questions related to the 5 add-ons (Figure 2). The IDD project also involved other parts of the Car Journey as shown below (3).



Cover type: Third Party Fire & Theft

Protect No Claims: Yes No

Payment: Annually Monthly

Cover Type: Third Party Fire & Theft

Total Excess: £125

★ Special Offer - 5 star rated roadside breakdown cover from only £15 for new customers (normally £39). T&C's apply

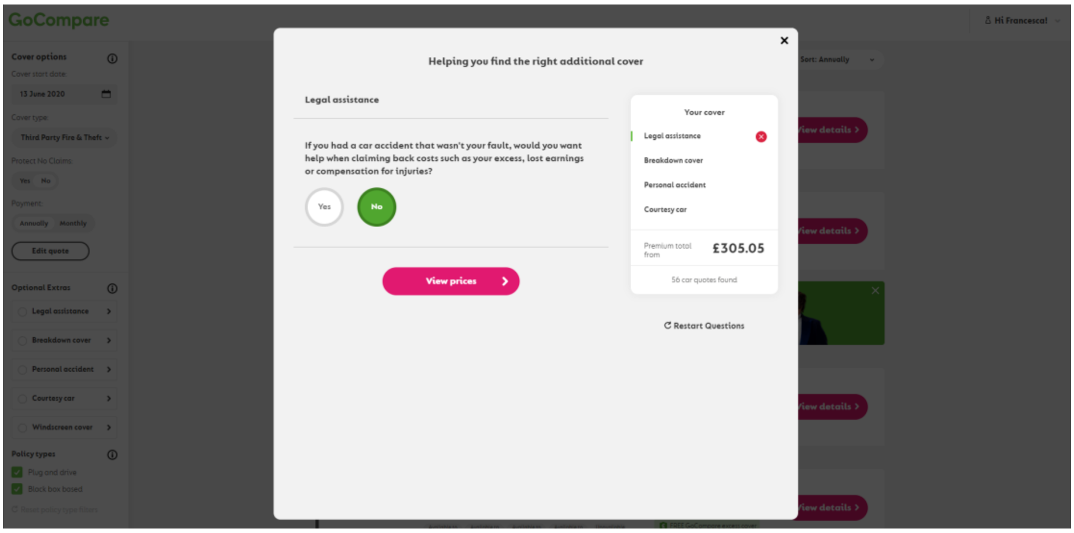
£792.06
Total Excess: £150

£803.03
Total Excess: £125

★ Special Offer - Price includes a discount for customers who have a clubcard

£805.85
Total Excess: £150

Figure 1 - Gocompare.com car journey results page.



Helping you find the right additional cover

Legal assistance

If you had a car accident that wasn't your fault, would you want help when claiming back costs such as your excess, lost earnings or compensation for injuries?

○ Yes ● No

Your cover

- Legal assistance
- Breakdown cover
- Personal accident
- Courtesy car

Premium total from £305.05

56 car quotes found

Cover options

Cover start date: 13 June 2020

Cover type: Third Party Fire & Theft

Protect No Claims: Yes No

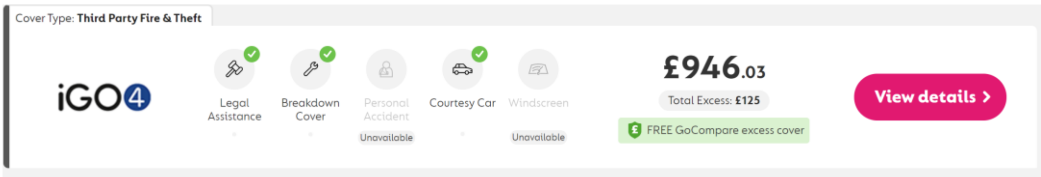
Payment: Annually Monthly

Sort: Annually

Figure 2- The Wizard is a Modal that pops-up when the customer clicks on any of the add-ons listed in the side bar on the left of the results page

Before the start of IDD regulation, the customers had to complete each question regardless of the relevance of the product that they were offered. The IDD project is aimed at offering the customer a tailored experience, avoiding confusion and unnecessary overload of information.

After completing the journey, customers are redirected on the results page, where they are provided with the matching prices from different partners, ordered from lowest to highest. The IDD project's initial focus was to provide complete and transparent information on the Add-ons that the customer can decide to add to each insurance policy (Figure 3). Customers can now choose the questions that they want to answer to in the Wizard.



Cover Type: Third Party Fire & Theft

Legal Assistance	Breakdown Cover	Personal Accident	Courtesy Car	Windscreen
✓	✓	Unavailable	✓	Unavailable

£946.03

Total Excess: £125

FREE GoCompare excess cover

Figure 3- available add-ons are displayed for each result

1.2.1 What Changed

The Page 4 of the Car Journey was updated with new content for the Cover Type options 'Comprehensive', 'Third Party Fire and Theft' and 'Third Party Only' to be more coherent. A new Modal is opened when the customer clicks on the question mark on the right side of the text which provides more extensive information (Figure 4).

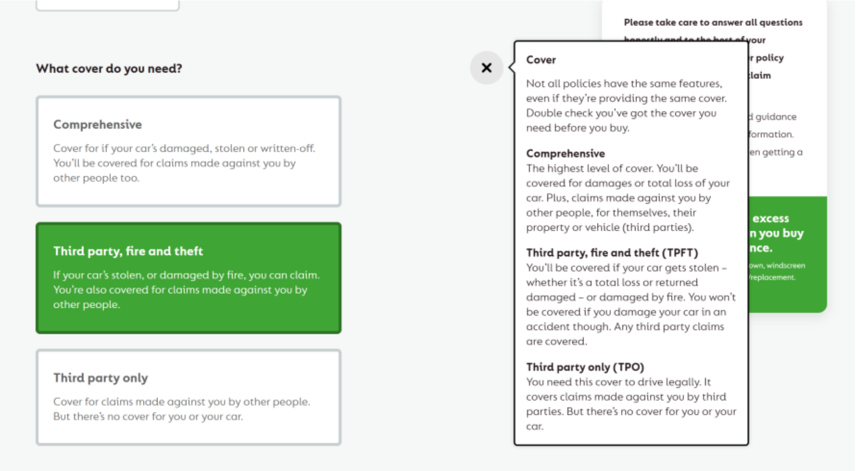
The results page side bar has been provided with clickable elements for the add-ons: Breakdown, Personal Accident, Courtesy Car and Legal Cover (as shown above in Figure 1). The links open an Optional Extra Modal containing additional help text to support the customers (Figure 5).

