

**MESG**  
**MESTRADO EM ENGENHARIA**  
**DE SERVIÇOS E GESTÃO**

**Padronização de testes de software e avaliação das equipas  
de desenvolvimento**

*Cristiana Pacheco Oliveira*

**Dissertação de Mestrado**

Orientador na FEUP: Prof. Jorge Pinho de Sousa

Orientador na msg Life Iberia : Engenheiro Luís Marques



2018-06-29

*“Tenho em mim todos os sonhos do mundo”*  
**Fernando Pessoa**

## Resumo

O projeto de estágio foi desenvolvido em contexto empresarial, tendo como principal objetivo a concepção de uma metodologia de apoio à execução de testes de software juntamente com a avaliação das equipes de desenvolvimento fomentando uma avaliação e monitorização coletiva e participativa das equipes.

A empresa selecionou as áreas que considera determinantes para a elevação da qualidade do produto, nomeadamente, o rastreamento de *bugs* e testes de *software*.

A avaliação de desempenho das equipes de desenvolvimento, no que diz respeito ao número de *bugs*, permite avaliar continuamente a posição e evolução das equipes. Esta metodologia permite o acompanhamento eficaz de cada área do processo e o novo formato de registo das ocorrências de *bugs*, recorrendo ao software Jira, promove o investimento no estudo das causas dos erros no software e a transparência dos resultados na organização.

Adicionalmente a este modelo, foi introduzida uma nova métrica, cumprimento do *sprint*, de modo a entender o impacto dos *bugs* no desempenho das equipes.

Como complemento foi introduzida uma nova ferramenta de apoio aos testes de software, plano de testes, que tornará o processo de testes mais eficaz. Um plano de teste garante que todos os aspetos do software sejam testados e cobertos, no momento em que os testes sejam realizados de maneira estruturada e seguindo o plano previamente detalhado.

## **Standardization of software testing and evaluation of the development teams**

### **Abstract**

The internship project was developed in a business context, with the main objective being the design of a methodology to support the execution of software tests together with the evaluation of the development teams, encouraging a collective and participatory evaluation and monitoring of the teams.

The company selected the areas that it considers determinants for the raising of product quality, namely, the tracking of bugs and tests of software.

The performance evaluation of development teams, in terms of the number of bugs, allows continuous assessment of team position and evolution. This methodology allows the effective monitoring of each area of the process and the new format of registration of the occurrences of bugs, using the software Jira, promotes the investment in the study of the causes of the errors in the software and the transparency of the results in the organization.

In addition to this model, a new metric, sprint compliance, was introduced to understand the impact of bugs on team performance.

As a complement, a new software testing support tool, test plan, has been introduced that will make the testing process more effective. A test plan ensures that all aspects of the software are tested and covered at the time the tests are performed in a structured manner and following the previously detailed plan.

## **Agradecimentos**

Agradeço ao professor orientador Jorge Pinho de Sousa, pelo apoio e disponibilidade manifestada.

À empresa onde desenvolvi a dissertação, a todos os funcionários pelo carinho e simpatia, em especial ao meu orientador, Engenheiro Luís Marques, pelas oportunidades, pelas ideias e sugestões apresentadas, pela atenção e apoio.

À minha família, em especial aos meus pais e irmãos, pelo apoio e compreensão.

Para finalizar, o meu grande agradecimento aos meus amigos pela convivência e amizade nos bons e maus momentos.

Obrigada a todos.

## Índice de Conteúdos

|       |                                                            |    |
|-------|------------------------------------------------------------|----|
| 1.    | Introdução                                                 | 0  |
| 1.1   | Enquadramento                                              | 0  |
| 1.2   | Objetivos                                                  | 1  |
| 1.3   | Metodologia e plano de trabalho                            | 1  |
| 1.4   | Estrutura do relatório                                     | 2  |
| 2.    | Revisão da Literatura                                      | 3  |
| 2.1   | Desenvolvimento de <i>software</i> ágil                    | 3  |
| 2.1.1 | Scrum                                                      | 4  |
| 2.1.2 | Kanbans                                                    | 6  |
| 2.1.3 | Software Jira: ferramenta para desenvolvimento de software | 6  |
| 2.2   | Lean aplicado ao desenvolvimento de <i>software</i>        | 7  |
| 2.3   | Garantia da qualidade                                      | 8  |
| 3.    | Caracterização do problema                                 | 10 |
| 3.1   | Enquadramento empresarial                                  | 10 |
| 3.1.1 | A empresa                                                  | 10 |
| 3.1.2 | Projetos                                                   | 11 |
| 3.2   | Descrição do problema a tratar                             | 12 |
| 3.2.1 | Questões identificadas                                     | 12 |
| 4.    | Abordagens adotadas                                        | 14 |
| 4.1   | Rastreamento de <i>bugs</i>                                | 14 |
| 4.1.1 | Identificação de oportunidades de melhoria                 | 14 |
| 4.1.2 | Abordagem adotada                                          | 15 |
| 4.2   | Testes de <i>software</i>                                  | 16 |
| 4.2.1 | Identificação de oportunidades de melhoria                 | 17 |
| 4.2.2 | Abordagem adotada                                          | 17 |
| 5.    | Resultados                                                 | 19 |
| 5.1   | Recolha de informação                                      | 19 |
| 5.2   | Análise dos Bugs                                           | 21 |
| 5.2.1 | Análise global                                             | 21 |
| 5.2.2 | Equipa GUP                                                 | 22 |
| 5.2.3 | Equipa UDP                                                 | 23 |
| 5.2.4 | Equipa VHV Web App                                         | 25 |
| 5.2.5 | Equipa VHV BE/Integration                                  | 26 |

|       |                                                                            |    |
|-------|----------------------------------------------------------------------------|----|
| 5.3   | Análise Sprints                                                            | 26 |
| 5.3.1 | Análise global                                                             | 26 |
| 5.3.2 | Equipa GUP                                                                 | 28 |
| 5.3.3 | Equipas VHV                                                                | 29 |
| 5.4   | Plano de testes                                                            | 31 |
| 6     | Conclusões e perspectivas de trabalho futuro                               | 32 |
| 6.1   | Conclusões                                                                 | 32 |
| 6.2   | Perspetiva de trabalhos futuros                                            | 33 |
|       | Referências                                                                | 34 |
|       | ANEXO A: Fluxo de trabalho JIRA                                            | 35 |
|       | ANEXO B: Exemplo ficheiro requisitos                                       | 36 |
|       | ANEXO C: Excerto ficheiro excel equipa GUP                                 | 37 |
|       | ANEXO D: Excerto ficheiro excel equipa UDP                                 | 38 |
|       | ANEXO E: Excerto ficheiro excel equipa VHV Web App                         | 39 |
|       | ANEXO F: Excerto ficheiro excel equipa VHV BE/Integration                  | 40 |
|       | ANEXO G: Gráficos relativos ao tipo de tarefas concluídas fora do planeado | 41 |
|       | ANEXO H: Plano de testes VHV versão P 0.1                                  | 42 |
|       | ANEXO I: Plano de testes VHV versão P 0.2                                  | 46 |

## Lista de Tabelas

|                                                            |    |
|------------------------------------------------------------|----|
| Tabela 1: Número de bugs encontrados anualmente .....      | 12 |
| Tabela 2: Lista de causas raíz .....                       | 14 |
| Tabela 3: Causas dos bugs da equipa GUP .....              | 22 |
| Tabela 4: Causas dos bugs da equipa UDP .....              | 23 |
| Tabela 5: Causas dos bugs da equipa VHV Web App .....      | 25 |
| Table 6: Causas dos bugs da equipa VHV BE/Integration..... | 26 |
| Tabela 7: Datas sprints GUP .....                          | 29 |

## Lista de Figuras

|                                                                           |    |
|---------------------------------------------------------------------------|----|
| Figura 1: Presença global do grupo msg .....                              | 10 |
| Figura 2: Rede msg life .....                                             | 10 |
| Figura 3: Especificação dos projetos .....                                | 11 |
| Figura 4: Número de bugs Março e Abril.....                               | 21 |
| Figura 5: Gráfico das equipas que originaram os bugs .....                | 22 |
| Figura 6: Equipas que causaram os bugs na equipa UDP .....                | 24 |
| Figura 7: Equipas que causaram os bugs na equipa VHV Web App .....        | 25 |
| Figura 8: Gráfico cumprimento sprint .....                                | 27 |
| Figura 9: Perceção global dos sprints da equipa GUP .....                 | 28 |
| Figura 10: Tipo de tarefas concluídas fora do planeado.....               | 28 |
| Figura 11: Perceção global dos sprints da equipa VHV Web App .....        | 29 |
| Figura 12: Perceção global dos sprints da equipa VHV BE/Integration ..... | 30 |

## Introdução

Devido à enorme competitividade que se vive atualmente no mundo empresarial, as empresas são obrigadas a procurar soluções que lhes permitam manter ou melhorar a sua posição no mercado em que atuam.

Nesse contexto, a exigência do mercado do *software* tem conduzido as empresas a procurarem novas estratégias que garantam a qualidade, elevando o foco na excelência.

Controlar a qualidade de *software* representa um grande desafio devido à alta complexidade e às barreiras relacionadas com o processo de desenvolvimento, sendo assim, a execução de testes torna-se uma etapa crítica do processo.

A par da melhoria processual, a avaliação do desempenho revela-se fulcral. Se não se consegue medir, não se consegue controlar; se não se consegue controlar, não se consegue gerir; se não se consegue gerir, não se consegue melhorar (Harrington, 1997). É um processo através do qual se pretende quantificar a eficiência e eficácia. Medir é o ponto chave.

Seguindo este pensamento, é essencial a construção de métricas que posteriormente facilitem a criação de ações e iniciativas que pretendam melhorar o processo e ao mesmo tempo que permitam o crescimento profissional e pessoal de cada colaborador.

### 1.1 Enquadramento

A presente dissertação tem como propósito expor o projeto de estágio realizado no âmbito do Mestrado em Engenharia de Serviços e Gestão (MESG) da Faculdade de Engenharia da Universidade do Porto (FEUP).

O projeto foi desenvolvido em contexto empresarial, na empresa *msg life Iberia*, tendo como área de negócio o desenvolvimento de soluções de software e consultoria no mercado segurador, aconselhando clientes na implementação das suas estratégias alinhadas às tecnologias de informação, tendo como principal objetivo a conceção de uma metodologia de apoio à execução de testes de software juntamente com a avaliação das equipas de desenvolvimento fomentando uma avaliação e monitorização coletiva e participativa das equipas.

O desafio é adequar a teoria do negócio à realidade de empresa, dentro da realidade exigida pelos clientes, encontrando um equilíbrio.

## 1.2 Objetivos

A msg life Iberia trabalha para levar aos seus clientes os processos e tecnologias mais inovadores e eficientes, oferecendo às seguradoras a capacidade de responder a este mercado cada vez mais dinâmico. Numa fase de crescimento, a empresa tem como objetivo adotar continuamente as melhores práticas para garantir a qualidade do produto e serviço.

Nesse cenário de melhoria contínua e prospeção no mercado, o projeto consiste em construir uma metodologia de apoio aos testes de software, de forma a executar o software de forma controlada com o objetivo de avaliar se o mesmo se comporta conforme as especificações e expectativas definidas e esperadas pelo cliente. É necessário também assegurar que o procedimento seja de fácil compreensão e intuitivo. Pretende-se também a avaliação de desempenho das equipas de desenvolvimento com o objetivo de melhorar a sua performance, através da adoção de boas práticas, paralelamente à identificação de necessidades de medição e recolha de dados para a deteção de problemas nos processos atuais, e também de medidas de otimização adequadas à realidade da *msg life Iberia*.

O mercado exige produtos de maior qualidade, reivindicando a constante preocupação em criar produtos de qualidade e sem erros, e é isso que se pretende com estas abordagens.

## 1.3 Metodologia e plano de trabalho

O trabalho foi realizado em conjunto com as unidades de desenvolvimento de produto e de implementação de projetos da *msg life Iberia*, com o seguinte plano de trabalho:

Primeiramente, a fase de reconhecimento empresarial visando analisar a situação atual dos processos e estudar a forma de trabalho das várias equipas de desenvolvimento, integrando os projetos correntes na *msg life Iberia*, conjugando também os projetos já finalizados que beneficiam agora do serviço de suporte.

Numa segunda fase, procedeu-se ao estudo de metodologias aplicadas ao desenvolvimento de software e implementação de projetos.

Posteriormente, em conjunto com as equipas e dependendo das necessidades dos projetos, estabeleceram-se melhorias processuais que ajudassem a garantir mais qualidade e eficácia das equipas, bem como a conceção de uma metodologia de apoio à execução de testes de software, integrando um dos projetos a decorrer.

A quarta fase corresponde à análise das soluções propostas, parte delas já vindo a ser aplicadas no decorrer do estágio.

## **1.4 Estrutura do relatório**

Este relatório está dividido em seis capítulos. No primeiro capítulo faz-se uma introdução inicial ao projeto. O segundo capítulo destina-se a referenciar todos os pressupostos relevantes para a fundamentação do projeto de estágio. De seguida, o terceiro capítulo efetua uma breve apresentação da empresa e introduz os motivos que levaram ao tema da dissertação e a sua importância para a organização. O quarto capítulo, inicia a componente prática do projeto, assentando no relato da situação inicial da empresa, expondo os problemas detetados, bem como a apresentação das abordagens adotadas. Relativamente ao quinto capítulo, o mesmo direciona-se para a expor os resultados que vão de encontro às propostas implementadas. No último capítulo, faz-se um balanço geral do projeto e do impacto que este teve para a empresa. Apresentam-se as principais conclusões. Por fim, nos anexos, estão agrupados documentos que servem de suporte para o presente desígnio.

## 2. Revisão da Literatura

O presente capítulo destina-se a elucidar e referenciar os conceitos teóricos que serviram de suporte para o desenvolvimento da dissertação.

Começando por abordar a metodologia ágil, pela qual a *msg life Iberia* se rege. Posteriormente serão apresentadas questões relacionadas com a introdução do Lean no desenvolvimento de *software*, que fornecerão de forma simples as fundações das novas abordagens adotadas. Por fim, será abordado o tema garantia da qualidade que se afigura de extrema importância na área de negócio que esta inserida a empresa.

### 2.1 Desenvolvimento de *software* ágil

“O modelo tradicional de desenvolvimento de software é a abordagem em cascata, em que o desenvolvimento “flui” para baixo, como uma cascata, através de seis fases: análise, projeto, implementação, teste de validação, integração e manutenção. Os defensores deste modelo sustentam que uma nova fase não deve ser realizada até que a fase anterior tenha sido concluída, revisada e verificada. Este processo é eficaz, mas possui falhas - não possui processos iterativos e desperdiça tempo devido à duração de cada fase. Esta abordagem é mais adequada para projetos com requisitos fixos, um produto firme e estável e tecnologia claramente compreendida” (Stiner, 2017).

A *msg life Iberia* rege-se por um desenvolvimento *Agile* que assenta em conceitos já implementados na empresa.

*Agile* significa ser capaz de “Entregar rapidamente. Mudar rapidamente. Mudar frequentemente ”(Highsmith et al., 2009), possibilitando a entrega rápida de software de alta qualidade em alinhamento com as necessidades do cliente através de práticas que facilitam a comunicação entre a empresa e o cliente.

O termo *Agile* foi introduzido em 2001, com a declaração do Manifesto que se baseia em quatro valores:

1. **indivíduos e interações** mais que processos e ferramentas;
2. **software em funcionamento** mais que documentação abrangente;
3. **colaboração com o cliente** mais que negociação de contratos;
4. **responder a mudanças** mais que seguir um plano.

Quando comparada com abordagens de gestão tradicionais, a metodologia ágil oferece um conjunto maior de benefícios, tais como: aumento da produtividade das equipas e a

satisfação dos colaboradores; redução do esforço gasto em documentação excessiva, reuniões redundantes e planeamento repetitivo; aumento da interação com o cliente, com foco na transparência e adaptação contínua às exigências do cliente, reajustando a oferta às necessidades do setor, levando ao mercado um produto mais valioso; e a possibilidade de gerir o risco associado ao negócio de uma forma mais controlada.

