

**INSTITUTO UNIVERSITÁRIO MILITAR**  
**DEPARTAMENTO DE ESTUDOS PÓS-GRADUADOS**  
**CURSO DE PROMOÇÃO A OFICIAL SUPERIOR**  
**2021/2022**



**TII**

**AUTOMATIZAÇÃO DE TESTES NO DESENVOLVIMENTO E  
MANUTENÇÃO DE SISTEMAS DE INFORMAÇÃO NA FORÇA AÉREA**

**O TEXTO CORRESPONDE A TRABALHO FEITO DURANTE A  
FREQUÊNCIA DO CURSO NO IUM SENDO DA RESPONSABILIDADE DO  
SEU AUTOR, NÃO CONSTITUINDO ASSIM DOCTRINA OFICIAL DAS  
FORÇAS ARMADAS PORTUGUESAS OU DA GUARDA NACIONAL  
REPUBLICANA.**

**Licínio Jorge Godinho Simões**  
**CAP/TINF**



**INSTITUTO UNIVERSITÁRIO MILITAR  
DEPARTAMENTO DE ESTUDOS PÓS-GRADUADOS**

**AUTOMATIZAÇÃO DE TESTES NO  
DESENVOLVIMENTO E MANUTENÇÃO DE SISTEMAS  
DE INFORMAÇÃO NA FORÇA AÉREA**

**CAP/TINF Licínio Jorge Godinho Simões**

Trabalho de Investigação Individual do CPOS-FA 2021/2022 2ª Ed.

Pedrouços 2022



**INSTITUTO UNIVERSITÁRIO MILITAR  
DEPARTAMENTO DE ESTUDOS PÓS-GRADUADOS**

**AUTOMATIZAÇÃO DE TESTES NO  
DESENVOLVIMENTO E MANUTENÇÃO DE SISTEMAS  
DE INFORMAÇÃO NA FORÇA AÉREA**

**CAP/TINF Licínio Jorge Godinho Simões**

Trabalho de Investigação Individual do CPOS-FA 2021/2022 2ª Ed.

Orientador: TCOR/TINF Mário Miguel Reis Pereira

Pedrouços 2022



### **Declaração de compromisso Antiplágio**

Eu, **Licínio Jorge Godinho Simões**, declaro por minha honra que o documento intitulado **Automatização de testes no desenvolvimento e manutenção de Sistemas de Informação na Força Aérea** corresponde ao resultado da investigação por mim desenvolvida, enquanto auditor do **Curso de Promoção a Oficial Superior - Força Aérea 2021/2022, 2ª Edição** no Instituto Universitário Militar, e que é um trabalho original, em que todos os contributos estão corretamente identificados em citações e nas respetivas referências bibliográficas.

Tenho consciência que a utilização de elementos alheios não identificados constitui grave falta ética, moral, legal e disciplinar.

Pedrouços, **19 de julho de 2022**

Licínio Jorge Godinho Simões  
CAP/TINF



## **Agradecimentos**

A realização deste Trabalho de Investigação Individual não seria possível sem a ajuda e apoio de várias pessoas e é chegado o momento de exprimir os meus sinceros agradecimentos.

Agradeço acima de tudo e em primeiro lugar à minha família, pelo apoio, encorajamento e compreensão durante os momentos mais difíceis.

Agradeço ao meu orientador, Tenente-Coronel Mário Pereira, pela orientação, paciência, conselhos, dedicação e também pelo tempo despendido.

Agradeço a todos os entrevistados, militares e civis, cujo contributo foi fundamental, pela colaboração e por disponibilizarem o seu tempo e sabedoria para responderem às entrevistas de forma a enriquecer este trabalho.

Agradeço ao Diretor de Curso, Coronel Diamantino Fernandes pela dedicação, empenho e apoio demonstrado.

Agradeço aos Docentes pela partilha de conhecimento e pela disponibilidade demonstrada durante o curso.

Agradeço a todos os camaradas auditores do Curso de Promoção a Oficial Superior 2021/2022, 2ª Edição, pela partilha de conhecimento e de experiências, pela entreaajuda e companheirismo durante todos os momentos.

Agradeço ainda ao meu amigo Carlos pelo apoio, amizade e ajuda prestada durante o curso.

A todos, um muito obrigado.