When the customer lands on the results page the costs of the Add-ons are no longer displayed. The Page will display a variety of texts depending on the date and level of cover chosen. To see the price, the user will be required to interact with the Wizard tool. The Wizard is composed of a series of questions that lets us drill down and assert the exact level of cover that the user is after. Each Add-on has its own set of questions which are not linked to each other.

One of the other major changes of the IDD is also the source of the prices displayed to the customer. In the past these were pulled from the static data provided by the insurance partners. This would mean that the accuracy of the cost could not be guaranteed. With the IDD the prices are taken from Defaqto[1] and they are updated daily. This also means that the additional information about the Add-ons are specific for each partner, constantly updated, and provided by an independent third-party business.

The customer is now also provided with the opportunity to change the cover type from the results page without having to go back to the quote. If the customer changes the cover type new information about the implications of the changes are displayed in a modal.

The original project did not involve the structural changes of the Page 4, but during the process new issues were raised and new solutions were implemented. Market research, UAT and user testing revealed new business needs which extended the times of the IDD project to a full year of work.



What cover do you need?

Comprehensive
Cover for if your car's damaged, stolen or written-off. You'll be covered for claims made against you by other people too.

Third party, fire and theft
If your car's stolen, or damaged by fire, you can claim. You're also covered for claims made against you by other people.

Third party only
Cover for claims made against you by other people. But there's no cover for you or your car.

Cover
Not all policies have the same features, even if they're providing the same cover. Double check you've got the cover you need before you buy.

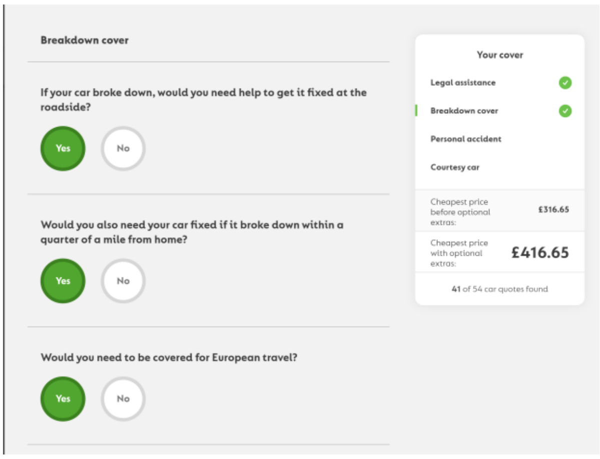
Comprehensive
The highest level of cover. You'll be covered for damages or total loss of your car. Plus, claims made against you by other people, for themselves, their property or vehicle (third parties).

Third party, fire and theft (TPFT)
You'll be covered if your car gets stolen – whether it's a total loss or returned damaged – or damaged by fire. You won't be covered if you damage your car in an accident though. Any third party claims are covered.

Third party only (TPO)
You need this cover to drive legally. It covers claims made against you by third parties. But there's no cover for you or your car.

Figure 4 - The Cover Types are explained extensively in the side modal that the customer can open by clicking on the question mark

[1] "Defaqto is an independent financial information business, focused on helping financial institutions and consumers make better informed decisions. From its financial product and fund database, Defaqto produces a range of services including ratings, software, data services, publications and events" From Wikipedia: https://en.wikipedia.org/wiki/Defaqto



Breakdown cover

If your car broke down, would you need help to get it fixed at the roadside?

● Yes ○ No

Would you also need your car fixed if it broke down within a quarter of a mile from home?

● Yes ○ No

Would you need to be covered for European travel?

● Yes ○ No

Your cover

Legal assistance ✓

Breakdown cover ✓

Personal accident

Courtesy car

Cheapest price before optional extras: £316.65

Cheapest price with optional extras: £416.65

41 of 54 car quotes found

Figure 5 - each add-on is provided with an individual optional modal where the customer can answer the relevant questions and have a preview of the total cost

1.2.2 The Stakeholders

The IDD project involved all departments of GoCompare and Goco Group functions. The external secondary stakeholders were the FCA – Financial Conduct Authority -, Defaqto and the totality of the car insurance partners.

1.2.2.1 Internal GoCompare Departments

The IDD project involved the totality of GoCompare internal teams that can be divided in four main areas of interest: strategy, technical implementation, finance and regulation, and business services.

1.2.2.1.1 Strategy – Product Team

The Product Team has the responsibility to implement the strategy, roadmap ad feature definitions for the product [2]. Product is also responsible for assessing the market needs which impact the structure of the project, the team size and structure, and influences the choice of the model of implementation. For example, timing and immediate availability of the releases can influence the team to choose an agile over a waterfall model. The Product team uses tools like market and competitive analysis to shape the product and works closely with both the Marketing and Sales Teams to understand customer behaviour and needs.

1.2.2.1.2 Strategy – Partner Development Managers (PDM)

The main stakeholders of the IDD project were the Partner Development Managers. The PDMs initiate new as well as follow existing partner relations to investigate the partner's needs and strategic goals. The PDMs' main responsibility was to monitor the IDD project's compliance to the partners' agreements and to raise any issue or new requests from the partners side. The PDMs active role was to mediate the business needs of the Car Insurance Partners with the developments and the needs of the GoCompare development team.

1.2.2.1.3 Technical Implementation – Tech Ops.

TechOps are responsible for running and maintaining systems that are already built. Their responsibilities include reporting, managing capacity, monitoring systems, and infrastructure planning. They also raise bugs and issues and monitor the new code releases. In the IDD project they overviewed releases through the different steps of the process.

1.2.2.1.4 Technical Implementation – Database Maintenance Team.

The IDD project involved changes in the GoCompare Database. The Database Maintenance team is responsible for the maintenance of the procedures, the planning and creation of new SQL tables and stored procedures. The Team also performs quality checks and testing.

1.2.2.1.5 Finance and regulation – Compliance

The Compliance department is responsible for ensuring that the product adheres to external rules and internal quality checks. The department also seeks to reduce system risk and financial crime. Within the IDD project they ensured that the final product would adhere to the IDD regulation by checking content quality, personal data usage, and promoting the Treating the Customer Fairly mandate.