O desenvolvimento de *software Agile* propõe um formato de trabalho eficaz e eficiente para o desenvolvimento de *software*, seguindo um registo incremental. Através de ciclos iterativos e entregas incrementais, a metodologia ágil permite responder eficazmente a possíveis adaptações ou mudanças nos requisitos, a qualquer altura do desenvolvimento. Resumindo, esta metodologia destaca-se pelo desenvolvimento incremental e iterativo; desenvolvimento adaptativo em vez de perspectivo; timeboxing e comunicação presencial em tempo real.

### 2.1.1 Scrum

O Scrum é baseado nos princípios e fundamentos da metodologia *Agile*, descrita como “Uma *framework* dentro do qual as pessoas podem tratar e resolver problemas complexos e adaptativos, enquanto que de forma produtiva e criativa entregam produtos com o mais alto valor possível”. (Schwaber et Sutherland, 2013)

Scrum assenta num conjunto de papéis, eventos e artefactos, que se baseia num desenvolvimento realizado de forma interativa e incremental, traduzido por *sprint*, cujo objetivo é promover a transparência e o constante *feedback* a nível da equipa e integrando também o cliente. Seguindo esta estrutura, o desenvolvimento é dividido em *sprints*, em que os produtos são progressivamente desenvolvidos e melhorados de forma iterativa e incremental, permitindo a entrega contínua de partes do produto. A criação de *sprints* otimiza a previsibilidade e o controlo de risco, constituindo a base de uma gestão eficaz.

Em suma, esta abordagem define uma estratégia flexível e holística de desenvolvimento de produtos.

Scrum consiste em três papéis: o *product Owner*, o *Scrum master* e a equipa de desenvolvimento.

1. *product owner*: como representante do cliente, é responsável por salvaguardar os interesses dos *stakeholders* e assegurar que o *product backlog* permanece priorizado.
2. *Scrum master*: responsável por garantir que os valores e as práticas do Scrum foram entendidos pela equipa e estão a ser seguidos, é quem gere todo o processo. Remover impedimentos ou problemas que possam prejudicar a equipa ou interromper a

atividade que move o projeto para frente, torna-se uma das características mais relevantes para esta função.

3. Equipa de desenvolvimento: são estruturadas e capacitadas pela organização para se tornarem multi-funcionais e auto-organizadas. A sinergia resultante otimiza a eficiência e a eficácia da equipa de desenvolvimento.

Relativamente aos artefactos, três são cruciais: o *product backlog*, o *sprint backlog* e o *burndown chart*.

1. *product backlog*: O *product owner* e a equipa de desenvolvimento colaboram (processo contínuo) para criar uma lista ordenada de tudo o que é conhecido como necessário no produto (requisitos). O *product backlog* é dinâmico, pois este está em constantemente mudança/atualização para satisfazer/identificar as necessidades do cliente. O *product owner* é o responsável por gerir o *product backlog* (atualizar, priorizar e esclarecer) e garantir que os requisitos estão a ser seguidos para atender às necessidades do cliente.
2. *sprint backlog*: é uma lista priorizada de tarefas ao qual a equipa se compromete a concluir durante o *sprint*.
3. *burndown chart*: este tipo de gráfico serve para monitorizar o progresso das tarefas no decorrer do *sprint*, de forma a determinar rapidamente se será cumprido o planeamento.

As reuniões ocorrem em momentos específicos do sprint, designadas como: o *sprint planning*, a *daily Scrum*, o *sprint review* e *sprint retrospective*.

1. *sprint planning*: nesta fase, é realizada uma reunião com o intuito de planear o próximo sprint. É feita uma análise prévia do *product backlog*, sendo analisado a viabilidade das item/tarefa e criadas estimativas da sua duração, com a finalidade de conceber/construir o o *sprint backlog*.
2. *daily Scrum*: esta reunião (também conhecida por *standup*) é reservada para os membros das equipas partilharem o que concluíram no dia anterior, o que está em curso no dia atual e identificar impedimentos ou obstáculos ao progresso, nada mais, não é uma reunião de solução de problemas, acontece todos os dias com o objetivo de sincronizar o trabalho e melhorar a comunicação da equipa. Esta reunião não deve demorar mais de 15 minutos.
3. *sprint review*: também conhecido com *showcase*, ocorre no último dia do sprint, com a finalidade de demonstrar o que foi desenvolvido durante o sprint, geralmente é feita

uma demonstração, permitindo a partilha de conhecimento e também a obtenção de um feedback rápido. Neste momento, o cliente também é envolvido.

4. *sprint retrospective*: no final de cada sprint, a equipa tem a oportunidade de refletir/rever o *sprint* e identificar oportunidades de melhoria para o próximo. Em síntese, é uma reunião de *lessons learned*.

### 2.1.2 Kanbans

O *Kanban* é um método integrante da filosofia Lean que consiste num sistema visual, baseado na utilização de cartões, que contempla um desenvolvimento e entregas contínuas, lidando com um pequeno número de tarefas, de forma fluída e simultânea. O objetivo do *Kanban* é equilibrar o fluxo de trabalho.

Enquanto que o *Kanban* foi inserido por Taiichi Ohno na indústria de produção, foi David J. Anderson o primeiro a introduzir este conceito em desenvolvimento de software, no ano 2004.

Independentemente do quadro *kanban* ser físico ou digital, a sua função é garantir que o trabalho da equipa seja visualizado, o fluxo de trabalho seja padronizado e que todos as vertentes bloqueantes e dependências sejam imediatamente identificados e resolvidos. Um quadro *kanban* básico tem um fluxo de trabalho de três etapas: *To-do*, *In progress*, *Done*. No entanto, dependendo da complexidade do projeto, o fluxo pode ser mapeado de acordo com o processo que está a ser seguido.

Esta metodologia não segue um processo iterativo, contrariamente ao que acontece com *Scrum*, em que o *sprint* é o conceito principal.

A metodologia *kanban* depende da total transparência do trabalho e da comunicação em tempo real da capacidade.

### 2.1.3 Software Jira: ferramenta para desenvolvimento de software

O *software* Jira é uma plataforma robusta que permite planear, rastrear e gerir projetos de desenvolvimento ágil. Destacam-se como principais características do Jira, a gestão de projetos, permitindo a criação e delegação de projetos e tarefas, com fluxos de processos customizados para o acompanhamento dos projetos; a gestão de defeitos e incidentes, com funcionalidades para criação, descrição de defeitos/incidentes e seguimento da resolução; a facultação de diversos tipos de relatórios incorporados, com base em dados visuais em tempo real; e a integração com outras ferramentas, como por exemplo o Confluence. A integração do Confluence e Jira permite a transparência nos projetos e a vinculação automática de problemas

abertos no Jira à documentação de software no Confluence. A capacidade de criar problemas e emitir relatórios com rapidez permite que sua equipe trabalhe de maneira mais inteligente e rápida. Resumidamente, o Jira proporciona ferramentas que apresentam o panorama completo do projeto.

Na fase de configuração, é necessário decidir qual a estrutura que melhor se adequa: Scrum ou Kanban? A estrutura Scrum concede a possibilidade de planejar o projeto em detalhe, o kanban, por outro lado, permite iniciar o desenvolvimento sem necessariamente ter um plano estruturado, não tendo a capacidade de organizar as tarefas em *sprints*. Ambas as abordagens possuem uma interface baseada em colunas, que permite visualizar o fluxo do trabalho.

## 2.2 Lean aplicado ao desenvolvimento de software

O conceito Lean, surgiu na Toyota, no Japão, após a II Guerra Mundial. O termo “lean” foi cunhado originalmente no livro “A Máquina que Mudou o Mundo (The Machine that Changed the World), de Womack, Jones e Roos, publicado em 1990.

O principal objetivo desta filosofia é agregar valor ao produto, tendo em vista a satisfação do cliente (interno e externo) e a eficiência. Esta filosofia assenta no aumento da eficiência mediante a eliminação consistente e completa dos desperdícios. É nesta base da eliminação dos desperdícios associados ao processo produtivo que assenta toda a filosofia lean, em conformidade com os seus princípios e práticas.

O conceito de desenvolvimento de software lean foi introduzido em 2003. Surgiu, assim, o conceito de Lean associado ao desenvolvimento de software, assentando em sete princípios basilares: (Poppendieck e Poppendieck, 2003)

- “Qualidade embutida” (*Build Quality in*) – controlar o processo de desenvolvimento para garantir que os defeitos não aconteçam. Existem algumas ferramentas de apoio a este princípio, tais como , *pair programming*, combinando o conhecimento de dois programadores em vez de um; desenvolvimento incremental e feedback constante; desenvolvimento orientado a testes, entre outros.
- Criação de conhecimento – criar conhecimento e partilhá-lo. Este princípio encoraja as equipas a fornecerem uma infraestrutura para documentar e reter adequadamente o que foi aprendido, por exemplo, através de documentação, formações, sessões de partilha de conhecimento e revisões de código (*code review*).
- Adiar o compromisso de decisão (*Defer commitment*) – apenas adotar decisões planeadas, evitando escolhas erradas e consequente desperdício. Este princípio Lean

incentiva a equipa a demonstrar responsabilidade, mantendo as opções em aberto até que estas sejam devidamente esclarecidas, em vez de tomar decisões sem a informação necessária.

- Entrega rápida – processo iterativo e incremental que permita um *feedback* constante do cliente, tornando o processo de alteração de requisitos mais flexível.
- Respeito pelas pessoas – as pessoas são a máquina por de trás do desenvolvimento do software. Respeitar e envolver os colaboradores, comunicar de uma forma proativa e eficaz, incentivando conflitos saudáveis, leva à deteção de oportunidades de melhoria e qualidade no produto desenvolvido.
- Otimizar o todo – um princípio lean transversal a qualquer setor. Consiste em melhorar numa perspetiva global, elevando a qualidade em toda a cadeia de valor.
- Eliminar desperdícios – tudo aquilo que não agrega valor ao produto deve ser eliminado, de forma a maximizar e otimizar o rendimento.

## 2.3 Garantia da qualidade

A qualidade pode ser traduzida no “grau de satisfação de requisitos dados por um conjunto de características intrínsecas”, (ISO 9000, 2005) consiste num objetivo estratégico que é estabelecido para fornecer produtos ou serviços uniformes que satisfaçam ou transponham as necessidades do cliente. De acordo com a ISO a garantia de qualidade faz parte da gestão da qualidade focada em fornecer confiança de que os requisitos serão cumpridos com qualidade.

A alta qualidade do processo de software está estritamente relacionada com o número de defeitos encontrados, e a qualidade dos processos afeta diretamente a qualidade dos produtos entregues. A qualidade pode ser garantida testando o produto de acordo com os padrões pré-definidos, com o objetivo do produto corresponder ao esperado pelo cliente. O processo de testes fornece consistência do sistema. Conforme a ISO, os seguintes parâmetros podem conduzir para a garantia de qualidade ao produto:

1. adquirir documentos que declaram os planos da organização para alcançar o nível de qualidade desejado;
2. produzir um plano que defina como obter a garantia de qualidade do produto;
3. organizar os recursos para implementar as estratégias definidas no plano;
4. avaliar se o produto/serviço proposto pela organização satisfaz as necessidades do cliente;
5. avaliar as operações, produtos e serviços da organização e determinar onde e quais são os riscos associados;

6. estabelecer se os planos da organização são adequados a nível do controlo, eliminação ou redução dos riscos identificados;
7. determinar até que ponto os planos da organização estão a ser implementados e os riscos contidos;
8. analisar se o produto/serviço cumpre as características expectadas.

Existem duas abordagens importantes usadas para determinar a qualidade do software: gestão de defeitos e definição dos atributos de qualidade, minimizando desta forma os riscos e entregar o sistema ao cliente com o mínimo de defeitos.

### 3. Caracterização do problema

Neste capítulo será feita uma breve apresentação da empresa onde o projeto foi desenvolvido, assim como a apresentação do problema que levou à escolha do tema da dissertação e a sua importância para a organização.

#### 3.1 Enquadramento empresarial

##### 3.1.1 A empresa

A *msg life* faz parte do grupo *msg*, um grupo internacional independente de empresas.

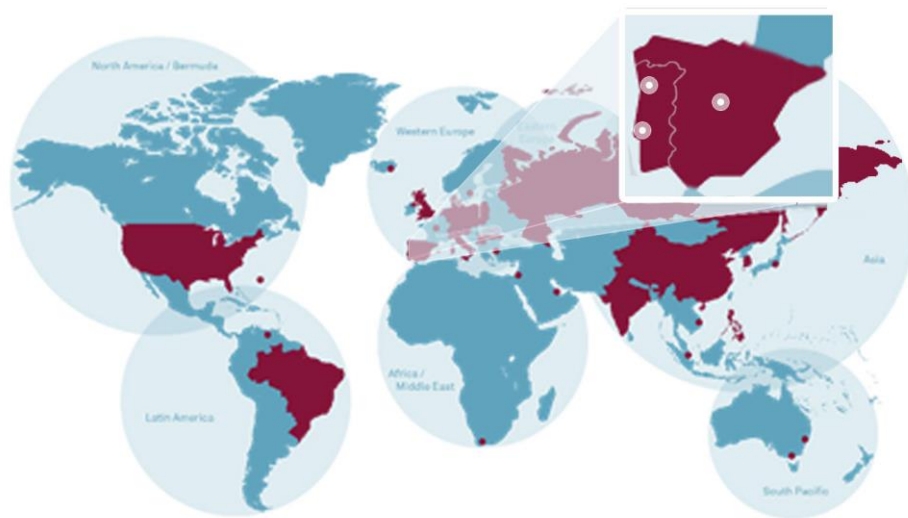


Figura 1: Presença global do grupo msg

Presente no mercado desde 1980, a *msg life* tem contribuído para a digitalização e inovação dos processos das seguradoras, solidificando, ano após ano, a sua forte presença no mercado global.

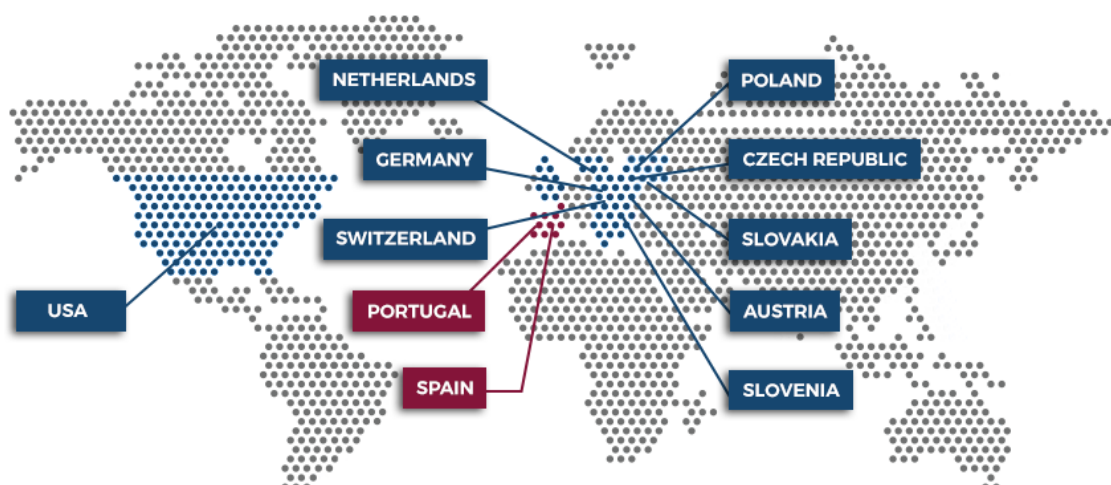


Figura 2: Rede msg life

A proposta de valor da *msg life* está focada na disponibilização de soluções tecnológicas disruptivas que agilizam o negócio das seguradoras em qualquer ramo de seguros – vida, não vida e saúde. O portfólio da *msg life* abrange *software*, consultoria e soluções *cloud* para companhias de seguros.

Parte integrante do grupo *msg life*, a *msg life Iberia* está sediada no Porto, possuindo ainda, escritórios em Lisboa e Madrid (Espanha).