## Índice

|                                                                                                                                                    |         |
|----------------------------------------------------------------------------------------------------------------------------------------------------|---------|
| 1. Introdução .....                                                                                                                                | 8       |
| 2. Enquadramento teórico e revisão de literatura .....                                                                                             | 11      |
| 2.1 Sistemas de Informação .....                                                                                                                   | 11      |
| 2.2 Automatização de Testes de SI .....                                                                                                            | 12      |
| 2.3 Modelo de Análise .....                                                                                                                        | 12      |
| 3. Metodologia .....                                                                                                                               | 13      |
| 4. Vantagens e desvantagens na adoção de um modelo de automatização de testes face aos processos atuais de desenvolvimento de SI na FA .....       | 15      |
| 4.1 Implementação .....                                                                                                                            | 15      |
| 4.2 Testes .....                                                                                                                                   | 17      |
| 4.3 Recursos Humanos .....                                                                                                                         | 18      |
| 4.4 Síntese conclusiva e resposta à PD1 .....                                                                                                      | 19      |
| 5. Boas práticas de outras organizações no que toca à utilização de modelos de automatização de testes no desenvolvimento e manutenção de SI ..... | 21      |
| 5.1 <i>Software</i> .....                                                                                                                          | 21      |
| 5.2 Gestão .....                                                                                                                                   | 22      |
| 5.3 Riscos .....                                                                                                                                   | 23      |
| 5.4 Síntese conclusiva e resposta à PD2 .....                                                                                                      | 24      |
| 6. Aplicabilidade de um modelo de automatização de testes no desenvolvimento e manutenção de SI na FA .....                                        | 26      |
| 7. Conclusões .....                                                                                                                                | 27      |
| Referências bibliográficas .....                                                                                                                   | 30      |
| <br>Apêndice A – Modelo de análise .....                                                                                                           | Apd A-1 |
| Apêndice B – Guião de entrevistas semiestruturadas .....                                                                                           | Apd B-1 |



## Resumo

A Força Aérea desenvolve e sustenta sistemas de informação necessários à sua atividade em áreas exclusivas e sensíveis, sendo alguns de elevado grau de complexidade, que necessitam de elevada qualidade e alta fiabilidade.

Tendo em conta a dificuldade de atrair e reter recursos humanos qualificados na área das tecnologias de informação, a garantia de qualidade dos sistemas atuais, o aumento dos pedidos de novos sistemas e a complexidade de algum do *software* já existente, torna-se cada vez mais importante a otimização de todo o processo de desenvolvimento de *software*.

Metodologicamente recorrendo a um raciocínio indutivo, assente numa estratégia qualitativa e num desenho de pesquisa de estudo de caso, recorreu-se a análise documental e entrevistas semiestruturadas, com o intuito de analisar se a adoção de um modelo de automatização de testes cria mais-valias durante o processo de desenvolvimento e manutenção de sistemas de informação na Força Aérea.

Através de exemplos em organizações civis que produzem sistemas de informação, bem como de análise documental, concluiu-se que poderá acrescentar valor ao processo de desenvolvimento de *software* na Força Aérea, aumentando a qualidade dos sistemas desenvolvidos e permitindo uma melhor gestão dos recursos humanos, utilizando-os em tarefas criativas.

**Palavras-chave:** Automatização de Testes, Testes de *Software*, Desenvolvimento de Sistemas de Informação, Força Aérea.



## **Abstract**

*The Air Force develops and maintains information systems required for its activity in exclusive and sensitive areas, some of which being highly complex, requiring a high level of quality and reliability.*

*Considering the difficulty to attract and retain qualified human resources in the areas of information technology, the quality assurance of current systems, the increase in requests for new systems and the complexity of some of the existing software, it becomes increasingly important to optimize the entire software development process.*

*Methodologically, using inductive reasoning, based on a qualitative strategy and a case study research design, supported by documental analysis and semi-structured interviews, to analyze whether the implementation of a test automation model creates added value during the process of developing and maintaining information systems in the Air Force.*

*Through examples in civil organizations that produce information systems, as well as document analysis, it was concluded that it can add value to the software development process in the Air Force, increasing the quality of the developed systems and allowing for a better management of human resources, using these in creative tasks.*