1.2.2.1.6 Business Services – Customer Service Team.

The Customer Service Team is the first point of contact between the customers and the company. For the IDD project the CS Team worked closely with the Developers, reporting issues related to the new releases and customer experience.

1.2.2.2 Goco Group Functions

Goco Group plc is a data-led FinTech that operates GoCompare, Look After My Bills, weflip, MyVoucherCodes and Energylinx. The Goco Group Leadership Team is one of the stakeholders of the IDD project as they oversee any operation within the GoCompare organization. The Leadership Team is composed of the CEO of GoCompare, the CEO of AutoSave, a General Counsel and Company Secretary and the Group CTO.

1.2.3 The Target Platform

The target platform of the IDD project is the GoCompare.com insurance comparison web-services. The GoCompare website is hosted partially in-house servers and partially on the Azure cloud services. Acceptance testing and user research have been carried out across all the GoCompare Group. UAT workshops were used to test the core functionality of the new car insurance journey across multiple platforms and devices. The user research testing workshops were used to gather internal customer feedback to iterate and enhance our new car insurance journey before launch.

1.2.4 The Development Team

DevOps is the combination of cultural philosophies, practices and tools in which a team combines software development and infrastructure management processes [3]. The structure of the development team for the IDD project was based on the DevOps model.

The team selected for the IDD project was an existing development team and it was composed by 4 developers (of which 2 Seniors and 2 Software Engineers), 1 test engineer and 1 product manager. Within the Development team the product manager is responsible for creating the stories and assigning priority to the work.

The team is responsible for choosing the development model, write the code and implement testing. The model chosen was the **Agile Process**, and the management system chosen was **Kanban**. The use of the Kanban methodology involved the organization of the work by flow. The work is organised on the Kanban Board(Figure 6), which is divided in columns which represent the stages of each individual piece of work, called **user story**. For example, the columns could represent stages like backlog, planned, in progress, development, testing (or UAT), release, and done. This means that the team member picks up a story from the Kanban Board each time the precedent piece has reached the "done" column.

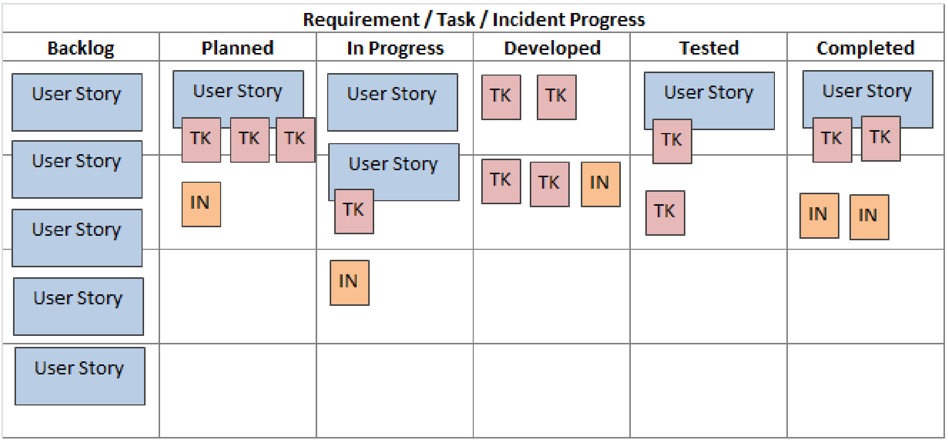


Figure 6 - Example of Kanban Board [4]

Whilst Kanban is not an **iterative** methodology(7), it is part of the Agile Process because it fulfils the twelve requirements of the Agile Manifesto and it is also incremental(7). In Kanban, the work is therefore not organised in iterations (or sprints), but consistent demos to the stakeholders were performed throughout the duration of the project before each release.

As the team used an Agile Process methodology, all team members could cover any of the roles within the team structure, allowing them to pair code and exchange testing responsibilities.

2 First Chosen Model

The first methodology we are going to analyse is the **Scrum Framework**. This framework is widely used within the OneTech department of GoCompare as part of the **Agile Software Development Process**.

In the next two sections we will introduce and describe the Agile Software Development Process, and subsequently we will discuss the Scrum Framework in detail.

2.1 The Agile Process

Agile is a methodology for modelling and documenting software systems based on best practices which are modelled and applied to software development projects in a flexible manner [5]. This model was first described in the year 2000 by a group of 17 developers in Oregon. The aim of the Agile Manifesto was to formalise a methodology that could reflect the market needs of tech companies in terms of flexibility, market speed and rapid feedback, especially they were aiming to create a process that could effectively react to change [6]. Some of the initial goals of the Agile Process were to enable face-to-face communication between team members and stakeholders instead of relying solely on documentation; to enable close collaboration between developers and business experts; to release work more frequently and to address the normal requirements problems.

2.1.1 Description of the Agile Process

The Agile Process is a combination of **iterative development**, and **incremental development** and **delivery**. Agile is a process that focuses on the adaptability of its stages to customer needs. It is based on rapid and consistent delivery of ready-for-market parts of the product which satisfy the acceptance criteria defined in the specifications of each stage of the project [7]. The Agile Process presupposes that the customer needs are ever changing and require rapid problem solving and flexibility.

We define **Iterative Development** as a way of breaking down the software development process into small easily manageable chunks of work. In this type of development, the code is developed and tested in repeated stages, in which new features are added according to business needs [8].

We define **Incremental Development** as a way to develop a system in increments and evaluate each increment before proceeding to the next increment.

We define **Incremental Delivery** as the deployment of increments for use by end-users. Each deployment can influence the specifications of the future increment.

2.1.1.1 Iterations

In Agile the small iterations in which the process is divided are generally called **sprints** (Figure 7**)**. Each sprint is a time box which usually lasts two to three weeks depending on the team and customer needs and the framework chosen, and it provides small incremental builds in which more features are added, and the work is reviewed. Each increment is evaluated before the start of the next stage. Customer requirements are prioritised and the requirements with highest priority are worked on at the earliest stages of the project. Once the sprint starts, the requirements are frozen for the current sprint and any further features are added to the backlog and picked up in future sprints. The final build is composed of all the features originally requested by the customer and stakeholders [7].