Dentro da vasta gama de produtos, a *msg life Iberia* detém como produto core, o produto *msg.Sales&Services*. *msg.Sales&Services* é uma solução modular *web* e multicanal que potencia a estratégia centrada no cliente, com o objetivo de responder rapidamente às expetativas dos clientes e aumentar a agilidade e eficiência na criação de novos produtos e no lançamento de novas ofertas, agilizando a disponibilização de ofertas comerciais em distintos canais de venda e envolvendo toda a estrutura comercial da seguradora na captação de novos negócios e clientes.

### 3.1.2 Projetos

Na realidade da *msg life Iberia*, cada projeto é único, o âmbito do projeto é desencadeado através da perceção das necessidades reais do cliente, existindo assim diferentes níveis de customização. Foi identificado o leque de possibilidades apresentadas pela empresa de forma generalista (Figura 3).



Figura 3: Especificação dos projetos

Os projetos são de longa duração e representam para os clientes um forte investimento. A empresa tem atualmente a decorrer três projetos que englobam desenvolvimento. O projeto VHV e o projeto Prudential, desenvolvidos em parceria com a *msg life* da Alemanha e *msg life* dos Estados Unidos da América (FJA-US), respetivamente. Ambos consistem na integração do *msg.Sales&Services* com plataformas existentes. Paralelamente a estes, decorre na empresa um

projeto de investimento próprio que é contínuo, com o objetivo de inovar o produto e desenvolver as demonstrações para os clientes.

O departamento de desenvolvimento é formado por quatro equipas, com quatro a seis elementos cada. Uma equipa para cada projeto, excepto o projeto VHV que exigiu duas equipas, uma concentrada no desenvolvimento do *front-end* e outra do *back-end* do software.

### 3.2 Descrição do problema a tratar

A qualidade do software está estritamente relacionada com o número de defeitos/*bugs* encontrados. Ciente desta realidade, a *msg life Iberia* tem vindo a adotar melhores práticas para garantir a qualidade do produto e serviço. Contudo, pode-se considerar que ainda não foram suficientes para travar o número de bugs identificados no software produzido, quer seja na fase de desenvolvimento ou pós implementação no cliente. Como demonstrado na Tabela 1, o número de *bugs* tem vindo a ganhar vivamente importância.

*Tabela 1: Número de bugs encontrados anualmente*

| Ano  | Nº bugs encontrados |
|------|---------------------|
| 2015 | 689                 |
| 2016 | 2685                |
| 2017 | 2035                |

Importa, por isso, compreender as causas para este número elevado de *bugs* e identificar ações corretivas, contribuindo para a garantia da qualidade dos produtos.

Face aos objetivos mencionados anteriormente, pretende-se desenvolver um plano de testes de modo a serem detetadas falhas precocemente, e paralelamente avaliar as equipas de desenvolvimento em parâmetros considerados pertinentes com foco na elevação da qualidade do produto.

#### 3.2.1 Questões identificadas

Com o intuito de perceber com maior detalhe e profundidade o problema em questão, o trabalho foi realizado em conjunto com as unidades de desenvolvimento do produto e de implementação de projetos.

Foram desempenhadas algumas tarefas, como, o rastreamento dos *bugs* encontrados e também execução de testes, no sentido de acompanhar de perto uma parte do processo de desenvolvimento, para melhor percepção da situação corrente.

Existe um grande foco no desenvolvimento, negligenciando, de certa forma, a importância do rastreamento de *bugs* e da execução de testes consecutivos no decorrer do processo. Após uma análise feita à empresa através da observação no gembá e também da recolha de informação junto dos colaboradores, foi possível identificar os seguintes problemas principais:

Relativamente ao rastreamento de *bugs*,

- Como o registo dos *bugs* é feito recorrendo a um ficheiro excel partilhado no servidor da organização, não é possível a sua edição em simultâneo;
- Falta de controlo na tarefa de registar os *bugs*, alguns elementos não efetuam o registo ou não o fazem atempadamente;
- Os campos presentes no excel na maioria das vezes não são preenchidos na totalidade;
- Informação dispersa por vários documentos;
- Processo de registar os *bugs* não é ágil;
- O tratamento de dados é insuficiente.

No que diz respeito aos testes de *software*,

- Inexistência de suporte aos testes de *software* na maioria dos casos.

Com a elaboração de um plano de testes e a avaliação das equipas de desenvolvimento, em parâmetros centrados na eficácia e eficiência, visa-se minimizar as falhas no *software*, falhas na organização da documentação e dificuldade na obtenção de informação relativa ao rastreamento de *bugs*.

Face aos objetivos mencionados anteriormente, pretende-se avaliar as equipas de desenvolvimento em parâmetros considerados pertinentes, assim como construir uma metodologia de apoio aos testes de software que garanta a qualidade do produto.

## 4. Abordagens adotadas

Esta secção destina-se a expor as metodologias aplicadas no desenvolvimento do projeto, explicando as barreiras e as tomadas de decisão. A intervenção foi feita em duas áreas distintas, o rastreamento de *bugs* e testes de *software*.

### 4.1 Rastreamento de *bugs*

A avaliação de *bugs* com a investigação das suas causas é um processo essencial na perspetiva da empresa. Esta abordagem pretende identificar as causas subjacentes de um incidente, para que as soluções mais eficazes possam ser identificadas e implementadas. A análise da causa raiz ajuda a identificar o que aconteceu, como e porquê, para serem especificadas medidas corretivas que impeçam ocorrências semelhantes. Este tipo de análise vai ao encontro do princípio Lean “*Build quality in*”.

#### 4.1.1 Identificação de oportunidades de melhoria

As equipas da *msg life Iberia* recorrem ao *software* Jira para registar e acompanhar os bugs no software, designados por *tickets*. Posteriormente à correção dos *bugs*, estes deveriam ser registados num ficheiro excel partilhado na rede interna da empresa, atendendo aos seguintes campos:

- **ID**, identificação do *ticket* correspondente ao *bug* no Jira;
- **Summary**, breve descrição do problema;
- **Root cause**, determinação da causa do problema. Existe uma lista de possíveis causas para servir de diretriz, apresentada na tabela 2;

Tabela 2: Lista de causas raiz

| Root causes                                                                   |
|-------------------------------------------------------------------------------|
| Feature was working once but implementation of some other feature(s) broke it |
| Code was inadequate or incorrect — Code Review didn't catch it                |
| Not Applicable                                                                |
| Metadata Issues                                                               |
| Feature was not described in the requirements                                 |
| Feature was described in the requirements but not in the use case             |
| Feature was tested but the test was inadequate                                |
| Environment/Version Compatibility Issues                                      |
| Code was inadequate or incorrect — Code Review not held                       |
| User training                                                                 |
| Design inadequate or incorrect -- Review not held                             |
| To Be Determined                                                              |
| Design was inadequate or incorrect — Design Review didn't catch it            |

- **Comment**, descrição detalhada do “porquê” da ocorrência;
- **Sprint**, forma de identificar temporalmente a ocorrência.

O esforço para identificar a causa raiz de um bug não é um processo fácil. Normalmente, as equipas mantêm o foco na implementação da solução para o determinado *bug*, sem desprender tempo com documentação do processo de determinar a causa raiz.

Com o acompanhamento feito junto das equipas de desenvolvimento foi possível chegar as seguintes conclusões:

- a utilização do excel como ferramenta para elaborar este registo torna este processo pouco prático;
- usando o excel não é possível que duas pessoas editem o documento ao mesmo tempo, o que por vezes leva ao esquecimento deste registo, por não ter sido feito na “hora”;
- alguns elementos não efetuam o registo, não permitindo a análise das ocorrências;
- informação encontra-se dispersa por vários documentos;
- não determinam ações corretivas para combater as causas encontradas;
- a análise feita aos dados recolhidos é insuficiente, sendo só quantificados o total de *bugs* agrupados pelas diferentes causas raiz, por trimestre, sem qualquer termo comparativo, ou seja, não há uma investigação profunda.

Estas situações levam a que muitas das análises não sejam realizadas por falta de conteúdo, ou quando realizadas os resultados podem não ser totalmente realísticos.

#### 4.1.2 Abordagem adotada

No seguimento das falhas identificadas, pretendia-se encontrar uma solução que tornasse o processo mais ágil, mais prático para os elementos das equipas e que de certa forma “força-se” as equipas a seguirem o processo. Por outro lado, tornar a recolha e tratamento de dados mais fácil e menos exaustivo. No entanto, era necessário garantir que a mudança seria de adaptação fácil e que não significasse um grande investimento para a empresa.

Optou-se por recorrer a recursos já existentes na empresa. Desta forma, como o software JIRA já é utilizado para registar e acompanhar os bugs, decidiu-se estender esta ferramenta para o rastreamento dos bugs, ficando a informação compactada no Jira.

Encontra-se especificado um fluxo de trabalho para os *tickets* no Jira. Cada etapa corresponde a um estado em que se encontra o *ticket*, existindo transições para o que esse mesmo seja movido dependendo da fase em que se encontra. O fluxo de trabalho está definido da acordo com a Anexo A.

Desta forma, nas transições é exibido um ecrã para preenchimento de informação para dar seguimento ao problema até este ser fechado.

Neste seguimento, a solução de melhoria consiste em configurar esse ecrã quando o estado muda de *In progress* para *Resolved*. As alterações pretendidas são as seguintes:

1. Tornar obrigatório o preenchimento da *Root cause*, não sendo colocada nenhuma opção por *default*, garantindo que este campo é preenchido corretamente;
2. No grupo *Linked issues* adicionar o parâmetro *Originated by*, e torná-lo obrigatório quando a *Root cause* corresponder a “*feature was working once but implementation of some other feature(s) broke it*”, de modo a identificar o que originou o bug e a equipa responsável;
3. Tornar o campo “*Comment*” obrigatório.

Os dados posteriormente seriam coletados através do JIRA, pois é possível filtrá-los e extraí-los em vários formatos, nomeadamente em excel, para ser efetuado o tratamento de dados.

Este novo método assegura todos os campos exigidos anteriormente no ficheiro excel, pois os campos *ID*, *Summary* e *Sprint* já se encontram descritos nos tickets do Jira; os campos *Root cause* e *Comment* são preenchidos quando o ticket transita para o estado *Resolved*; e ainda foi acrescentado um campo *Originated by*, considerado pertinente para analisar com mais detalhe a origem dos *bugs*, sendo possível verificar o responsável individual/equipa.

Face aos problemas apresentados anteriormente todos são cobertos com esta solução, este novo método distingue-se pela simplificação do processo de rastreamento de *bugs* e do tratamento de dados.

De modo a perceber o impacto dos *bugs* no desempenho das equipas de desenvolvimento, foi definida uma nova métrica, o cumprimento do *sprint*. Como a empresa segue a metodologia Scrum, planeando de forma quinzenal os *sprints*, decidiu-se verificar o cumprimento dos *sprints*, consoante o planeado, analisando com detalhe o seu decorrer.

## 4.2 Testes de software

Um dos pilares para a garantia da qualidade do produto de *software* é o processo de teste. O objectivo principal de um teste de *software* consiste em garantir se a implementação deste cumpre todos os requisitos definidos pelo cliente, ou seja, comparar se o que foi acordado na fase de requisitos corresponde ao desenvolvido, paralelamente procura-se por erros no *software*.

Devido à complexidade dos *softwares* a atividade de testes deve ser efectuada ao longo de diferentes fases do processo, de acordo com as necessidades encontradas.

#### 4.2.1 Identificação de oportunidades de melhoria

Foi necessário efetuar em primeiro lugar um levantamento do estado atual, tanto do processo de testes como das ferramentas/métodos para a execução dos mesmos. Para tal, começou-se por estudar os diferentes *softwares* desenvolvidos pela empresa, de seguida as equipas foram observadas enquanto efetuavam os procedimentos normais, com principal foco os elementos responsáveis pelos testes, e numa fase posterior participou-se na atividade de testes, com o objetivo de perceber as dificuldades inerentes a este processo.

Através do acompanhamento intensivo do processo, foram retiradas as seguintes conclusões:

- informação/conhecimento dos cenários de teste centralizada nos responsáveis;
- o único documento de suporte aos testes é o documento de requisitos que inclui também breve descrição do comportamento esperado. No Anexo B é possível visualizar o aspeto de um dos ficheiros de requisitos;
- por vezes não são contemplados todos os cenários possíveis o que leva a falhas nos testes;
- não há informação necessária sobre os diferentes *softwares* para que um iniciante consiga proceder aos testes sem recorrer à ajuda de outros.

Assim sendo, é necessário desenvolver uma metodologia de apoio aos testes de *software*, culminando os problemas identificados anteriormente.

#### 4.2.2 Abordagem adotada

Com apoio no princípio lean criação de conhecimento, que menciona a importância de criar uma infraestrutura para documentar e reter adequadamente o conhecimento adquirido, a proposta consiste na criação de um plano de testes acessível a toda a organização.

Desta forma, recorrendo ao *Confluence* (*software de gestão de conteúdos*) que se encontra integrado com o Jira, elaborou-se documentação relativa a cada projeto, detalhando os cenários de testes de modo a abranger todos os requisitos do cliente.

Como a empresa segue uma metodologia ágil, várias versões do software são entregues no decorrer dos projetos, existindo a necessidade de dividir a informação por vários documentos de acordo com a versão, descrevendo os cenários de testes previstos para a dada versão. Assim

sendo, à medida que o projeto vai progredindo os cenários de testes vão sendo atualizados sendo criado o documento de testes para a próxima versão.

Os cenários de teste foram descritos com base nos documentos de requisitos já existentes na empresa, relatando passo a passo os vários cenários, estabelecendo as atividades que devem ser realizadas, as etapas que devem ser seguidas, a ordem cronológica da sua execução e os comportamentos esperados para cada ação. Para além dos cenários de teste, consta no documento, o(s) ambientes de teste e as credenciais de acesso aos mesmos.

Desta forma, o plano de testes serve de guia à execução e controlo de testes.

## 5. Resultados

Neste capítulo serão apresentadas as melhorias efetuadas, assim como demonstrados os resultados conseguidos com toda a informação recolhida. Os resultados serão expostos de acordo com a sequência apresentada no capítulo anterior.

### 5.1 Recolha de informação

Atendendo ao processo de rastreamento de bugs foi recolhida informação a partir dos ficheiros excel existentes, havendo a necessidade de abordar os elementos das equipas de modo a recuperar os dados em falta, assim como, a recolha facilitada após a implementação do novo método de registo dos *bugs* recorrendo ao *software* JIRA, sendo extraídos os dados diretamente para um ficheiro excel, para posterior tratamento e análise.

Nos Anexos C a F encontram-se expostos os dados recolhidos para cada equipa, através de um processo inicial completamente manual e numa segunda fase uma recolha de dados usufruindo das potencialidades do *software* JIRA, com uma pesquisa de informação com aplicação de filtros e de seguida a sua extração para um ficheiro excel, complementando os dados já adquiridos.

No que diz respeito à elaboração da nova métrica, cumprimento do *sprint*, numa primeira instância foram verificados os ficheiros excel que continham o planeamento dos sprints, de seguida esses dados foram agregados num novo ficheiro excel de forma incremental acompanhando o decorrer dos *sprints* e numa última fase foi verificado o cumprimento dos objetivos, *sprint* a *sprint*, recorrendo ao JIRA que permitiu confirmar o que realmente foi desenvolvido.

Paralelamente, foi desenvolvido um plano de testes para o projeto da VHV, sendo que este foi o único que foi possível acompanhar mesmo antes à entrega da primeira versão ao cliente.

O levantamento da informação relativa aos testes da VHV foi recolhida com base no documento de requisitos (ANEXO B), bem como a partilha de conhecimento por parte do responsável de testes deste projeto e simultâneamente foram executados testes ao software que permitiu detalhar passo a passo os vários cenários de teste, incluindo os comportamentos esperados para cada ação e também chamadas de atenção permitindo “guiar” a pessoa que vá testar o ambiente.

Para cada versão entregue ao cliente, foi elaborado um plano de testes face às funcionalidades entregues nessas mesmas fases. Com efeito, elaborou-se documentos de testes para as versões 1 e 2, designadas por P 0.1 e P 0.2, respetivamente. Atualmente, está a decorrer o desenvolvimento da versão P 0.3, o documento de testes respetivo a esta versão está a ser elaborado à medida que vão sendo desenvolvidas as funcionalidades previstas para a próxima entrega, tendo como ponto de partida o plano de testes da versão anterior, sendo feitas as alterações necessárias para que os cenários de teste estejam em concordância com o que vai ser entregue.