**Keywords:** *Test Automation, Software Testing, Information Systems Development, Air Force.*



### **Lista de abreviaturas, siglas e acrónimos**

|         |                                                             |
|---------|-------------------------------------------------------------|
| ADIES   | Administradores da Informação da Estrutura Superior         |
| ADIAF   | Administradores da Informação das Áreas Funcionais          |
| ADIAFIN | Administrador da Informação da Área Financeira              |
| ADIAL   | Administrador da Informação da Área de Logística            |
| ADIAO   | Administrador da Informação da Área Operacional             |
| ADIAP   | Administrador da Informação da Área de Pessoal              |
| CPOS-FA | Curso de Promoção a Oficial Superior da Força Aérea         |
| DCSI    | Direção de Comunicações e Sistemas de Informação            |
| EPR     | Entidade Primariamente Responsável                          |
| FA      | Força Aérea                                                 |
| GCGI    | Grupo Coordenador de Gestão de Informação                   |
| IUM     | Instituto Universitário Militar                             |
| OE      | Objetivo Específico                                         |
| OG      | Objetivo Geral                                              |
| PD      | Pergunta Derivada                                           |
| PDSIFA  | Plano Diretor dos Sistemas de Informação da Força Aérea     |
| PI      | Projeto de Investigação                                     |
| PP      | Pergunta de Partida                                         |
| RSI     | Repartição de Sistemas de Informação                        |
| SASI    | Sub-Repartição de Administração de Sistemas de Informação   |
| SDSI    | Sub-Repartição de Desenvolvimento de Sistemas de Informação |
| SI      | Sistemas de Informação                                      |
| TI      | Tecnologias de Informação                                   |
| TII     | Trabalho de Investigação Individual                         |



## 1. Introdução

Segundo o MCLAFA 305-2, a Direção de Comunicações e Sistemas de Informação (DCSI) é a direção responsável pelo desenvolvimento, manutenção e evolução de Sistemas de Informação (SI) na Força Aérea (FA), sendo esta função executada pela Repartição de Sistemas de Informação (RSI) (MCLAFA 305-2, 2013, p. 179).

Tal como indicado no PDSIFA, (2015, p. 31), “A Força Aérea tem necessidade de desenvolver aplicações ajustadas à sua atividade em áreas exclusivas e sensíveis”, sendo alguns SI de elevado grau de complexidade. Ainda de acordo com o PDSIFA (2015, p. 12), alguns destes sistemas são mesmo considerados críticos, dos quais se destacam os sistemas de gestão operacional e manutenção de aeronaves.

Na FA, a DCSI desenvolve sistemas de informação nas mais diversas áreas, passando pelas áreas operacional, logística, pessoal, financeira e ensino garantindo a sua manutenção evolutiva, de acordo com o que está descrito no PDSIFA (2015, p. 65), “as necessidades de sistemas de informação, ou as grandes alterações aos mesmos, podem surgir por três vias distintas; através do ciclo de planeamento refletido nos Planos Diretores dos Sistemas de Informação e das Tecnologias de Informação e Comunicação, por necessidades pontuais identificadas pelos Administradores da Informação das Áreas Funcionais (ADIAF) ou pelo Grupo Coordenador de Gestão de Informação (GCGI)”.

Conforme indicado pelo Capitão C. Pombeiro, chefe da Sub-Repartição de Desenvolvimento de Sistemas de Informação (SDSI) (entrevista por email, 4 de junho de 2022), a metodologia utilizada pela RSI, repartição com a função de desenvolvimento e manutenção dos SI da FAP, é a Agile. Esta permite um desenvolvimento iterativo e uma maior colaboração com a entidade que efetua o pedido de SI, permitindo disponibilizar os produtos de forma mais rápida e mais flexível.

Segundo o Capitão C. Pombeiro, os testes aplicacionais são efetuados manualmente durante a fase de desenvolvimento pelos programadores e após a passagem para o ambiente de qualidade, são os ADIAF que efetuam os testes funcionais que se considerem suficientes, não existindo durante o ciclo de vida dos SI, nenhuma fase de automatização de testes.

Ainda de acordo com o Capitão C. Pombeiro, atualmente, a SDSI tem mais de 50 SI e apesar de todos contribuírem para a missão da FA de uma forma ou de outra, 15 deles apresentam um grau elevado de complexidade, constituindo-se, por isso, e pela



importância que representam no contributo para a missão da FA, prioritários relativamente aos demais. No entanto, apenas possui 13 programadores para a sua manutenção bem como para o desenvolvimento de novos SI, sendo que o seu nível de experiência é bastante heterogéneo, o que restringe em muito o universo de elementos que podem ser alocados aos SI mais complexos, fruto do seu nível de autonomia e de conhecimento destes SI. Esta crescente necessidade de novos SI ou de evoluções a SI já desenvolvidos, conjugada com a falta de recursos humanos com experiência, poderá levantar problemas de operacionalidade do ramo com custos financeiros e organizacionais extraordinariamente elevados.