In Agile multiple teams work simultaneously on different areas of the project. For example, the development team, the finance and compliance and the design team, the testing team and the product team would be working on different aspects of the same project and communicating at all times. The short duration of the sprints is one of the features of the Agile model which guarantees continuous updates of all the stakeholders and the teams which are participating to the project.

Each time a sprint ends, a working product is **released** for stakeholders and customer use. Acceptance criteria for each task are described at the beginning of each sprint and discussed between the Product Manager and the development team.

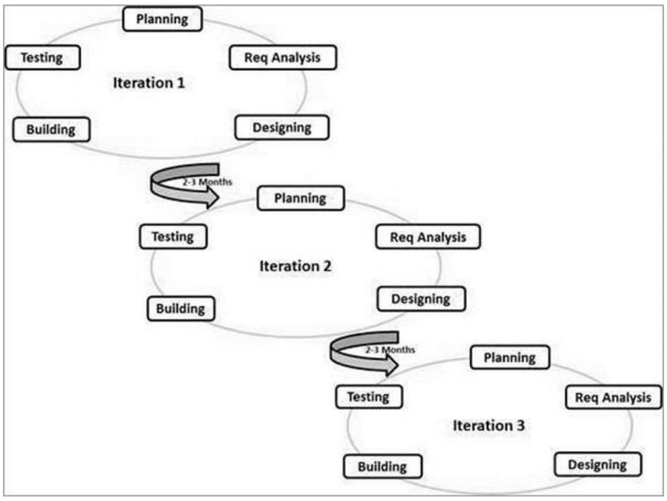


Figure 7 - Example of the Iteration cycle in the Agile Process [7]

2.1.2 The Stages of the Agile Process

The Agile Process assumes that each project is unique and therefore needs to be treated differently from the others. Each iteration of the Agile Process deals with the unique nature of the customer's needs and tailors the stages of traditional software development on the current situation.

2.1.2.1 The Agile Process Life Cycle

The Agile Process Workflow is divided in: Concept, Inception, Iteration/Construction, Release, Production, Retirement [9].

1. At the **Concept** stage the project is envisioned and prioritised over other projects. This is usually a stage in which the Product team discusses possible projects and current business needs. At this stage market research is used to understand the requirements of the project. Within the GoCompare organization, the main responsible for this stage are the Product Team and the Partner Development Management team. The Product team shapes the internal requirements for the product, whilst the PDMs connect with the external stakeholders to define the requirements for the partners.
2. During the **Inception** stage, the teams are selected from the organization. Within each team, members are selected for their skills and roles. The costs and timings of the project are also discussed. General requirements and target platforms are also identified at this stage. Within the GoCompare organization this stage sees the collaboration between the Product team and the Delivery Management, who assign the work to specific teams. The Engineering teams are organised in squads, which follow a flexible structure in terms of roles and skills of the individual team members.
3. The stage which introduces **iterations** (or sprints) is subsequently started, and working software is delivered on the basis of the sprint requirements. Within GoCompare this stage is reserved to the Engineering teams in the OneTech department. Each Engineering team applies the Agile Processes in different ways. The team which deals with the creation of new robots for example, assumes longer working times and a lower number of releases. The Agile Processes are respected in terms of team dynamics and constant updates with stakeholders, daily meetings and reviews of the work.
4. At the end of each sprint, the **release/transition** stage involves testing and development of documentation. The results of the sprint are released to the target platform. Within GoCompare the Test Engineers are responsible for testing the work before release. UAT testing are also carried out before release. These are usually carried out internally and externally. The release is operated by the TechOps who are also responsible for overseeing any malfunctioning once the project is live.
5. The software is subsequently put into **Production**, in which the team manages and update the software, investigates and raises possible bugs.
6. In Agile the **Retirement** phase is the end of life of the release of a system. This typically happens when a company wants to remove a system which becomes obsolete or redundant.

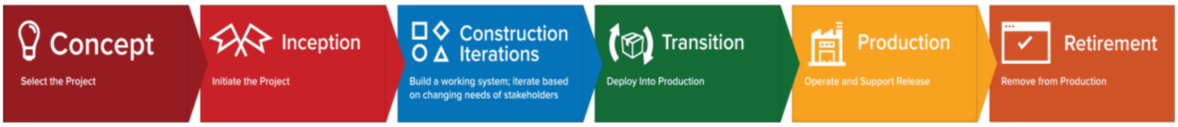


Figure 8- Agile Process Stages [9]

It is important to remember that the Agile Process is based on best practices and flexibility and therefore it is open to manipulation to better serve the purpose of each project. Different frameworks can be applied to the Agile Model and they all differ in the detail of the process.

In the next section we will focus on the Scrum Framework and its application to the IDD Project.

2.2 The Scrum Framework

The Scrum Framework is a methodology of software development based on the Agile Software Development Processes. Scrum was introduced by Ken Schwaber and Mike Beedle and is based on six main characteristics: result flexibility, deadline flexibility, small teams, frequent reviews, cooperation, object orientation [10].

2.2.1 Definition of Scrum

Scum is an iterative and incremental framework for managing complex and extensive software development. As Scrum is part of the Agile Software Development Processes, it focuses on the needs of stakeholders and customers and it assumes that requirements will change along the way. The term scrum is used to identify the chaotic and unpredictable nature of a rugby scrum with the complex and unpredictable nature of software development.

2.2.2 Scrum Phases and Practices

The main phases of the Scrum Framework are **planning** (or Product Backlog), **sprint** and **release** [10].

Scrum starts with an initial phase in which the architecture and outline of the project is **planned**. Once these have been selected, the project enters the iterative process. The Iterative process is composed by four basic stages: **Assertion, Selection, Development** and **Review** (Figure 9**)**. Once all the requirements have been iterative developed the project is closed. During the closure phase the team analyses the previous stages and how the work has been carried out, outlining any issues and comments for future work.

2.2.2.1 Planning

Planning is the first stage in the Scrum Methodology Framework. This stage is prior to the iterative process. During this stage, the backlog is created. The backlog is a list of all the requirements and the necessary properties of the project.

During the Planning stage the product owner draws the road map of the project. The product owner is responsible for determining which features are the priority for the customer and based on that the number and functionality of the releases. The product owner is the business representative in the team. Product owners interact daily with the stakeholders and asks the team to change the flow of the work if necessary.