Foi tido em consideração o facto do cliente ser uma empresa alemã, o que levou à construção de um planos de testes realizado em inglês mas com a introdução dos conceitos da aplicação traduzidos para a língua alemã. Os planos agregam também os ambientes de teste e as credenciais de acesso ao sistema, estando desta forma concentradas num único documento todas as condições para testar os softwares.

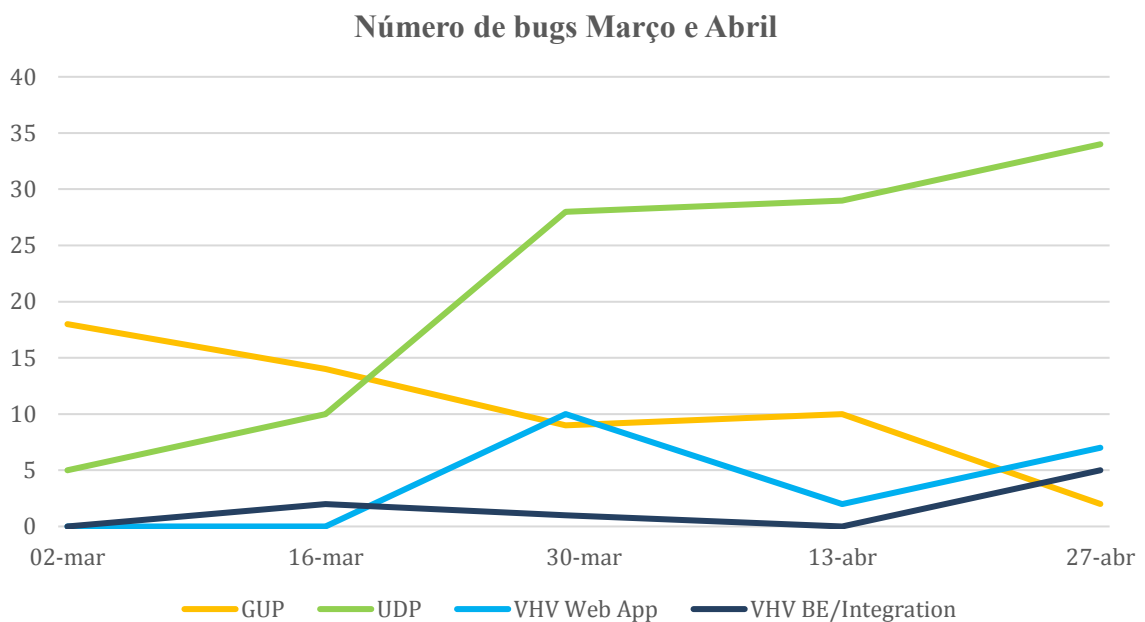
Os documentos de teste foram validados pelos responsáveis do processo e também pela execução de testes seguindo o plano, garantindo que estes era intuitivos e que abrangiam toda a complexidade do sistema.

## 5.2 Análise dos Bugs

### 5.2.1 Análise global

Para esta análise foi considerado o período de março e abril do ano corrente.

Através da avaliação relativa aos meses de março e abril, assente nos dados dos Anexos C a F, determinou-se que a equipa que apresenta mais bugs é a equipa da UDP, como se pode verificar na figura 4. Esse facto é justificado por esta equipa desenvolver demonstrações para o cliente e apostar na inovação do produto, ou seja, modificam constantemente o código *core* da empresa o que leva ao surgimento de bugs.



*Figura 4: Número de bugs Março e Abril*

As equipas VHV apresentam um número reduzido de bugs pois ainda estão numa fase inicial do projeto, reutilizando na maioria das vezes código já existente na empresa.

Aprofundando esta análise, estudaram-se as situações relativas a cada equipa, percebendo as causas para o surgimento dos bugs, assim como, a deteção da equipa que lhes deu origem, no caso do bug ter sido criado pela implementação/alterações de outra funcionalidade. O segundo formato consiste na avaliação das equipas individualmente no mesmo horizonte de tempo.

### 5.2.2 Equipa GUP

Tabela 3: Causas dos bugs da equipa GUP

| Root Cause                                                                    | Bugs      | Bugs (%) |
|-------------------------------------------------------------------------------|-----------|----------|
| To be determined                                                              | 13        | 25%      |
| Not Applicable                                                                | 3         | 6%       |
| Feature was working once but implementation of some other feature(s) broke it | 20        | 38%      |
| Code was inadequate or incorrect — Code Review not held                       | 6         | 11%      |
| Code was inadequate or incorrect — Code Review didn't catch it                | 6         | 11%      |
| Design inadequate or incorrect -- Review not held                             | 1         | 2%       |
| Design was inadequate or incorrect — Design Review didn't catch it            | 4         | 8%       |
| Feature was defined but not tested                                            | 0         | 0%       |
| Feature defined but not tested/correctly designed                             | 0         | 0%       |
| Developer didn't read properly the ticket                                     | 0         | 0%       |
| Environment/Version Compatibility Issues                                      | 0         | 0%       |
| Feature was described in the requirements but not in the use case             | 0         | 0%       |
| Feature was not described in the requirements                                 | 0         | 0%       |
| Feature was working but not the way described in the ticket                   | 0         | 0%       |
| Metadata Issues                                                               | 0         | 0%       |
| Installation or configuration issue                                           | 0         | 0%       |
| <b>TOTAL</b>                                                                  | <b>53</b> |          |

Como se pode verificar de acordo com a tabela 3, a causa da ocorrência de bugs nesta equipa deve-se maioritariamente ao facto de a funcionalidade que estava a funcionar corretamente ter sido quebrada pela implementação de outra, representando 38% da totalidade.

Destacando este facto, verificaram-se que equipas estiveram na origem destes bugs, de acordo com a figura 5.

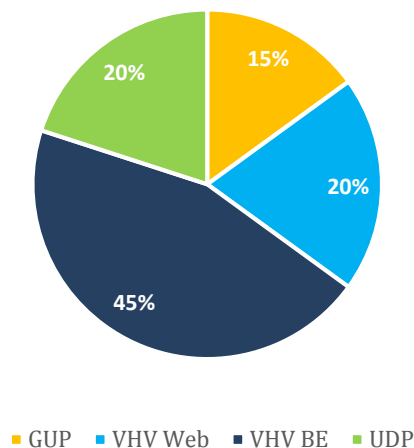


Figura 5: Gráfico das equipas que originaram os bugs

Conclui-se que a equipa que introduziu mais bugs na equipa GUP no período analisado, foi a equipa VHV BE, originando 45% dos bugs do tipo “*Feature was working once but implementation of some other feature(s) broke it*”.

É possível verificar que, ao contrário do esperado, 85% os bugs foram criados por elementos externos à equipa GUP, ou seja, as outras equipas estão a influenciar de forma negativa a performance desta. Verificou-se que esta situação ocorreu devido a alterações/restruturações de alto nível do código core, os elementos das outras equipas não possuem conhecimento suficiente sobre o *software desenvolvido pela GUP*, não sendo considerando alguns cenários possíveis ou até mesmo porque os programadores “não têm noção” que pode impactar o desenvolvimento da GUP e por último, alguns elementos não seguem o processo corretamente por consideraram que é demorado e complexo.

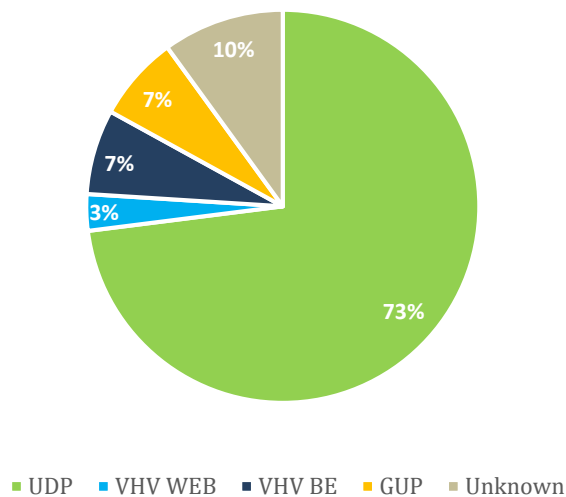
### 5.2.3 Equipa UDP

Tabela 4: Causas dos bugs da equipa UDP

| Root Cause                                                                    | Bugs       | Bugs (%) |
|-------------------------------------------------------------------------------|------------|----------|
| To be determined                                                              | 3          | 3%       |
| Not Applicable                                                                | 19         | 18%      |
| Feature was working once but implementation of some other feature(s) broke it | 41         | 40%      |
| Code was inadequate or incorrect — Code Review not held                       | 4          | 4%       |
| Code was inadequate or incorrect — Code Review didn't catch it                | 1          | 1%       |
| Design inadequate or incorrect -- Review not held                             | 0          | 0%       |
| Design was inadequate or incorrect — Design Review didn't catch it            | 0          | 0%       |
| Feature was defined but not tested                                            | 7          | 7%       |
| Feature defined but not tested/correctly designed                             | 22         | 21%      |
| Developer didn't read properly the ticket                                     | 0          | 0%       |
| Environment/Version Compatibility Issues                                      | 1          | 1%       |
| Feature was described in the requirements but not in the use case             | 1          | 1%       |
| Feature was not described in the requirements                                 | 2          | 2%       |
| Feature was working but not the way described in the ticket                   | 1          | 1%       |
| Metadata Issues                                                               | 1          | 1%       |
| Installation or configuration issue                                           | 0          | 0%       |
| <b>TOTAL</b>                                                                  | <b>103</b> |          |

Verifica-se que a causa mais incidente, representando 40%, é quando um bug é criado pela implementação de outra funcionalidade, o mesmo acontece na equipa GUP. Destaca-se também, com 21%, as falhas relacionadas com design incorreto que traduzem em falhas nos testes.

Seguindo o pensamento de verificar os responsáveis pela criação dos bugs, quando estes podem ser internos ou externos à equipa, representa-se de forma gráfica (Figura 6) as equipas que introduziam erros na equipa UDP.



*Figura 6: Equipas que causaram os bugs na equipa UDP*

Contrariamente ao que acontece na equipa GUP, neste caso os responsáveis pela introdução da maioria dos bugs, 73%, cuja a causa é a quebra pela implementação de outra funcionalidade, é a própria equipa. Desta forma, as outras equipas não criaram grande impacto nesta equipa.

Para esta análise são desconhecidos os responsáveis por 10 % dos bugs em questão, pois o método que era seguido não garantia que fosse feita a correspondência entre o bug e a equipa que o originou. Os colaboradores do departamento de desenvolvimento foram abordados para perceber a origem dos bugs, mas não foi possível estabelecer essa ligação devido ao tempo passado desde a sua deteção.

Entende-se que é necessário tomar medidas que reforcem o número de testes, com base nos factos apresentados.

### 5.2.4 Equipe VHV Web App

A equipe VHV Web App apresenta um número reduzido de bugs comparativamente às equipes GUP e UDP. É possível verificar na tabela 5, que praticamente metade dos bugs são provocados por falta de testes. São necessárias medidas para atacar essa vertente, aumentando a cobertura do software.

Tabela 5: Causas dos bugs da equipe VHV Web App

| Root Cause                                                                    | Bugs      | Bugs (%)   |
|-------------------------------------------------------------------------------|-----------|------------|
| To be determined                                                              | 2         | 11%        |
| Not Applicable                                                                | 2         | 11%        |
| Feature was working once but implementation of some other feature(s) broke it | 5         | 26%        |
| Code was inadequate or incorrect — Code Review not held                       | 0         | 0%         |
| Code was inadequate or incorrect — Code Review didn't catch it                | 0         | 0%         |
| Design inadequate or incorrect -- Review not held                             | 0         | 0%         |
| Design was inadequate or incorrect — Design Review didn't catch it            | 0         | 0%         |
| <b>Feature was defined but not tested</b>                                     | <b>9</b>  | <b>47%</b> |
| Feature defined but not tested/correctly designed                             | 0         | 0%         |
| Developer didn't read properly the ticket                                     | 1         | 5%         |
| Environment/Version Compatibility Issues                                      | 0         | 0%         |
| Feature was described in the requirements but not in the use case             | 0         | 0%         |
| Feature was not described in the requirements                                 | 0         | 0%         |
| Feature was working but not the way described in the ticket                   | 0         | 0%         |
| Metadata Issues                                                               | 0         | 0%         |
| Installation or configuration issue                                           | 0         | 0%         |
| <b>TOTAL</b>                                                                  | <b>19</b> |            |

A causa “*Feature was working once but implementation of some other feature(s) broke it*”, assume também relevância, representando 26% da totalidade. Nota-se que a equipe GUP não criou nenhuma incidência na equipe VHV Web App, e a responsabilidade da maioria dos bugs recai sobre as equipes da VHV (Figura 7).

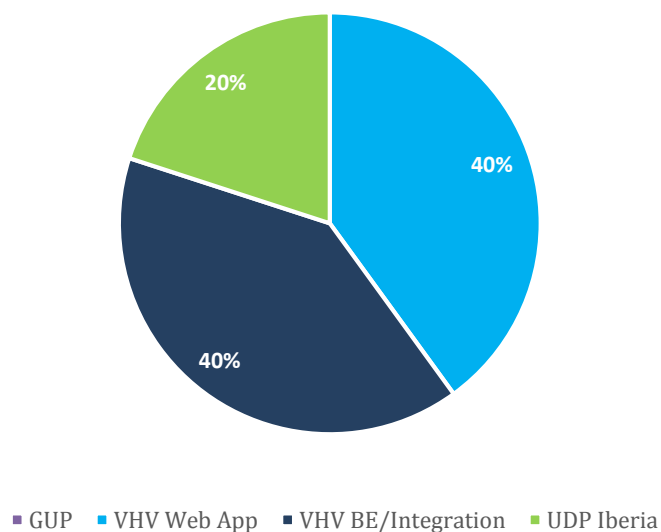


Figura 7: Equipes que causaram os bugs na equipe VHV Web App

### 5.2.5 Equipa VHV BE/Integration

A equipa VHV BE/Integration apresenta o menor número de bugs em relação às outras. As causas do mesmo incidiram no design incorreto, ou seja, falha na revisão do design e em problemas de instalação/configuração, como se pode verificar na tabela 6.

Table 6: Causas dos bugs da equipa VHV BE/Integration

| Root Cause                                                                    | Bugs     | Bugs (%) |
|-------------------------------------------------------------------------------|----------|----------|
| To be determined                                                              | 0        | 0%       |
| Not Applicable                                                                | 0        | 0%       |
| Feature was working once but implementation of some other feature(s) broke it | 1        | 14%      |
| Code was inadequate or incorrect — Code Review not held                       | 0        | 0%       |
| Code was inadequate or incorrect — Code Review didn't catch it                | 0        | 0%       |
| Design inadequate or incorrect -- Review not held                             | 0        | 0%       |
| Design was inadequate or incorrect — Design Review didn't catch it            | 2        | 29%      |
| Feature was defined but not tested                                            | 0        | 0%       |
| Feature defined but not tested/correctly designed                             | 1        | 14%      |
| Developer didn't read properly the ticket                                     | 0        | 0%       |
| Environment/Version Compatibility Issues                                      | 1        | 14%      |
| Feature was described in the requirements but not in the use case             | 0        | 0%       |
| Feature was not described in the requirements                                 | 0        | 0%       |
| Feature was working but not the way described in the ticket                   | 0        | 0%       |
| Metadata Issues                                                               | 0        | 0%       |
| Installation or configuration issue                                           | 2        | 29%      |
| <b>TOTAL</b>                                                                  | <b>7</b> |          |

O único bug cuja causa é “*Feature was working once but implementation of some other feature(s) broke it*”, foi introduzido pela própria equipa. Conclui-se que elementos externos a esta equipa não causaram bugs na equipa VHV BE, o contrário não se verifica.