A presente investigação tem por objeto a automatização de testes no desenvolvimento e manutenção de SI na FA, sendo delimitado pelos domínios espacial, temporal e de conteúdo (Santos & Lima, 2019). Sendo assim, o domínio espacial será a FA, o domínio temporal será o contexto atual e o domínio de conteúdo será a Automatização de testes de SI.

As conclusões deste estudo, a serem válidas, poderão constituir uma proposta para a sua implementação na FA, contribuindo para uma melhor racionalização dos meios e num melhor produto operacional.

Apresenta-se como Objetivo Geral (OG) “analisar a aplicabilidade de um modelo de automatização de testes no desenvolvimento e manutenção de SI na FA”, para o qual concorrem os Objetivos Específicos (OE):

**OE1:** Analisar as vantagens e desvantagens de um modelo de automatização de testes face aos processos atuais de desenvolvimento de SI na FA.

**OE2:** Analisar as práticas de outras organizações na utilização de modelos de automatização de testes no desenvolvimento e manutenção de SI.

Tendo em conta estes objetivos, definiu-se a seguinte Pergunta de Partida (PP): “Será que a introdução de automatização de testes no desenvolvimento e manutenção de SI na FA cria valor?” e as duas Perguntas Derivadas (PD):

**PD1:** Quais as vantagens e desvantagens na adoção de um modelo de automatização de testes face aos processos atuais de desenvolvimento de SI na FA?

**PD2:** Quais as boas práticas de outras organizações no que toca à utilização de modelos de automatização de testes no desenvolvimento e manutenção de SI?



Estruturalmente, este TII encontra-se organizado em sete capítulos. No primeiro capítulo é efetuada a introdução. No segundo capítulo encontra-se o enquadramento teórico e revisão de literatura. O terceiro capítulo diz respeito à metodologia utilizada durante o desenvolvimento da investigação. No quarto capítulo, são apresentadas vantagens e desvantagens da adoção de testes automáticos, bem como a resposta à PD1. No quinto capítulo, são apresentadas as práticas de duas organizações que utilizam modelos de automatização de testes, bem como a resposta à PD2. No sexto capítulo temos a resposta á PP. No sétimo capítulo, são apresentadas as conclusões, as limitações encontradas, bem como algumas recomendações para implementação.



## **2. Enquadramento teórico e revisão de literatura**

O tema deste trabalho de investigação insere-se no âmbito das Ciências Militares, na área das Ciências da Engenharia, incidindo na subárea de Engenharia de Sistemas de Informação. No presente capítulo será realizada uma contextualização do tema, integrando os conceitos estruturantes e o modelo de análise adotado.

### **2.1 Sistemas de Informação**

Os SI são ferramentas de grande importância em todas as organizações dado o seu impacto na sustentabilidade, evolução e transformação estrutural, o que leva à simplificação e otimização dos processos operacionais ou de gestão (Ribeiro, 2016).

Os SI são cada vez mais uma realidade em todas as organizações, fruto da flexibilidade e dinâmica que introduzem nos processos organizacionais. Estes SI, quando definidos corretamente e bem implementados, trazem ganhos de produção bastante relevantes, no entanto, a constante evolução destes, cria desafios enormes na sua sustentação. Com a crescente dependência destes SI, é de enorme relevância a viabilidade, a garantia de qualidade e taxa de disponibilidade (Ribeiro, 2016).

De forma a garantir o correto funcionamento de um SI, é necessária a execução de um conjunto de testes de forma a validar os processos implementados. Estes são passíveis de ser executados manualmente por um indivíduo ou uma equipa, no entanto é demasiado dispendioso e contraproducente, tendo em conta as limitações humanas em executar um elevado número de tarefas repetitivas de forma fiável (Carvalho, 2010, p. 34). Estes testes podem, contudo, ser efetuados de forma automática, garantindo a execução de um conjunto de testes pré-definidos sempre que uma nova funcionalidade é implementada, ou mesmo regularmente, ao longo do ciclo de vida de um SI (Pittet, s.d.).

Segundo Amaral (2021) “A automatização de testes é uma das principais prioridades das organizações na atualidade. Todas, ou quase todas, estão a apostar nessa vertente com o objetivo de reutilizar o esforço de realização de testes manuais, evitando a rotatividade de pessoas nos projetos, motivando os colaboradores e economizando tempo na comercialização de produtos e serviços”.

É de salientar que antes de qualquer automatização de testes, existem alguns fatores a serem considerados. Primeiramente, é necessário identificar quais os processos a analisar, as ferramentas que melhor respondem às necessidades e qual o nível de *Continuous*



*Testing*<sup>1</sup> que se pretende implementar, sendo que todos estes fatores contribuem para a maior ou menor rapidez com que se consegue implementar um determinado nível de automatização (Amaral, 2021). Não menos importante será o nível de equilíbrio entre o custo de automatização e os custos de desenvolvimento, sustentação e tarefas manuais (Amaral, 2021).