The Planning stage is also used to determine the team structure and size according to the size and complexity of the project. The technical tools are also discussed at this stage, like for example the language and framework to use, and any additional technical tools that the project might require.

Within GoCompare, the planning is discussed amongst the product manager of the team and the project stakeholders. The product owner would than organise the backlog and assign priority to the user stories. Since the Engineering teams are fixed, at this stage the team responsible for the project is selected according to availability and skills.

2.2.2.2 Sprint

The sprint is one of the main features of an iterative Agile Process. In Scrum this iteration usually lasts two to four weeks. Each sprint provides small functional releases of the work.

The sprint is composed by four stages:

1. **Assess**: the team assesses the remaining backlog items and the requirements.
2. **Select**: the team selects the requirements on which they will be working during the current sprint.

3. **Develop**: in which the team develops the work. This entails designing the program, coding and testing.
4. **Review**: in which the team reviews the work done during the current sprint.

2.2.2.3 Release

The release is the last stage of the scrum cycle in which the newest version of the product is presented to the customer and stakeholders. Feedback is gathered and any issues, bugs or flaws are added to the product backlog. If any requirements have not been met during the current sprint they will be discussed, and an updated version will be added to the new sprint backlog [10].

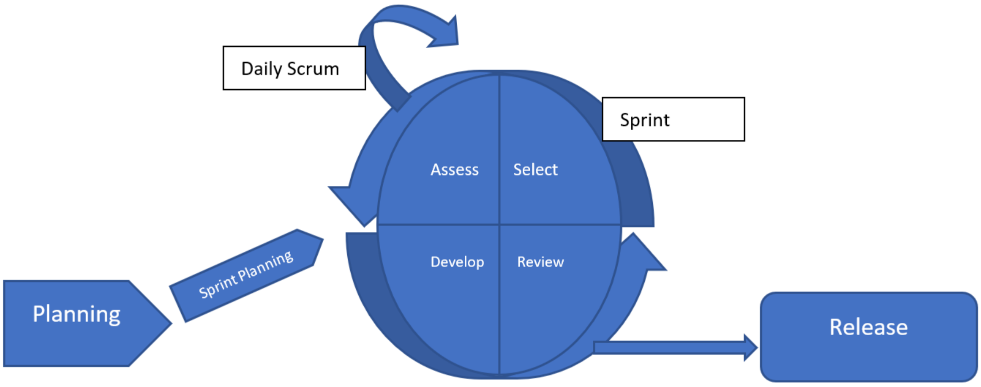


Figure 9- The Scrum Structure

2.3 The Scrum Methodology applied to the IDD Project

2.3.1 Planning the IDD Project

2.3.1.1 Identifying the core requirements

The planning stage of the IDD project would entail the collaboration of a series of department within the GoCompare organization. The Product team would need to deliver appropriate research on the new legal requirements to which the website needs to comply, but also an extended market research on the necessary layout of the additional information and tools provided to the customer. These changes need to be discussed with the Compliance team for the approval of the legal aspect of the changes.

Meetings with the Partner Development Managers need to take place in order to define partners needs. The information displayed in the Wizard tool need to be authorised by the PDMs.

The Minimal Viable Product is defined at this stage. The MVP is the product at its initial stage, complete and functional enough to be released and to get feedback from the stakeholders and early adopters. This is important to minimise the risk of “scope creep” or “gold plating” where unnecessary things are added along the way.

The MVP for the IDD regards the functionality of the Wizard tool as follows:

- The Add-ons can be clicked on the side bar on the results page
- When clicking on an Add-on the Wizard tool is open in a modal
- Customers can answer the questions that they are interested in adding to their insurance
- The price and information displayed to the customers regarding the Add-ons are pulled from the Defaqto Database.

The MVP as specified above entails the collaboration of the development team with several other teams. Possible issues could arise in terms of availability of the other teams (like for example the Database Administrators). This could cause delays in the release of the MVP and subsequent slowing down of the process of feedback.

2.3.1.2 Defining the structure of the team

Before starting any work on the project, the team would meet to define team norms. These norms are necessary to build team dynamics, understand how the team will work together and discuss acceptable behaviours upfront. This meeting would produce a team contract, in which the team lists behaviours for conflict resolution, working relationships and the methodology to assess consensus.

The IDD project requires engineers who are familiar with the Car Insurance Journey. As not all teams work on the same product, a selected number of engineers will be available to take part in the project. The selected team will always need to be provided of a test engineer. The length of the sprints is also discussed during this phase, and they depend on business needs and technical capacity to deliver.

The IDD project requires both backend and frontend implementation. The team structure will also consider the level of experience and the skills of each team member.

Possible issues could arise in terms of team size. Scrum usually requires a team of 6 to 8 developers. GoCompare OneTech is organised in squads of up to 5 members. This might cause overloading the team with work and possibly cause delays in the releases.

2.3.1.3 Writing User Stories

User Stories are the smallest units of work for the Scrum Methodology. Since the end consumer of the product is a user, each sprint is divided in small stories that tell the developer how the product, or part of the product, is going to be used by the user.

Anyone on the team can write a user story but it is usually a responsibility left to the product owner. This ensures that each user story will produce value for the finished product and that the work will stay focused on the requirements and the road map of the project. This will also ensure that all the legal aspects of the IDD project are respected and implemented.

2.3.2 The Sprint

2.3.2.1 Sprint planning and Sprint review

The sprint planning covers the assessment and selection phases of the Sprint. The sprint planning is a meeting in which the activities that must be carried out during the sprint are discussed. In this meeting the sprint priorities are identified, and the Sprint Backlog is drawn. During this meeting the user stories are discussed amongst the team and the complexity and length of the work for each story is identified.

Within the IDD project the sprint planning meeting would be organised by the Delivery Manager of the team and the whole team takes part. During the sprint planning the team checks that the user stories in the backlog are complete with all the information necessary for any member to pick up the story and start working on it. Each team decides on the methodologies of classification of the user stories which can be classified by hours needed to complete the work or story points. If a user story is too high in story points or hours of work, it is divided into different sub tasks which can be worked on separately.

During the sprint review the team discusses what went wrong during the sprint and ways to facilitate work in the future. The team members write down anonymously what they think the team should stop doing, what they should continue doing and what they should start doing in the next sprint. Within the IDD project the teams can use different sprint review tools, like for example Retrium, to organise the review. After discussing each point that has been raised on the correct and incorrect practices followed during the sprint, the team lists the actions to take onto the next sprint.