Pretende-se com esta análise comparativa, com foco no rastreamento dos bugs, que o departamento de desenvolvimento elabore um plano de ações corretivas, de modo a eliminar as causas identificadas. A melhoria será alcançada através do adequado tratamento, que tem início com o rastreamento dos bugs, que através do novo método, recorrendo ao JIRA, é totalmente assegurado.

## 5.3 Análise Sprints

### 5.3.1 Análise global

Foram analisados os planeamentos das equipas, exceto para a equipa UDP que segue atualmente a metodologia Kanban, não elaborando desta forma um planeamento.

Esta abordagem permite avaliar a posição das equipas, no que diz respeito ao cumprimento do sprint definido a cada 15 dias. Este formato foi desenvolvido contemplando os primeiros quatro meses do ano.

Numa perspetiva global, verifica-se que as equipas GUP, VHV WEB e VHV BE, têm uma previsibilidade de entrega conforme o planeado cerca de 40%, 85% e 95%, respetivamente, como se pode concluir através da figura 8 (não existiam dados do planeamento para alguns períodos).

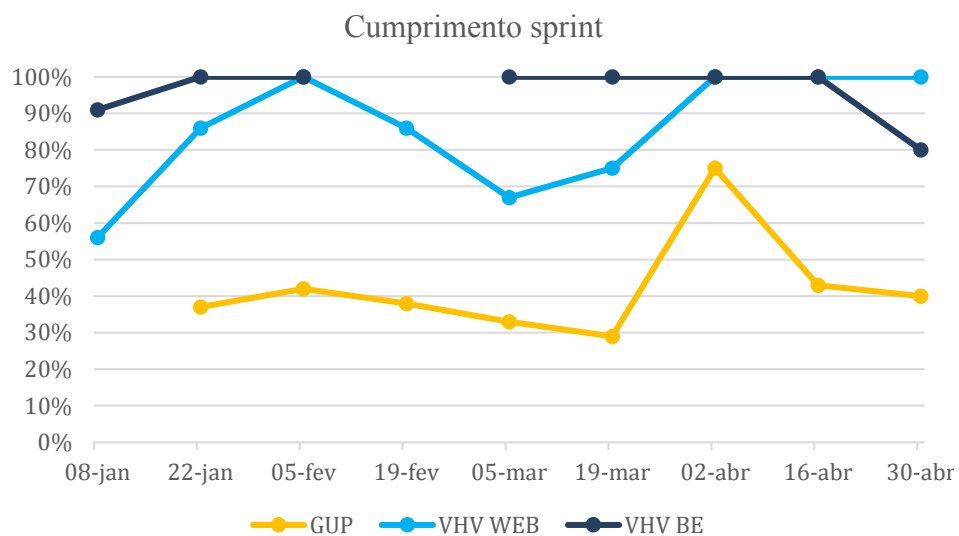
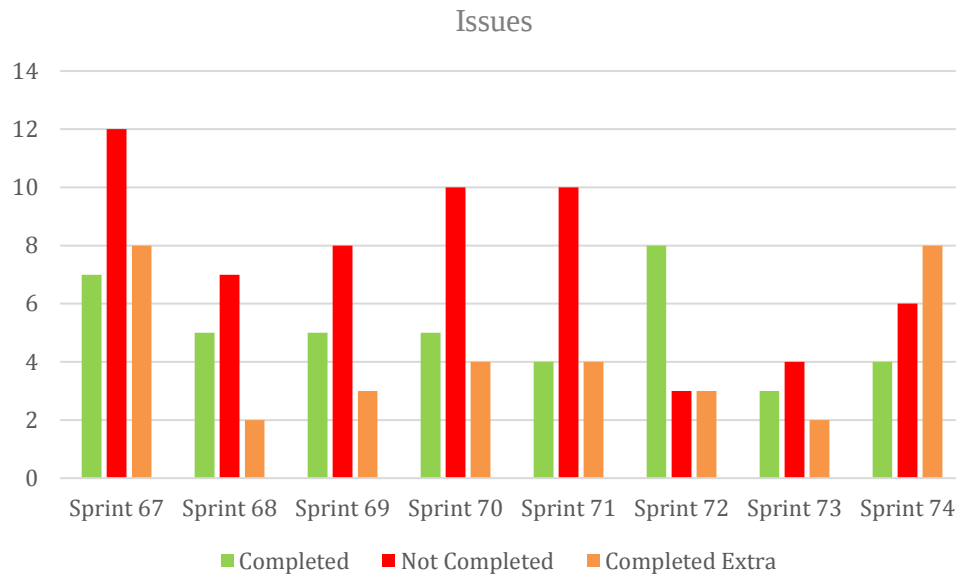


Figura 8: Gráfico cumprimento sprint

De modo a perceber a discrepância entre a equipa de GUP e as outras, analisou-se sprint a sprint, demonstrando o que foi desenvolvido por cada equipa. O primeiro formato permite perceber o que foi concluído ou não de acordo com o planeado, assim como o que foi desenvolvido mesmo não estando no planeamento. O segundo tem como objetivo distinguir o que foi desenvolvido paralelamente, se foi uma funcionalidade nova (*story*), um *bug*, um pedido especial do cliente/cliente interno (*task*) ou documentação (*documentation*).

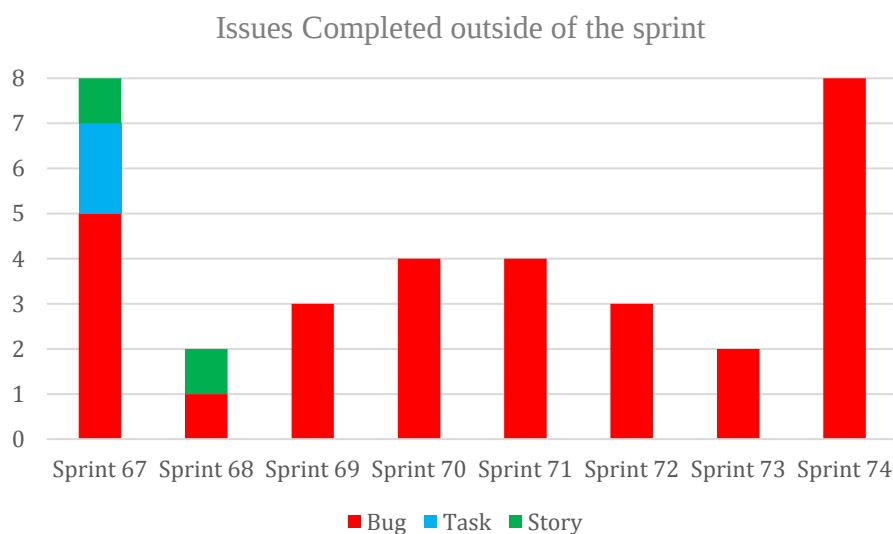
### 5.3.2 Equipe GUP



*Figura 9: Percepção global dos sprints da equipa GUP*

Entende-se através da figura 9, que o cumprimento dos sprints pode ter sido comprometido devido à existência de elementos paralelos para concluir.

Analisando o que foi desenvolvido fora do contexto do planeamento, de acordo com a figura 10, conclui-se que a correção dos bugs que foram surgindo foi considerada prioridade relativamente a algumas tarefas que tinham sido planeadas. A ocorrência de bugs impediu que a equipa completa-se o *sprint*. De destacar que as outras equipas tiveram de certa forma influência no desempenho da equipa GUP, visto que introduziram 17 bugs nesta equipa.



*Figura 10: Tipo de tarefas concluídas fora do planeado*

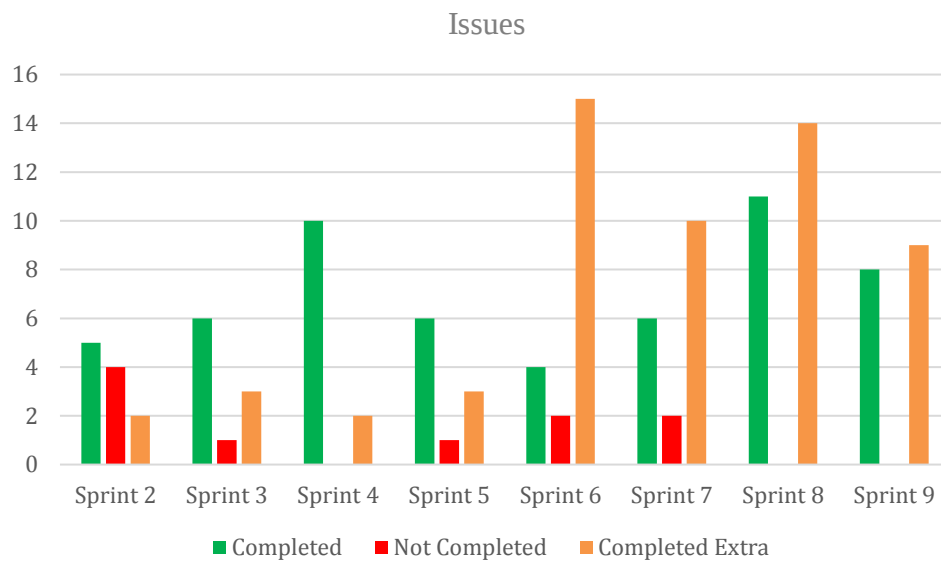
*Tabela 7: Datas sprints GUP*

|                  | From       | To         |
|------------------|------------|------------|
| <b>Sprint 66</b> | 08-01-2018 | 19-01-2018 |
| <b>Sprint 67</b> | 22-01-2018 | 03-02-2018 |
| <b>Sprint 68</b> | 05-02-2018 | 16-02-2018 |
| <b>Sprint 69</b> | 19-02-2018 | 02-03-2018 |
| <b>Sprint 70</b> | 05-03-2018 | 16-03-2018 |
| <b>Sprint 71</b> | 19-03-2018 | 29-03-2018 |
| <b>Sprint 72</b> | 02-04-2018 | 13-04-2018 |
| <b>Sprint 73</b> | 16-04-2018 | 27-04-2018 |
| <b>Sprint 74</b> | 30-04-2018 | 11-05-2018 |

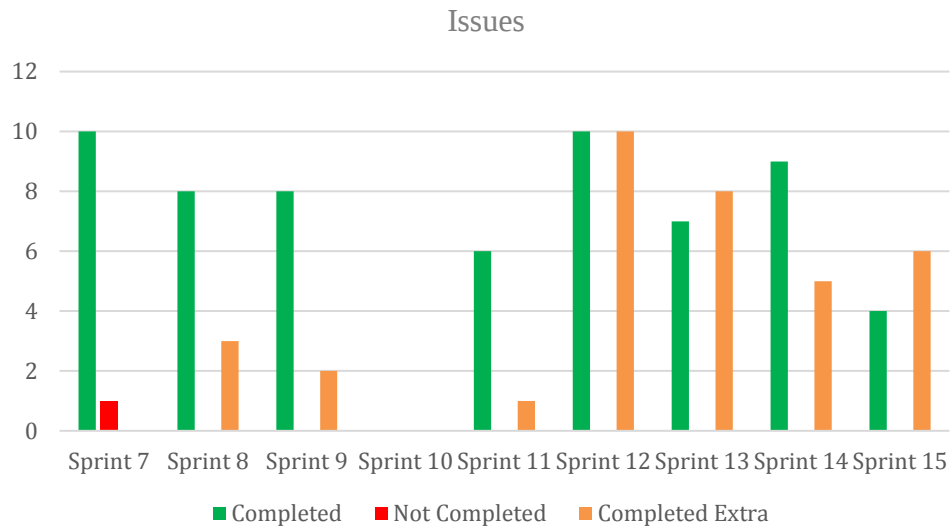
Nota: Não existia planeamento para o sprint 66.

### 5.3.3 Equipas VHV

Analisando os sprints das duas equipas da VHV, nota-se que na maioria das vezes o planeamento é cumprindo, sobrando tempo extra para realizar tarefas que não foram planeadas, como se pode verificar nas figuras 11 e 12. Pode-se concluir que as equipas não estão a estimar corretamente, levando a que as equipas a desenvolver mais daquilo que era previsto.



*Figura 11: Perceção global dos sprints da equipa VHV Web App*



*Figura 12: Percepção global dos sprints da equipa VHV BE/Integration*

No Anexo G é possível comprovar que as equipas da VHV planeam de forma incorreta, pois o trabalho realizado fora do objetivo inicial do sprint consiste maioritariamente no desenvolvimento de novas funcionalidades, não necessariamente tempo despendido a corrigir bugs.

## 5.4 Plano de testes

A elaboração dos planos de teste para o projeto da VHV foi pensado de forma a simplificar o processo de testes por parte dos responsáveis mas também dos programadores que em colaboração testam a aplicação antes da entrega ao cliente, de modo a reforçar que todos os erros que o sistema possa apresentar possam ser corrigidos antes da implementação no cliente. Esta abordagem consiste num conjunto de diretrizes para cobrir todos os cenários de testes, garantindo que os requisitos sejam cumpridos na totalidade, com o foco na satisfação do cliente, agregando mais qualidade ao produto.

Os documentos de teste para as duas versões do projeto, agregam a ligação para o ambiente de testes e as credenciais de acesso ao mesmo, como se pode verificar nos Anexos H e I. A construção baseou-se nos requisitos definidos para cada versão, sendo detalhado passo a passo a forma como proceder os testes e qual o comportamento esperado para cada cenário, exigindo o investimento consecutivo a testes ao sistema para que fossem cenários intuitivos e fáceis de compreender e ao mesmo tempo que realmente fosse detalhado ao promenor o modo de funcionamento do software, para que não existissem dúvidas no momento de testar. . Desta forma, mesmo os elementos que não tenham conhecimento sobre o software desenvolvido para a VHV, serão capazes de testar a aplicação recorrendo apenas aos documentos produzidos, sendo uma fonte verdadeira de conhecimento sobre a aplicação.

Nos Anexos H e I encontram-se os planos de teste produzidos para a versão 1 e 2 do software desenvolvido para a VHV

Estes documentos foram criados com o intuito de promover um maior nível de qualidade do produto no momento de entrega ao cliente e reduzir o potencial de *bugs*.

## 6 Conclusões e perspectivas de trabalho futuro

### 6.1 Conclusões

A avaliação de desempenho das equipas de desenvolvimento, no que diz respeito ao número de *bugs* e ao cumprimento dos *sprints*, permite avaliar continuamente a posição e evolução das equipas. Esta metodologia permite o acompanhamento eficaz de cada área do processo e o novo formato de registo das ocorrências de *bugs* promove o investimento no estudo das causas dos erros no software e a transparência dos resultados na organização.

Os modelos de análise desenvolvidos neste projeto constituem agora uma ferramenta complementar de avaliação do desempenho das equipas aquando a definição dos objetivos para cada trimestre, servindo de inputs aos novos ciclos. Estes modelos permitem:

- a análise da evolução do desempenho, relativamente ao número de bugs;
- a identificação das causas mais frequentes dos bugs criados;
- a análise do cumprimento dos objetivos em cada sprint e comparativamente com os sprints anteriores;
- a análise do impacto dos bugs criados no desempenho das equipas de desenvolvimento.

Adicionalmente, na definição do novo método de rastreamento das causas dos bugs foi introduzida uma componente até agora não monitorizada, as equipas que originaram os bugs. Numa abordagem global, permitirá, numa primeira fase, identificar ações de melhoria nos modelos processuais que cada equipa segue, contribuindo também para a melhoria do desempenho dos colaboradores.

Esta avaliação, através dos modelos de análise desenvolvidos, servirá ainda como barómetro de eficácia para as ações de melhoria definidas para cada trimestre, determinando a pertinência de definir ações adicionais para as áreas do processo com desempenhos inferiores ou até mesmo influenciar mudanças processuais na empresa.

A introdução de uma nova metodologia de apoio aos testes de software resulta numa simplificação deste processo, tornando-o mais fácil, agora com documentos que suportam os testes, não sendo necessário o esforço contínuo dos responsáveis de teste para transmitirem conhecimento de cada detalhe da aplicação. Esta ferramenta permite agora analisar mais facilmente todos os cenários dos sistemas.

Um plano de teste garante que todos os aspetos do software sejam testados e cobertos, no momento em que os testes sejam realizados de maneira estruturada e seguindo o plano previamente detalhado.

Os resultados deste projeto foram validados juntamente com os colaboradores da empresa, mostrando ser de grande utilidade e uma mais valia para os utilizadores.

## **6.2 Perspetiva de desenvolvimentos futuros**

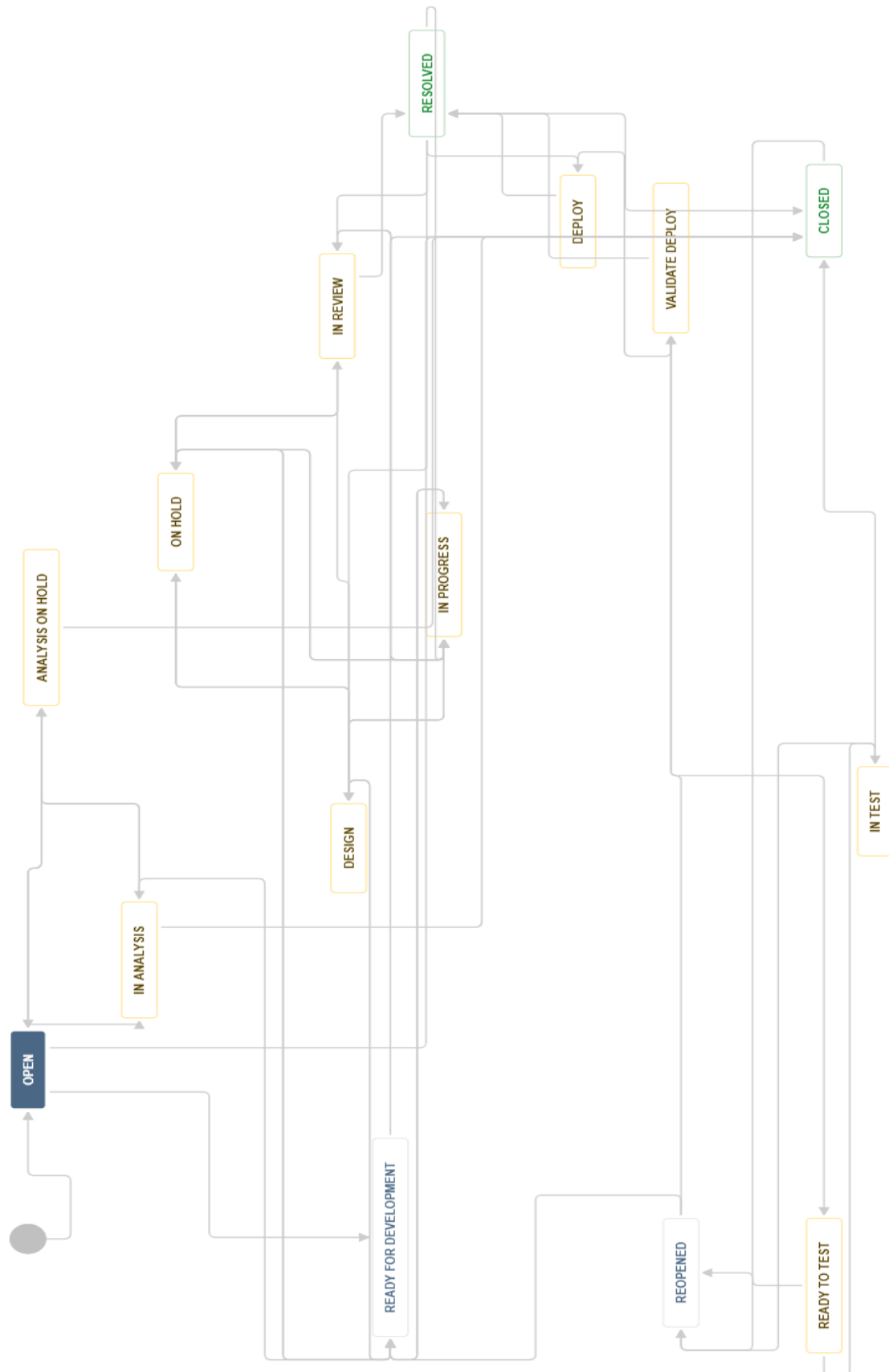
Como trabalhos futuros, poderia ser aplicada uma nova forma de planeamento, nomeadamente introduzir um planeamento deslizando que se caracteriza por reduzir o ciclo de revisão do planeamento, aumentando assim a previsibilidade do que será concluído no final de cada *sprint*, não comprometendo a entrega do produto ao cliente.

Na perspetiva dos testes de software, ponderar a automatização dos testes, para evitar trabalho manual em excesso, garantir que não há regressões no software, obter um feedback mais rápido, economizar tempo executando testes repetidos. Este tipo de mudança carece de um investimento elevado por parte da empresa, mas num contexto ágil a automação de testes é ainda mais pertinente, pois estes auxiliam um *feedback* contínuo e rápido, além de garantir uma maior qualidade na entrega do software.

## Referências

- Beck, Kent, Beedle, M., Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., Grenning, J., Highsmith, J., Hunt, A., Jeffries, R., Kern, J., Marick, B., Martin, R., Mellor, S., Schwaber, K., Sutherland, J. and Thomas, D. 2001, Manifesto for Agile Software Development. <http://agilemanifesto.org/>
- Highsmith J., Orr K., Cockburn A., 2000, Extreme programming, E-Business Application Delivery.
- Holey, D. 2001, ISO 9000 Quality Systems Handbook, 4th edn, Butterworth Heinmann, Oxford.
- Humble, J., Molesky, J., O'Reilly, B. 2015, Lean Enterprise: How High Performance Organizations Innovate at Scale, O'Reilly Media, Inc, USA.
- Lacey, M. 2015, The Scrum Field Guide: Agile Advice for Your First Year and Beyond, 2<sup>nd</sup> edn, Pearson Education, USA.
- Poppendieck, M. and Poppendieck T. 2006, Implementing Lean Software Development: From Concept to Cash, Pearson Education, USA.
- Stiner, S. 2017, The 12 Daily Principles Of Agile Software Development, Forbes Council. <https://www.forbes.com/sites/forbestechcouncil/2017/03/20/the-12-daily-principles-of-agile-software-development/>

## ANEXO A: Fluxo de trabalho JIRA



36

## ANEXO C: Excerto ficheiro excel equipa GUP

[illegible]

ANEXO D: Excerto ficheiro excel equipa UDP

| Iberia_Reopened_and_Bugs_Tracking.xlsx - Excel                           |  |  |  |  |  |  |  |  |  |  |  |  |  | Oliveria Christiana |  |  |  |  |  |  |  |  |  |  |  |
|--------------------------------------------------------------------------|--|--|--|--|--|--|--|--|--|--|--|--|--|---------------------|--|--|--|--|--|--|--|--|--|--|--|
| FILE HOME INSERT PAGE LAYOUT FORMULAS DATA REVIEW VIEW DEVELOPER ADD-INS |  |  |  |  |  |  |  |  |  |  |  |  |  |                     |  |  |  |  |  |  |  |  |  |  |  |
| A B C D E F G H I J K L M N O P Q                                        |  |  |  |  |  |  |  |  |  |  |  |  |  |                     |  |  |  |  |  |  |  |  |  |  |  |
| Data Inicio                                                              |  |  |  |  |  |  |  |  |  |  |  |  |  |                     |  |  |  |  |  |  |  |  |  |  |  |
| SPRINT TYPE                                                              |  |  |  |  |  |  |  |  |  |  |  |  |  |                     |  |  |  |  |  |  |  |  |  |  |  |
| ID                                                                       |  |  |  |  |  |  |  |  |  |  |  |  |  |                     |  |  |  |  |  |  |  |  |  |  |  |
| ROOT CAUSE                                                               |  |  |  |  |  |  |  |  |  |  |  |  |  |                     |  |  |  |  |  |  |  |  |  |  |  |
| COMMENT                                                                  |  |  |  |  |  |  |  |  |  |  |  |  |  |                     |  |  |  |  |  |  |  |  |  |  |  |
| CATEGORY                                                                 |  |  |  |  |  |  |  |  |  |  |  |  |  |                     |  |  |  |  |  |  |  |  |  |  |  |
| ROOT CAUSE TICKET                                                        |  |  |  |  |  |  |  |  |  |  |  |  |  |                     |  |  |  |  |  |  |  |  |  |  |  |
| RESPONSIBLE                                                              |  |  |  |  |  |  |  |  |  |  |  |  |  |                     |  |  |  |  |  |  |  |  |  |  |  |
| 1                                                                        |  |  |  |  |  |  |  |  |  |  |  |  |  |                     |  |  |  |  |  |  |  |  |  |  |  |
| 2                                                                        |  |  |  |  |  |  |  |  |  |  |  |  |  |                     |  |  |  |  |  |  |  |  |  |  |  |
| 3                                                                        |  |  |  |  |  |  |  |  |  |  |  |  |  |                     |  |  |  |  |  |  |  |  |  |  |  |
| 4                                                                        |  |  |  |  |  |  |  |  |  |  |  |  |  |                     |  |  |  |  |  |  |  |  |  |  |  |
| 5                                                                        |  |  |  |  |  |  |  |  |  |  |  |  |  |                     |  |  |  |  |  |  |  |  |  |  |  |
| 6                                                                        |  |  |  |  |  |  |  |  |  |  |  |  |  |                     |  |  |  |  |  |  |  |  |  |  |  |
| 7                                                                        |  |  |  |  |  |  |  |  |  |  |  |  |  |                     |  |  |  |  |  |  |  |  |  |  |  |
| 8                                                                        |  |  |  |  |  |  |  |  |  |  |  |  |  |                     |  |  |  |  |  |  |  |  |  |  |  |
| 9                                                                        |  |  |  |  |  |  |  |  |  |  |  |  |  |                     |  |  |  |  |  |  |  |  |  |  |  |
| 10                                                                       |  |  |  |  |  |  |  |  |  |  |  |  |  |                     |  |  |  |  |  |  |  |  |  |  |  |
| 11                                                                       |  |  |  |  |  |  |  |  |  |  |  |  |  |                     |  |  |  |  |  |  |  |  |  |  |  |
| 12                                                                       |  |  |  |  |  |  |  |  |  |  |  |  |  |                     |  |  |  |  |  |  |  |  |  |  |  |
| 13                                                                       |  |  |  |  |  |  |  |  |  |  |  |  |  |                     |  |  |  |  |  |  |  |  |  |  |  |
| 14                                                                       |  |  |  |  |  |  |  |  |  |  |  |  |  |                     |  |  |  |  |  |  |  |  |  |  |  |
| 15                                                                       |  |  |  |  |  |  |  |  |  |  |  |  |  |                     |  |  |  |  |  |  |  |  |  |  |  |
| 16                                                                       |  |  |  |  |  |  |  |  |  |  |  |  |  |                     |  |  |  |  |  |  |  |  |  |  |  |
| 17                                                                       |  |  |  |  |  |  |  |  |  |  |  |  |  |                     |  |  |  |  |  |  |  |  |  |  |  |
| 18                                                                       |  |  |  |  |  |  |  |  |  |  |  |  |  |                     |  |  |  |  |  |  |  |  |  |  |  |
| 19                                                                       |  |  |  |  |  |  |  |  |  |  |  |  |  |                     |  |  |  |  |  |  |  |  |  |  |  |
| 20                                                                       |  |  |  |  |  |  |  |  |  |  |  |  |  |                     |  |  |  |  |  |  |  |  |  |  |  |
| 21                                                                       |  |  |  |  |  |  |  |  |  |  |  |  |  |                     |  |  |  |  |  |  |  |  |  |  |  |
| 22                                                                       |  |  |  |  |  |  |  |  |  |  |  |  |  |                     |  |  |  |  |  |  |  |  |  |  |  |
| Total_VHV_UDP                                                            |  |  |  |  |  |  |  |  |  |  |  |  |  |                     |  |  |  |  |  |  |  |  |  |  |  |
| Bugs                                                                     |  |  |  |  |  |  |  |  |  |  |  |  |  |                     |  |  |  |  |  |  |  |  |  |  |  |
| Analysis                                                                 |  |  |  |  |  |  |  |  |  |  |  |  |  |                     |  |  |  |  |  |  |  |  |  |  |  |
| Global                                                                   |  |  |  |  |  |  |  |  |  |  |  |  |  |                     |  |  |  |  |  |  |  |  |  |  |  |

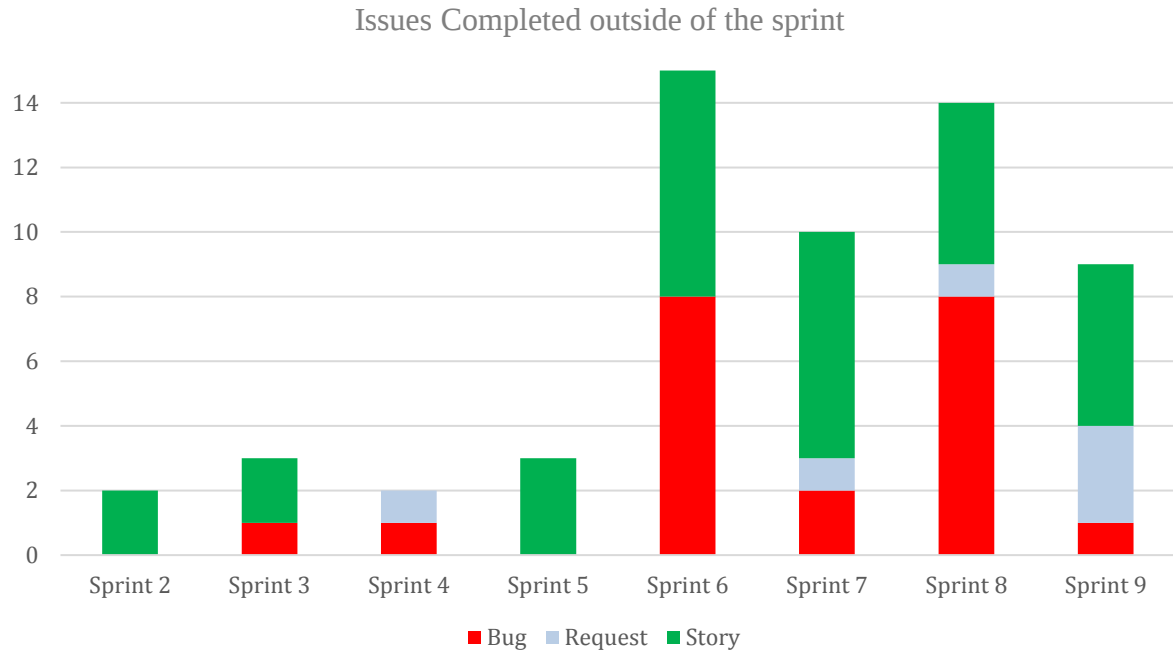


**ANEXO F: Excerto ficheiro excel equipa VHV BE/Integration**

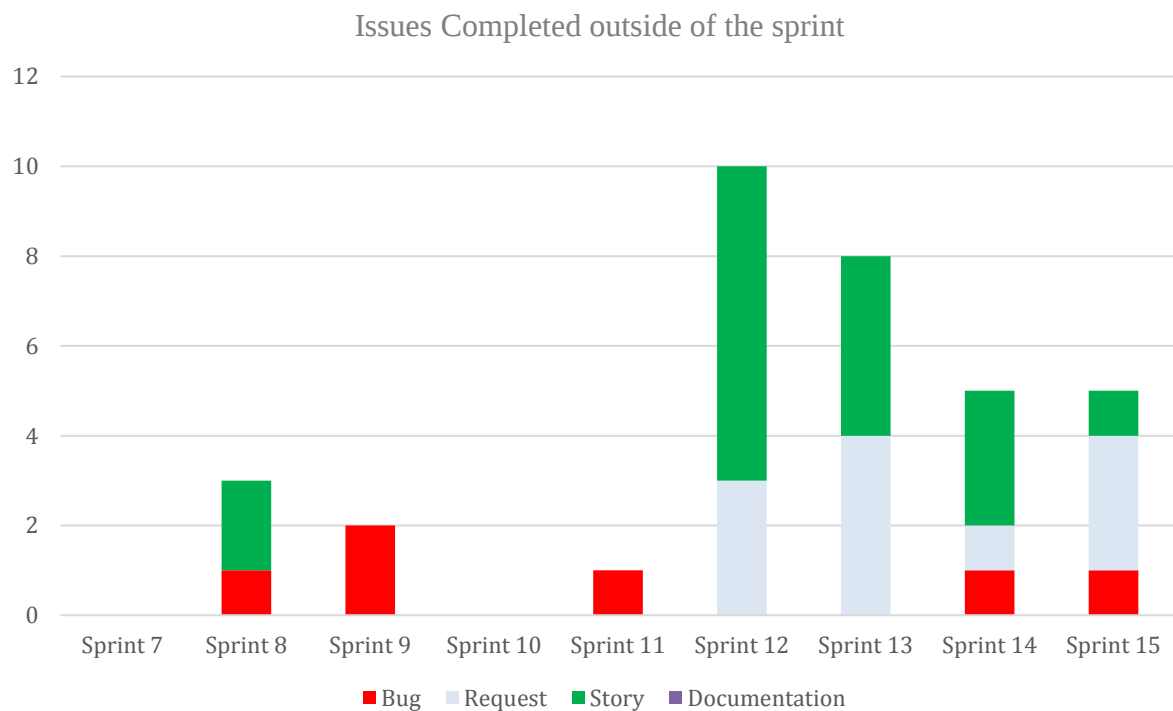
[illegible]

## ANEXO G: Gráficos relativos ao tipo de tarefas concluídas fora do planeado

### Equipa VHV Web App



### Equipa VHV BE/Integration



## ANEXO H: Plano de testes VHV versão P 0.1

### VHV Release Test 1.0.1 (P 0.1)

#### Introduction

- test primarily in Google Chrome
- but also on IE 10+ and Firefox
- test both in English and German browser settings

#### Environments

<http://le-s-po-w2012r2:20188/cid/Login.do> (CID QA Environment)

#### Log in

User:lf4

Password:lf4

#### Sales and Service

##### 1. Authentication & Authorization

1. Access "Insurance Suite" log in page.
2. Log in to "Insurance Suite" with valid credentials.  
>> S&S menu is displayed with all the options: "Offer", "Search partner", "My opportunities" "Search opportunities", "New opportunity", "My marketable offers" and "New marketable offer".