2.3.2.2 Daily Scrum

The daily scrum is a daily meeting in which the team members discuss what they will be working on during the day. The results of the previous day and any blockers for any previous task are discussed as well. It is common practice to stand during the meeting.

Within GoCompare, the teams generally stand up during the meetings to underline the quick nature of the catch up and stimulate a quicker discussion.

Each team member answers the three questions about “what did I do yesterday?”, “what am I doing today?”, “do I have any blockers?”. The Scrum Master ensures that the Daily Scrums maintain their structure and that they are not used as team meetings. The GoCompare squad structure do not allow for a fixed Scrum Master. This would be a role that each team member covers in turns. This practice can enhance shared responsibility and engagement. The Scrum Master works as a spoke person for the team. He/she shares the team’s progress and is the first escalation point when the team has any blockers. The Scrum Master also checks the team’s performance throughout the project.

2.3.2.3 Development

During development each team member picks up one user story to work on. The IDD project entails the delivery of SQL implementations of new tables and stored procedures for the Car Product and for Defaqto. For the frontend, Vue components can be built separately which allows the team to divide the work in smaller tasks. The Scrum methodology’s tendency to divide the work in small sections allows for the simultaneous implementation of the product’s features to be more functional and effective.

Possible issues might arise if there is a lack of communication between the developers who work on different aspects of the project. The use of very detailed user stories and the daily communication between team members is vital to avoid these issues.

3 The Second Chosen Model

3.1 The Waterfall Model

The Waterfall Model is the first process to be formally described and introduced in Software Development. The first description of the phases of the Waterfall Model can be found as early as 1956, but its formal introduction is usually cited to be in 1970 by William Royce [11]. The name Waterfall recalls the nature of the process which works in cascade from one stage to the other.

3.1.1 Description of the Waterfall Model

The Waterfall Model is a linear-sequential life cycle model. The model follows a linear schema, in which the phases do not overlap, and each phase must be completed before the next phase can begin.

3.1.2 The Stages of the Waterfall Model

The Waterfall Model has six main stages: requirements, design, implementation, verification, deployment, and maintenance [12]. In the Waterfall Model it is possible to move back from one stage to another, but this only happens if there are serious issues to be addressed. (Figure 10)

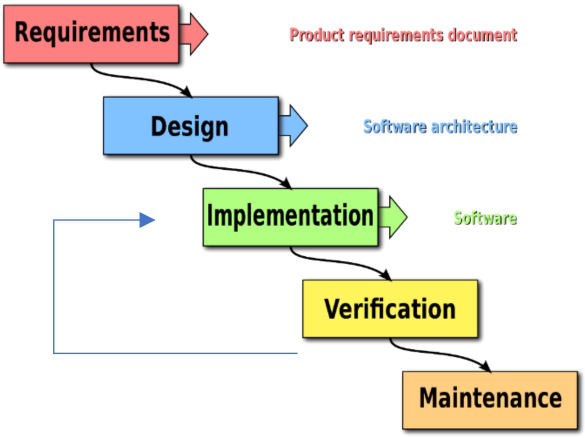


Figure 10- Waterfall Model Stages [11]

3.1.2.1 Requirements

The Waterfall Model is based on the premise that the requirements for the project will be complete and extensive, as it will not be possible to dispute the requirements once the other phases of the process begin.

During this stage the Project Manager will spend an extensive amount of time understanding in detail what the customer wants from the final product. The Project Manager will collaborate closely with the customer and other stakeholders to define in detail the business needs that the software will address, the problems that the software will resolve and the specific functionalities that the customer expects from the final product.

When all the stakeholders are satisfied with the requirements identified, the Requirements Document is compiled and approved, closing the first stage of the Waterfall Model.

3.1.2.2 Design

During this stage the implementation team analyses the requirements and designs the software. The design phase is composed of the Logical Design and the Physical Design. The Logical Design entails the definition of the abstract structure of the software. It is usually presented as the diagram of the flow of the software, whilst the Physical Design is the definition of the hardware which will be used to physical implement the software.

The implementation team must approve the Software Design before the next phase can begin.

3.1.2.3 Implementation

During the Implementation phase the software developers build the software based on the requirements. The implementation of the software is usually divided in units which are passed to the testing team for the verification phase. In this phase functional testing is carried out, also called Unit Testing. This type of testing allows to test each unit of work individually.

3.1.2.4 Verification

During the Verification Phase non-functional testing is carried out on the finished product. This type of testing is carried out on a selected group of users. The user feedback is recorded and checked against the original requirements of the product. If the software does not satisfy the requirements and fails user testing, it is sent back to the development team for a second implementation phase.

3.1.2.5 Deployment

During the deployment phase the software is released into the selected environment or on the market. This phase can be also broken into units to allow for the maintenance to be carried out. The product can be released into different locations, once it is fully deployed in one location, the software can be deployed in a second location to allow for the maintenance of the first location to take place.

3.1.2.6 Maintenance

During the maintenance phase, issues start to be raised and resolved. The customer feedback allows bugs and malfunctioning to be analysed. More testing can be carried out during this phase in the live environment.

3.2 The Waterfall Model applied to the IDD project

3.2.1 Project Management

The requirements are described in detail by the Project Manager. In the case of the IDD Project, the Product team owns the project and will be responsible for the definition of the requirements. The Product Team will organise several meetings with the stakeholders, like the Compliance team and the Finance Team who are responsible for the legal aspects of the project as seen above.