##### 2. Search Offer

1. Click in the option "Offer" in S&S.  
>> A new tab is opened with the corresponding screen.
2. Check if the following search fields are available:
  - a. Opportunity ID "Angebots/Antrags-Nr"
  - b. Create Date "Erstelldatum"
  - c. Update Date "Letzte Änd."
  - d. Status
  - e. Partner-number "Partner-Nummer"
3. Click button "Suchen" to perform search:
  - a. Using Opportunity ID
  - b. Using Create Date
  - c. Using Update Date
  - d. Using Status
  - e. Using Partner-Number
  - f. Using some combinations ( e.g. Opportunity ID and Create Date; Status and Partner-Number)
4. Check if all contracts are listed according to the entered criteria(s), a) to g)  
>> In the list of results the system displays contracts, i.e. if an opportunity has 3 contracts and all 3 meet the search criteria, then 3 rows should be displayed.
5. Check if the following columns are displayed in the list:
  - a. Actions "Aktionen"
    - i. Open "Angebot/Antrag beaufschaffen"
    - ii. Bündel
    - iii. Copy "Eine kopie dem Bündel Hinzufügen"
  - b. Insured person name "Vollständiger Name"
  - c. Policy holder name "Vollständiger Name"
  - d. Tarif "Tarif"
  - e. Sum insured "Leistg. Erleben (abs.)"

- f. Effective date "Versicherungsbeginn"
- g. Opportunity status "Vertrags status"
- h. Update date "Aktualisierungsdatum"
- 6. Check actions for each contract.
  - a. Click open/edit quote icon
    - >> Contract should be opened in the summary page
  - b. Click Bündel icon
    - >> Bundle screen should be displayed
  - c. Click copy icon
    - >> Product selection screen is displayed allowing the user to create a new offer based on the data from the previous one
- 7. Click New Opportunity button "Neues Angebot"
  - >> Product Selection Screen should be displayed

### 3. Search Partner (Innovas Integration)

- 1. Click in the option "Partnersuche" in S&S.
  - >> A new tab is opened with the corresponding screen.
- 2. Check if the following search fields are available:
  - a. Name "Zweitname"
  - b. Surname "Vorname"
  - c. Postal code "Postleitzahl"
  - d. Town "Stadt"
  - e. Street "Straße"
  - f. Date of birth "Geburtsdatum"
  - g. Partner id "Customer Id"
- 3. Click search "Suchen" to perform search:
  - b. Using Zweitname
  - c. Using Vorname
  - d. Using Postleitzahl
  - e. Using Stadt
  - f. Using Straße
  - g. Using Geburtsdatum
  - h. Using Customer Id
  - i. Using some combinations ( e.g. Zweitname and Vorname; Straße and Customer Id)
- 4. Check if all contracts are listed according to the criteria(s) entered, a) to g)
- 5. Check if the following columns are displayed in the list:
  - a. Actions "Aktionen"
    - i. Search quotes for partner
    - ii. Create opportunity for customer
  - b. Name "Zweitname"
  - c. Surname "Vorname"
  - d. Date of birth "Geburtsdatum"
  - e. Street "Straße"
  - f. Postal code "Postleitzahl"
  - g. Town "Stadt"
  - h. Partner id "Customer Id"
- 6. Check actions for each partner:
  - a. Click "Search quotes for partner" icon
    - >> Open search offer screen in the same tab and displays the list of contracts in which the partner exists as policy holder or Insured person
  - b. Click "Create opportunity for customer" icon
    - >> Open the product selection screen in the same tab allowing the user to create a new offer; partner will be pre-filled as policy holder;
- 7. Click new partner "Kunden anlegen" button
  - >> A new tab is opened with msg.Innovas

### 4. Offer screens

- 1. Access the search offer option.
- 2. Click new opportunity "Neues Angebot".
- 3. Select an effective date.
  - >> 8 Possible dates: first day of the previous month, current month and next 6 months.
- 4. Check the list of products displayed
  - >> Table including single products and marketable offers.
- 5. Select the product by clicking in the respective row.

- >> The row is highlighted; Only product HL\_T\_201810 and corresponding marketable offers are available;
- 6. Click in one of the following options
  - a. "OK"
  - >> Offer screen is displayed.
  - b. Cancel "Abbrechen"
  - >> Search offer screen is displayed.
- 7. (Considering 6. a.) Offer screen is presented, consisting on a wizard with 2 steps: "Basisdaten" and "Summary".
- 8. In the first screen the following sections are displayed:
  - a. Basic data "Daten Allgemein"
    - >> The effective date selected in the product selection screen is copied to the begin date of the opportunity and appears read only
    - >> The dropdown "Tarif" is not displayed if a marketable offer is selected.
  - b. Partner data "Versicherungsnehmer"
    - >> The checkbox "VN=VP1" is selected by default;
    - >> The "gender" and "date of birth" of the insured are editable when there is no partner selected as insured person;
    - >> All the data provided by Innovas is read only;
  - c. Main tariff data "Daten Haupttarif":
    - >> The fields displayed depend on the value of the field "Tarif" of the basic data section - when the value for the tariff changes the fields are reset;
    - >> The value for the field "Vorgabe" is "Leistungsvorgabe", which means that the field "Versicherungssumme" is enabled and the field "Beitrag" is read only.
  - d. Additional coverages data "Daten Zusatzversicherungen"
    - >> Coverages UZ\_201201 and "BZ\_201705" are displayed.
- 9. Click "Suchpartner" and select a partner as policy holder.
  - >> Name and Surname are copied from Innovas.
- 10. Uncheck the field "VN=VP1".
  - >> A new button "Suchpartner" is displayed; a new button "VN und VP1 tauschen" is displayed;
- 11. Click "Suchpartner" and select a partner as insured person.
  - >> Name, Surname, Date of Birth and Gender are copied to the screen; all these fields become read-only.
- 12. Click "VN und VP1 tauschen".
  - >> Data of the policy holder and insured person are exchanged.
- 13. Check the field "VN=VP1".
  - >> Data of the policy holder is copied to the insured person 1.
- 14. Fill in the remaining fields in the insured person section.
- 15. Fill in the value for the field "Versicherungssumme".
- 16. Change the value of the field "Vorgabe" to "Nettobeitragsvorgabe".
  - >> Value of the field "Versicherungssumme" is reset and the field becomes read-only; Value of the field "beitrag" becomes editable.
- 17. Fill in the remaining fields on the main tariff section.
- 18. Select the additional coverage "UZ\_201201" by clicking in the corresponding checkbox.
  - >> System copies the value of the fields "Versicherungssumme" and "Dauer" of the main tariff to the additional coverage.
- 19. Unselect the additional coverage "UZ\_201201" by clicking again in the corresponding checkbox.
  - >> System resets the fields of the additional coverage.
- 20. Click in one of the following options:
  - a. Cancel "Abbrechen".
  - >> Search offer screen is displayed.
  - b. OK.
  - >> The process moves to the second step of the offer wizard.
- 21. (Considering 19. b.) Summary screen is displayed.
  - >> In case of simple products, only one variant is displayed; in case of combined offers (S&S marketable offers) 2 or 3 columns are displayed, each one with data of one contract.
- 22. Validate the illustration identification.
  - >> Illustration identification consists on a prefix LT followed by a 12 digit sequence number (e.g.: LT2000000000001)
- 23. Check the following items in the top of the screen:
  - a. Click Edit "Ändern":
    - >> Move back to the first screen; the button appears only once, even for the scenario of a marketable offer with more than one contracts;
  - b. Click Copy:
    - >> Create a new opportunity based on the selected contract; the icon appears also below the table, for each variant;
  - c. Click Propose:
    - >> Move the selected contract to application; the icon appears also below the table, for each variant;
- 24. Check the following items in the bottom of the screen:
  - a. Click Cancel "Abbrechen"
  - >> Move to the search offer screen without saving changes;
  - b. Click Bundle "Bündel"
  - >> Move to the bundle screen and save the opportunity.
- 25. Check the following info in the summary table:
  - a. "Generic info", "Versicherte Person", "Basisdaten", "Beiträge gemäß Zahlungsweise" and "Zuschläge gemäß Zahlungsweise".
  - b. Some fields in bold style.
  - c. If there is more than one variant (more than one contract), the fields with different values are highlighted.
- 26. Repeat the process and create a new opportunity filling in all the data except importing insured person from Innovas (meaning, gender and date of birth should be manually selected - scenario of "anonymous quote");
  - >> When clicking OK the summary screen should be displayed; premium is calculated even without an explicit partner as insured person.

## 5. Bundle and compare screen

1. Perform a search in the search offer screen and access bundle screen by clicking "B" action item.  
 >> Bundle screen is displayed.
2. Validate the bundle identification.  
 >> The bundle identification consists of a prefix LB followed by a 12 digit sequence number (e.g.: LB200000000001)
3. Check the following columns in the list of with the opportunities:
  - a. First column with a checkbox (in case of single contracts) or an expand/collapse icon (in case of combined contracts - marketable offers);
  - b. Column "Aktionen" with the possible actions. following action items: copy data for a new opportunity, open opportunity, delete opportunity;
  - c. Insured person "Vollständiger N"
  - d. Policy holder "Vollständiger N"
  - e. Product "Produkt"
  - f. Tarif "Tarif"
  - g. Sum insured "Leistung. Erleben"
  - h. Begin date "Versicherungs..."
  - i. Contract status "Vertrags Status"
  - j. Creator "Erstellt von"
  - k. Update date "Aktualisierungs"
4. Check the actions items for each illustration/contract:
  - a. Click Copy data for a new opportunity.  
 >> Product selection screen is displayed with the indication of the base contract.
  - b. Click Open opportunity.  
 >> Summary screen is displayed.  
 >> In case it is an illustration, all the contracts are displayed; in case it is a single contract, only the selected contract is displayed
  - c. Click Delete opportunity.  
 >> This action item is only available at the level of the illustration; contracts of an illustration should
5. Select 1 contract (click on the checkbox);  
 >> Button compare "Auswahl vergleichen" continues disabled;
6. Select 2 contracts;  
 >> Button compare "Auswahl vergleichen" turns enabled;
7. Select 3 contracts;  
 >> Button compare "Auswahl vergleichen" continues enabled; remaining checkboxes are disabled;
8. Click compare "Auswahl vergleichen";  
 >> Compare screen is opened with the selected contracts; differences are highlighted;
9. Click button to add a new opportunity "Hinzufügen"  
 >> An opportunity creation process is started (product selection screen is displayed); the created opportunity will be added to the same bundle;
10. Click action item to delete an opportunity.  
 >> The opportunity (and all its contracts) is removed from the bundle (in fact it is removed from the database, which means that it should not be accessible even from the search offer screen).
11. Click option to expand and collapse an opportunity.  
 >> When expanding an opportunity, a header is displayed (in bold style; attributes that are common to all the contracts are displayed, otherwise the column remains empty) and a row is displayed for each contract of the opportunity;  
 >> When collapsing an opportunity, only a header is displayed (in bold style; attributes that are common to all the contracts are displayed, otherwise the column remains empty)

## 8. Office Integration

1. Access Office screen
2. Open a "BUNDLE\_NEW" operation defining a partner id and move to S&S  
 >> S&S Search partner screen is displayed with the field "Partner id" filled in and the search performed.  
 >> The field partner id is not editable.
3. Open a "BUNDLE\_NEW" operation without defining a partner id and move to S&S  
 >> S&S Search partner screen is displayed with empty search criteria.  
 >> All the fields are editable.

## ANEXO I: Plano de testes VHV versão P 0.2

### VHV Release Test 1.0.2 (P 0.2)

#### Introduction

- test primarily in Google Chrome
- but also on IE 10+ and Firefox
- test both in English and German browser settings

#### Environments

<http://le-s-po-w2012r2:20188/cid/Login.do> (CID QA Environment)

#### Log in

User:if4

Password:if4

#### Sales and Service

##### 1. Authentication & Authorization

1. Access "Insurance Suite" log in page.
2. Log in to "Insurance Suite" with valid credentials.  
>> S&S menu is displayed with all the options: "Offer", "Search partner", "My opportunities" "Search opportunities", "New opportunity", "My marketable offers" and "New marketable offer".

##### 2. Search Offer

1. Click in the option "Offer" in S&S.  
>> A new tab is opened with the corresponding screen.  
>> Information about the S&S deployed version is displayed in the bottom of the screen.
2. Check if the following search fields are available:
  - a. Opportunity ID "Angebots/Antrags-Nr"
  - b. Create Date "Erstelldatum"
  - c. Update Date "Letzte Änd."
  - d. Status
  - e. Partner-number "Partner-Nummer"
3. Click search "Suchen" to perform search:
  - b. Using Opportunity ID
  - c. Using Create Date
  - d. Using Update Date
  - e. Using Status
  - f. Using Partner-Number
  - g. Using some combinations ( e.g. Opportunity ID and Create Date; Status and Partner-Number)
4. Check if all contracts are listed according to the entered criteria(s). a) to g)  
>> In the search results the system displays contracts, i.e. if an opportunity has 3 contracts and all meet the search criteria, then 3 rows are displayed.
5. Check if the following columns are displayed in the list:
  - a. Actions "Aktionen"
    - i. Open "Angebot/Antrag beauskunften"
    - ii. Bündel
    - iii. Copy "Eine kopie dem Bündel Hinzufügen"
    - iv. Propose  
>> This action item will only be available for contracts in WIP or Printed status
- b. Insured person name "Vollständiger Name"

- c. Policy holder name "Vollständiger Name"
- d. Tarif
- e. Sum insured "Leistung. Erleben (abs.)"
- f. Effective date "Versicherungsbeginn"
- g. Opportunity status "Vertrags status"
- h. Update date "Aktualisierungsdatum"
- 6. Check actions for each contract.
  - a. Click open/edit quote icon
    - >> Contract should open on the summary page
  - b. Click Bündel icon
    - >> Bundle screen should be displayed
  - c. Click copy icon
    - >> Product selection screen is displayed allowing the user to create a new offer based on the data from the previous one
  - d. Click Propose icon
    - >> Associated contract moves to the application step; first screen of the application wizard is displayed;
- 7. Click New Opportunity button "Neues Angebot"
  - >> Product Selection Screen should be displayed

### 3. Search Partner (Innovas Integration)

- 1. Click in the option "Partnersuche" in S&S.
  - >> A new tab is opened with the corresponding screen.
  - >> Information about the S&S deployed version is displayed in the bottom of the screen.
- 2. Check if the following search fields are available:
  - a. Name "Zweitname"
  - b. Surname "Vorname"
  - c. Postal code "Postleitzahl"
  - d. Town "Stadt"
  - e. Street "Straße"
  - f. Date of birth "Geburtsdatum"
  - g. Partner id "Customer Id"
- 3. Click search "Suchen" to perform search:
  - b. Using Zweitname
  - c. Using Vorname
  - d. Using Postleitzahl
  - e. Using Stadt
  - f. Using Straße
  - g. Using Geburtsdatum
  - h. Using Customer Id
  - i. Using some combinations ( e.g. Zweitname and Vorname; Straße and Customer Id)
- 4. Check if all contracts are listed according to the criteria(s) entered, a) to g)
- 5. Check if the following columns are displayed in the list:
  - a. Actions "Aktionen"
    - i. Search quotes for partner
    - ii. Create opportunity for customer
  - b. Name "Zweitname"
  - c. Surname "Vorname"
  - d. Date of birth "Geburtsdatum"
  - e. Street "Straße"
  - f. Postal code "Postleitzahl"
  - g. Town "Stadt"
  - h. Partner id "Customer Id"
- 6. Check actions for each partner:
  - a. Click "Search quotes for partner" icon
    - >> Open search offer screen in the same tab and displays the list of contracts in which the partner exists as policy holder or Insured person
  - b. Click "Create opportunity for customer" icon
    - >> Open the product selection screen in the same tab allowing the user to create a new offer; partner will be pre-filled as policy holder;
- 7. Click new partner "Kunden anlegen" button
  - >> A new tab is opened with msg.Innovas

### 4. Offer screens

- 1. Access the search offer option.

2. Click new opportunity "Neues Angebot".
3. Select an effective date.  
 >> 8 Possible dates: first day of the previous month, current month and next 6 months.
4. Check the list of products displayed  
 >> Table including single products and marketable offers.
5. Select the product by clicking in the respective row.  
 >> The row is highlighted; All the products (HL\_T\_201810, HL\_ET\_201810, HL\_TP\_201810) and corresponding marketable offers are available;
6. Click in one of the following options
  - a. "OK"  
 >> Offer screen is displayed.
  - b. Cancel "Abbrechen"  
 >> Search offer screen is displayed.
7. (Considering 6. a.) Offer screen is presented, consisting on a wizard with 2 steps: "Basisdaten" and "Summary".
8. In the first screen the following sections are displayed:
  - a. Basic data "Daten Allgemein"  
 >> The effective date selected in the product selection screen is copied to the begin date of the opportunity and appears read only  
 >> The dropdown "Tarif" is not displayed if a marketable offer is selected.
  - b. Partner data "Versicherungsnehmer"  
 >> The checkbox VN=VP1 is selected by default;  
 >> The "gender" and "date of birth" of the insured are editable when there is no partner selected as insured person;  
 >> All the data provided by Innovas is read only;  
 >> The group of fields related with the Insured person 2 is displayed for the product HL\_TP\_201810 and corresponding marketable offers;
  - c. Main tariff data "Daten Haupttarif":  
 >> The fields displayed depend on the value of the field "Tarif" of the basic data section - when the value for the tariff changes the fields are reset;  
 >> The value for the field "Vorgabe" is "Leistungsvorgabe", which means that the field "Versicherungssumme" is enabled and the field "Beitrag" is read only.
  - d. Additional coverages data "Daten Zusatzversicherungen"  
 >> Coverages UZ\_201201 and "BZ\_201705" are displayed for the product HL\_T\_201810 and corresponding marketable offers;  
 >> Coverage UZ\_201201 is displayed for the product HL\_ET\_201810 and corresponding marketable offers;  
 >> Coverage UZ\_201201 is displayed for the product HL\_TP\_201810 and corresponding marketable offers;
9. Click "Suchpartner" and select a partner as policy holder.  
 >> Name and Surname are copied from Innovas.
10. Uncheck the field "VN=VP1".  
 >> A new button "Suchpartner" is displayed; a new button "VN und VP1 tauschen" is displayed;
11. Click "Suchpartner" and select a partner as insured person.  
 >> Name, Surname, Date of Birth and Gender are copied to the screen; all these fields become read-only.
12. Click "VN und VP1 tauschen".  
 >> Data of the policy holder and insured person are exchanged.
13. Check the field "VN=VP1".  
 >> Data of the policy holder is copied to the insured person 1.
14. Start typing the occupation "Beruf" of the insured.  
 >> Automatic search is performed.
15. Select an occupation.
16. Click in the button "Vorgelegen"  
 >> Occupation group is automatically updated based on the occupation or an error message is displayed;
17. Manually update the occupation group of the insured.
18. Fill in the height "Grobe" of the insured.
19. Fill in the weight "Gewicht" of the insured.  
 >> The system automatically invokes the BMI service and the result is displayed in the BMI field; if the value is out of bounds an error message is displayed and the BMI field remains empty.
20. Update the height "Grobe" of the insured.  
 >> The BMI service is invoked and the value is updated.
21. Fill in the remaining fields in the insured person section.
22. Repeat steps from 14 to 21 for the insured person 2 if the product is HL\_TP\_201810.
23. Fill in the value for the field "Versicherungssumme".
24. Change the value of the field "Vorgabe" to "Nettobeitragsvorgabe".  
 >> Value of the field "Versicherungssumme" is reset and the field becomes read-only; Value of the field "beitrag" becomes editable.
25. Check the values for the field "Tilgungsfreie Jahre" (T4 tariffs)  
 >> Possible values are 1, 2, 3, 4 or 5.
26. Fill in the remaining values on the main tariff section.
27. Select the additional coverage "UZ\_201201" by clicking in the corresponding checkbox.  
 >> System copies the value of the fields "Versicherungssumme" and "Dauer" of the main tariff to the additional coverage.
28. Unselect the additional coverage "UZ\_201201" by clicking again in the corresponding checkbox.  
 >> System resets the fields of the additional coverage.
29. Click in one of the following options:
  - a. Cancel "Abbrechen".  
 >> Search offer screen is displayed.
  - b. OK.  
 >> The process moves to the second step of the offer wizard.

30. (Considering 29. b.) Summary screen is displayed.  
 >> In case of simple products, only one variant is displayed; in case of combined offers (S&S marketable offers) 2 or 3 columns are displayed, each one with data of one contract.
31. Validate the illustration identification.  
 >> Illustration identification consists on a prefix LT followed by a 12 digit sequence number (e.g.: LT2000000000001)
32. Check the following items in the top of the screen:
  - a. Click Edit "Andem":  
 >> Move back to the first screen; the button appears only once, even for the scenario of a marketable offer with more than one contracts;
  - b. Click Copy:  
 >> Create a new opportunity based on the selected contract; the icon appears also below the table, for each variant;
  - c. Click Propose:  
 >> Move the selected contract to application; the icon appears also below the table, for each variant;
33. Check the following items in the bottom of the screen:
  - a. Click Cancel "Abbrechen"  
 >> Move to the search offer screen without saving changes;
  - b. Click Bundle "Bündel"  
 >> Move to the bundle screen and save the opportunity.
34. Check the following info in the summary table:
  - a. "Generic info", "Versicherte Person", "Versicherte Person 2" (if product HL\_TP\_201810 or variants), "Basisdaten", "Beiträge gemäß Zahlweise" and "Zuschläge gemäß Zahlweise".
  - b. Some fields in bold style.
  - c. If there is more than one variant (more than one contract), the fields with different values are highlighted.
35. Repeat the process to create a new opportunity filling in all the data except importing insured person 1 from Innovas (meaning, gender and date of birth should be manually selected - scenario of "anonymous quote");  
 >> When clicking OK the summary screen should be displayed.

## 5. Bundle and compare screen

1. Perform a search in the search offer screen and access bundle screen by clicking "B" action item.  
 >> Bundle screen is displayed.
2. Validate the bundle identification.  
 >> The bundle identification consists of a prefix LB followed by a 12 digit sequence number (e.g.: LB2000000000001)
3. Check the following columns in the list of with the opportunities:
  - a. First column with a checkbox (in case of single contracts) or an expand/collapse icon (in case of combined contracts - marketable offers);
  - b. Column "Aktionen" with the possible actions. following action items: copy data for a new opportunity, open opportunity, delete opportunity;
  - c. Insured person "Vollständiger N"
  - d. Policy holder "Vollständiger N"
  - e. Product "Produkt"
  - f. Tarif "Tarif"
  - g. Sum insured "Leistung. Erleben"
  - h. Begin date "Versicherungs..."
  - i. Contract status "Vertrags Status"
  - j. Creator "Erstellt von"
  - k. Update date "Aktualisierungs"
4. Check the actions items for each illustration/contract:
  - a. Click Copy data for a new opportunity.  
 >> Product selection screen is displayed with the indication of the base contract.
  - b. Click Open opportunity.  
 >> Summary screen is displayed.  
 >> In case it is an illustration, all the contracts are displayed; in case it is a single contract, only the selected contract is displayed
  - c. Click Create a new document request.  
 >> Document request generation screen is displayed.
  - d. Click Delete opportunity.  
 >> This action item is only available at the level of the illustration; contracts of an illustration should
5. Select 1 contract (click on the checkbox);  
 >> Button compare "Auswahl vergleichen" continues disabled;
6. Select 2 contracts;  
 >> Button compare "Auswahl vergleichen" turns enabled;
7. Select 3 contracts;  
 >> Button compare "Auswahl vergleichen" continues enabled; remaining checkboxes are disabled;
8. Click compare "Auswahl vergleichen";  
 >> Compare screen is opened with the selected contracts; differences are highlighted;
9. Click button to add a new opportunity "Hinzufügen"  
 >> An opportunity creation process is started (product selection screen is displayed); the created opportunity will be added to the same bundle;
10. Click action item to delete an opportunity.  
 >> The opportunity (and all its contracts) is removed from the bundle (in fact it is removed from the database, which means that it should

- not be accessible even from the search offer screen).
11. Click option to expand and collapse an opportunity.
    - >> When expanding an opportunity, a header is displayed (in bold style; attributes that are common to all the contracts are displayed, otherwise the column remains empty) and a row is displayed for each contract of the opportunity;
    - >> When collapsing an opportunity, only a header is displayed (in bold style; attributes that are common to all the contracts are displayed, otherwise the column remains empty)

## 6. Document requests

1. Access the bundle screen.
2. Check section "Documents" below the table of opportunities.
3. Check the columns of the document requests table:
  - a. Actions "Aktionen"
  - b. Date "Datum".
  - c. Type of document "Dokumenttyp".
  - d. Number of business object "Ang/Ant ID".
4. In the opportunities table, click in the action item to start a new document request.
  - >> A new screen is displayed allowing the user to fill in the data of the request:
  - a. Type of document and shipping method.
5. Check the buttons on the document request generation screen.
  - a. Cancel "Abbrechen"
  - >> The system displays the previous screen without any change.
  - b. OK
  - >> A new row is added to the document requests table.
6. (Considering 5. b.) Check the actions for the newly created document request:
  - a. Open the document request:
    - >> Document request definition screen is displayed in read only mode; only cancel "Abbrechen" button is displayed;
  - b. Delete the document request:
    - >> Document request is removed from the list;
  - c. Edit the document request:
    - >> Document request definition screen is displayed in edit mode; button cancel and OK are displayed;
7. Add an additional document request.
8. Click on "Dokument versenden"
  - >> Non submitted document requests are submitted to Life Factory;
  - >> In case of error a message is displayed and there is a rollback of the whole operation;
  - >> In case of success the documents requests become read-only (only possible action is open the document request) and the corresponding offer is updated to a "Printed" status;
9. Click on "Abbrechen"
  - >> All the opportunities in "WIP" status are marked as "Abandoned"
  - >> Search offer screen is displayed.

## 7. Application screens

1. Access the search offer screen.
2. Click on the action item to propose an offer.
  - >> Application screen is displayed.
3. Check the application screen.
  - >> The application screen consists on a wizard with 5 steps: "Partner", "Basisdaten", "Inkasso", "Summary" and "Dokumente".
  - >> It is possible to navigate in the wizard by clicking in the respective screen.
4. Check the header for each screen of the application wizard
  - >> The fields policy holder name, begin date, illustration id, insured person 1 name, insured person 1 date of birth and insured person 1 age are displayed.
5. Below the header the following action items are displayed:
  - a. Save application.
  - b. Perform calculation.
  - c. Move to the next screen.
    - >> This action is also available by clicking a button in the bottom of the screen.
  - d. Move to the previous screen.
    - >> This action is also available by clicking a button in the bottom of the screen.
6. Check the data on the first screen.
  - a. Information about the partners is displayed.
    - >> The layout is the same that is displayed in the offer screens, considering only the partner management.
7. Check the data on the second screen.
  - >> The layout is the same that is displayed in the offer screens, considering only the generic data, the main tariff data and the additional

- coverages data.
- >> It is possible to know that there are unsaved changes by the look of the save button;
- >> It is possible to navigate between the screens without losing unsaved data.
- 8. Check the data on the third screen.
  - a. A section "Bezugsrech" is displayed.
  - b. A section "Bankverbindung" is displayed.
  - >> User select a bank account by clicking in the respective row of the table; by default any bank account should be selected;
  - >> Listed bank accounts are associated with the policy holder and provided by Innovas;
  - c. A section "Identifizierung GWG" is displayed.
- 9. Check the data of the fourth screen.
  - >> Data in this screen is always read only.
  - a. A section "Versicherte Personen" is displayed.
  - >> A table with the insured person(s) is displayed including the columns role, name, first name, date of birth, age, raucher status, BMI index and occupation;
  - b. A section "Hauptversicherung" is displayed with data of the main coverage.
  - c. A section "Zusatztarif" is displayed with the data of the additional coverages.
  - >> This section only appears if there are additional coverages selected.
  - d. A section "Beiträge" is displayed with the list of coverages and, for each of them, the relevant data: name, begin date, end date, premium and sum insured.
- 10. Check the data of the fifth screen.
  - a. A section "Antragsdaten" is displayed.
  - b. A section "Dokumente" is displayed.
  - >> This section includes a list of documents requests associated with the specific contract; all the document requests are in read only mode (only action "open" is available).
- 11. Change data and perform "Save" operation.
- 12. Perform calculation.
- 13. Perform all the operations (Save, Calculate, Previous and Next) using the respective shortcuts:
  - a. Save: ALT + S
  - b. Calculate: ALT + B
  - c. Next: ALT + W
  - d. Previous: ALT + Z
- 14. Access the application fro the search offer screen and click open.
  - >> Summary screen should be displayed.

## 8. Office Integration

1. Access Office screen
2. Open a "BUNDLE\_NEW" operation defining a partner id and move to S&S
  - >> S&S Search partner screen is displayed with the field "Partner id" filled in and the search performed.
  - >> The field partner id is not editable.
3. Open a "BUNDLE\_NEW" operation without defining a partner id and move to S&S
  - >> S&S Search partner screen is displayed with empty search criteria.
  - >> All the fields are editable.
4. Open an "ORDER\_NEW" operation defining the offer to be proposed
  - >> S&S application first screen is displayed with the referenced opportunity.
5. Open an "ORDER\_NEW" operation without definig the offer to be proposed
  - >> S&S Search Offer screen should be displayed together with an error message.
6. Access the bundle screen with an Office task and click cancel "Abbrechen".
  - >> Office task is updated with a specific reason.
7. Access the bundle screen with an Office task and click cancel "Dokumente versenden".
  - >> Office task is updated with a specific reason.

## 9. CID Integration

1. Check different types of message:
  - a. Error messages
  - b. Info messages
  - c. Warning messages
  - d. Confirmation messages
- >> Messages appear on top of the screen, each type with a specific layout.