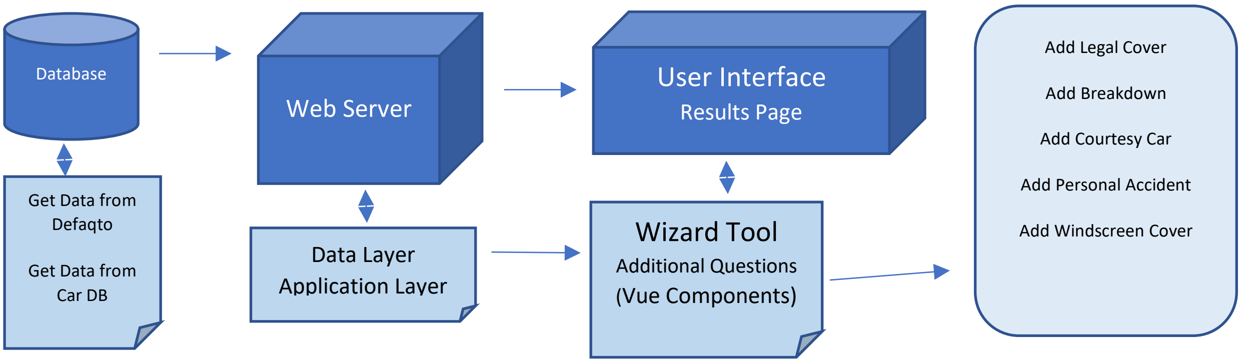
Once the requirements are discussed and defined with the stakeholders, the Requirements Document is approved by the Project Manager and the stakeholders.

3.2.2 Development Team

The development team is responsible for the planning and the implementation of the system. Based on the requirements documentation, the development team will start the Design phase. Since the IDD project entails structural changes in the treatment of static and non-static data, the development team should be composed of both the software developers and the database administration team. The DBAs will overview the changes in the Defaqto database and ensure appropriate unit testing.

3.2.3 The Design

During the Design phase the development team defines the System Architecture. This is composed of an abstract representation of the system. For example, the system to implement for the IDD project could be represented at a high level as follows:



The Data Flow of the application is also defined and approved by DBA (Figure 11).

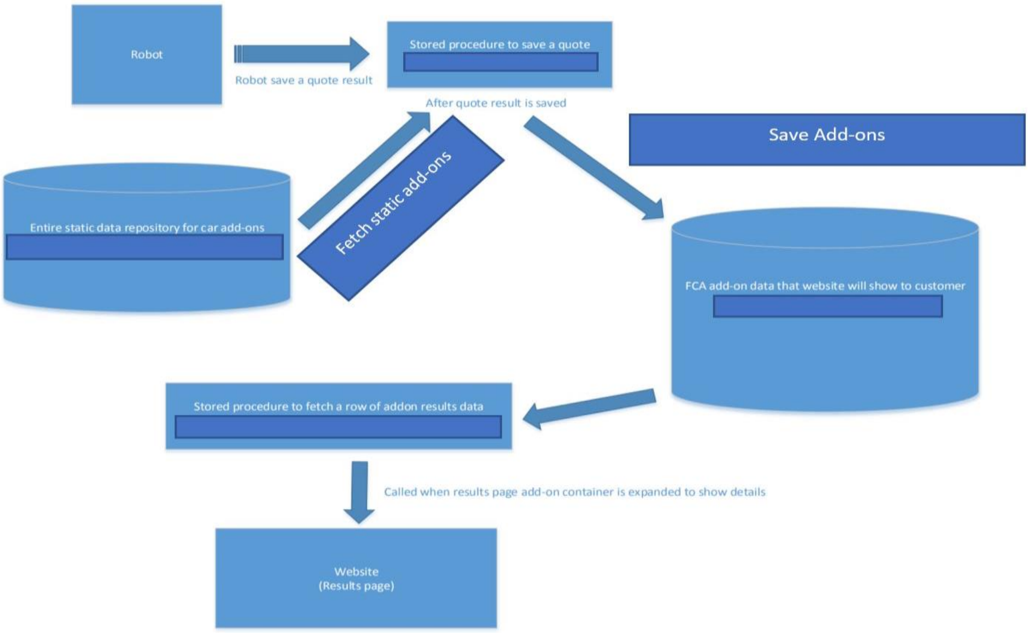

Robot
Robot save a quote result
Stored procedure to save a quote
After quote result is saved
Save Add-ons
Entire static data repository for car add-ons
Fetch static add-ons
FCA add-on data that website will show to customer
Stored procedure to fetch a row of addon results data
Called when results page add-on container is expanded to show details
Website
(Results page)

Figure 11- IDD Data Flow Example

The Verification Plan is also described in the Design Phase. The Verification plan for the IDD project would entail Unit Testing and user-oriented testing. Tests should include cases to ensure the functioning of each unit of work. For example:

- When the customer lands on the results page, testing should ensure that the correct text is displayed under the Add-ons. If the Add-on is available to be added to the policy, the Add-on should say “Add-on Available”
- When clicking on the side bar the Wizard tool should open in a modal. The questions should produce the new result for the customer
- The add-ons data should be pulled from the non-static data in the Defaqto Database.

Issues at this stage might arise. Whilst the Waterfall model is easy to understand and manage, it is also not particularly flexible and is not suitable for long-term and complex system development. Whilst the actual changes that the IDD project aims to implement might not seem too extended, they become very complex when implemented in a very large existing system.

During the **implementation**, new issues would arise. For example, the existing code might not allow for a certain solution to be achieved in a way that can fully satisfy the original requirements. Nevertheless, the legal nature of the project could also present an issue, as legislation or its interpretation at the hands of internal stakeholders might change at any point. As we have seen above (3), the original requirements did not foresee the need to change other questions in the Car Journey.

4 Agile Model vs Waterfall Model

4.1 Advantages of the Agile Model

As we have seen above, the Agile model presents many advantages linked to its flexibility and adaptability to change.

- The constant delivery of a useable product allows the company to receive feedback and customer satisfaction. The stakeholders, customers and developers are in constant communication.
- The continuous delivery also allows the company to keep up with business trends and adapt to sudden market changes, rather than having to work at the same project for months only to discover that the market has moved on.
- The daily communication between different areas of the business allows employees to be conscious about what goes on in other teams, practice which would not be possible otherwise.
- The tendency to being open to change also stimulates developers to find newer ways to improve the original work and produce better code. This would favour developers' engagement with the product and encourage their input in raising features and new solutions.
- The multiple releases also allow for more testing. A higher number of testing can enhance behaviours which favour automated solutions and formalise testing processes.
- The presence of the Scrum Master in the Scrum Methodology also encourages engagement and can ensure the exchange of ideas with the business when it comes to raising issues regarding the amount of work or the quality of the project itself. Having an official figure as spoke person for the team can encourage employees to speak out in case of misbehaviour both from the business and from other members.

4.2 Disadvantages of the Agile Model

The Agile Model also presents disadvantages.

- The Agile Model does not produce extensive documentation. The flexible nature of the model means that there is a lack of emphasis on the documentation. The extensive requirements which we can see in the Waterfall Model, would not be adaptable to the fast-paced nature of the Agile Model, where requirements are constantly re-discussed at the beginning of every spring. The requirements are also usually stored in different places, like team boards, TFS boards or Trello, or even email chains. This can cause the lack of structure in the way projects are archived, making it hard for developers to produce Handover documentation.
- The lack of strict requirements can lead the project to get out of track. This can happen for many reasons, like for example the customer not being sure about the outcome, or not foreseeing the real size of the project. This happened in the IDD project, where the squad had several issues with new features being added as the project progressed. The choice of Kanban also had an impact on the work, as this methodology does not use iterations, which caused an increased lack of structure. A sudden change in personnel, in particular the turnover of Product Owners, has also influenced negatively the development of the IDD project.
- Issues could arise in terms of confidence when taking decisions during the development of the system. With the Scrum Master being a role that all team members can cover, the junior

developers and younger developers might be put in a difficult position when covering the role.

- Agile Model Methodologies require training which the company is responsible for. Usually this requires the presence of an agile coach in the company. This leads to costs and possible confusion in terms of roles and responsibilities.

4.3 Advantages of the Waterfall Model

- The Waterfall Model follows a clear and strict structure. The steps of this model are more defined and easier to follow. The structure is simpler compared to other methodologies and is followed in the same way for every project.
- The project needs to be completed under every aspect before being deployed. This means that future developers will not find half finished projects in which only the MVP has been deployed.
- Waterfall does not require the presence of a coach or any particular figure to be implemented as it is a more intuitive and simpler model than Agile.
- The end product is known from the beginning. The squads are not subject to ever growing backlogs and demands that could get them out of track.
- The Waterfall Model provides extensive documentation. All information must be complete and effective when describing the requirements and commenting and Javadoc documentation makes it easier for developers to exchange information with newcomers or other stakeholders.

4.4 Disadvantages of the Waterfall Model

- The Waterfall Model is not flexible and adaptable to change. This can be negative when working on projects like the IDD where the developers are working with an existing complex system with historical Technical Debt. When working with existing code, developers would need a framework which would allow them to react to blockers more effectively.
- The process of user feedback is slower and harder to achieve. In this model the customer and stakeholders are not the focus during implementation. This could lead to solutions which do not fully represent the needs of the customer.
- Testing is only carried out towards the end of the project. This could mean that by than the project has already been shaped and has been implemented for several months. This means that it will be harder to modify the work done so far and could lead to delays and inefficient compromises.

4.5 What is best for the IDD?

The characteristics of the IDD Project suggest that the Agile Model would be the most appropriate for a more efficient development.

Whilst taking into account the disadvantages of the Agile Model as explained above (17), and the fact that the teams could easily fall into any of those blockers whilst implementing the project, these are risks worth taking if the stakeholders want to arrive to a more accurate representation of the business needs.

As explained above, the changes implemented are part of an important piece of regulation which could cause the company financial loss and legal repercussions in case of noncompliance. Adapting to the needs of the internal stakeholders like the Finance and Compliance teams, and external stakeholders like the FCA, is vital to the success of the project.

Nevertheless, the project aims to implement code for an existing more complex system. This makes the process unpredictable as the developers will only have a general idea of how the existing code has been written and how it behaves. Existing bugs could be found during the process which could need to affect the details of the requirements.

The IDD project is composed of several elements which can be easily divided in smaller sections of functional product. The use of an iterative process like Scrum, allows the company to identify and work on the most pressing parts of the work, like updating the data displayed to the customer with non-static data from Defaqto or displaying important additional text to the customer. This also allows the Product Owner to receive the appropriate feedback and report to the development team.

5 References

[1] FCA, “FCA Financial Conduct Authority,” 2018. [Online]. Available: https://www.fca.org.uk/firms/insurance-distribution-directive.

[2] Aha!, “Product Management,” Aha, [Online]. Available: https://www.aha.io/roadmapping/guide/product-management/what-makes-up-the-product-team#:~:text=The%20product%20team%20plays%20a,and%20loss%20(P%26L)%20responsibilities..

[3] A. W. Series, “What is DevOps?,” Amazon, [Online]. Available: https://aws.amazon.com/devops/what-is-devops/.

[4] “What is Agile Kanban Methodology?,” Inflectra, [Online]. Available: https://www.inflectra.com/methodologies/kanban.aspx#:~:text=Kanban%20is%20an%20agile%20methodology,in%20one%20large%20development%20cycle..

[5] Various, “Agile modeling,” Wikipedia, [Online]. Available: https://en.wikipedia.org/wiki/Agile_modeling#Core_practices.

[6] R. Lynn, “The History of Agile,” planview, [Online]. Available: https://www.planview.com/resources/guide/agile-methodologies-a-beginners-guide/history-of-agile/#:~:text=It%20all%20started%20in%20the,new%20software%20to%20market%20faster..

[7] SDLC, “SDLC - Agile Model,” Software Development Life Cycle, [Online]. Available: https://www.tutorialspoint.com/sdlc/sdlc_agile_model.htm#:~:text=Agile%20SDLC%20model%20is%20a,builds%20are%20provided%20in%20iterations..

[8] M. Rouse, “iterative development,” TechTarget, [Online]. Available: https://searchsoftwarequality.techtarget.com/definition/iterative-development#:~:text=Iterative%20development%20is%20a%20way,and%20tested%20in%20repeated%20cycles..

[9] L. C. Team, “The Stages of the Agile Software Development Life Cycle,” Lucidchart, [Online]. Available: https://www.lucidchart.com/blog/agile-software-development-life-cycle#:~:text=Retirement,contrary%20to%20your%20business%20model..

[1 B. V. D. C. a. C. H. P. Mello, “Scrum agile product development method -literature review,
0] analysis and classification,” *Research Gate,* no. DOI: 10.4322/md.2011.005, 2011.

[1 t. f. e. Wikipedia, “Waterfall model,” Wikipedia, [Online]. Available:
1] https://en.wikipedia.org/wiki/Waterfall_model#History.

[1 S. D. L. Cycle, “SDLC - Waterfall Model,” SDLC, [Online]. Available:
2] https://www.tutorialspoint.com/sdlc/sdlc_waterfall_model.htm.