## **2.2 Automatização de Testes de SI**

Este conceito surge com a necessidade da garantia de qualidade de um *software* logo desde a sua definição inicial e com a redução de tempo despendido na fase de testes antes de distribuição de uma nova versão de *software* (Axelrod, 2018, p. 3). O seu objetivo principal é identificar constantemente falhas, otimizando o tempo necessário à execução destes testes e reduzindo a probabilidade de problemas que possam causar prejuízos nos mais diversos níveis (Silva, 2019, p. 5).

Consoante o nível de automatização que se pretenda, podem ser definidos testes de unidade ou unitários, funcionais, desempenho ou carga, ponta-a-ponta e de aceitação, que contribuem, como a componente chave, para se conseguir ter um processo de qualidade e que garanta escalabilidade destes SI (Pittet, s.d.):

## **2.3 Modelo de Análise**

Este trabalho de investigação foi enquadrado com os objetivos e sustentado nos conceitos estruturantes sendo definidas as respetivas dimensões e indicadores, que suportam o modelo de análise elaborado no Apêndice A.

---

<sup>1</sup> Fase de execução de testes automatizados como parte do processo de entrega de *software*, de forma a obter *feedback* sobre os riscos relativos ao lançamento de uma versão de *software*.



### 3. Metodologia

Esta investigação, foi desenvolvida utilizando um raciocínio indutivo, tendo “como ponto de partida a observação de factos particulares para, através da sua associação, estabelecer generalizações” (Santos & Lima, 2019, p. 18).

O desenho de pesquisa a ser utilizado enquadra-se num estudo de caso, pois “o estudo de caso enquadra-se no âmbito das estratégias de investigação qualitativas e apresenta uma natureza essencialmente empírica e descritiva” (Santos & Lima, 2019, p. 37).

Para materializar os objetivos deste TII, foram selecionados os participantes e o procedimento, os instrumentos de recolha e as técnicas de análise de dados, que permitem atingir o conhecimento adequado do objeto de estudo (Santos & Lima, 2019).

A investigação desenvolveu-se com a colaboração dos seguintes elementos:

- Coronel Luís Cordeiro (ADIAO);
- Major Paulo Sousa (ADIAFIN);
- Capitão Marta Matos (Gabinete do ADIAL);
- Capitão Maria Côrte-Real, Chefe da SASI;
- Capitão Carlos Pombeiro, Chefe da SDSI;
- Engenheiro Cristopher Pereira - *Quality Automation Engineer* da empresa TalkDesk<sup>2</sup>;
- Engenheiro Ricardo Costa - *Lead Architect* nos *Professional Services* da empresa OutSystems<sup>3</sup>.

Segundo Santos e Lima (2019), a etapa fundamental para uma correta ligação entre a teoria e os factos é uma adequada seleção dos instrumentos de recolha de dados, de forma a transitar de um determinado nível de conhecimento para um nível superior. Assim, foram utilizados os seguintes instrumentos:

- Análise documental de regulamentos da FA, livros, documentos científicos e artigos *online* sobre a automatização de testes de SI de forma a entender o processo.
- Entrevistas semiestruturadas realizadas a entidades intervenientes no processo de desenvolvimento de SI na FA e a dois elementos pertencentes a organizações que já possuem um elevado conhecimento e experiência.

---

<sup>2</sup> Empresa portuguesa de desenvolvimento de *software* na *cloud* para *contact centers*

<sup>3</sup> Empresa portuguesa líder do mercado de plataformas para o desenvolvimento rápido de aplicações



“A estratégia de investigação qualitativa por força da sua abrangência permite que o investigador analise entrevistas, discursos e até o próprio texto literário, tudo é suscetível de ser analisado” (Santos & Lima, 2019).

Durante a realização deste trabalho, as entrevistas semiestruturadas foram analisadas de forma qualitativa usando a análise de conteúdo e recorreu-se à análise documental de forma a complementar os dados contribuindo para avaliação e discussão dos resultados, apresentação de conclusões e implicações, contributos para o conhecimento, limitações e recomendações. (Santos & Lima, 2019).



#### **4. Vantagens e desvantagens na adoção de um modelo de automatização de testes face aos processos atuais de desenvolvimento de SI na FA**

##### **4.1 Implementação**

O processo de criação de um SI pressupõe antes de mais a adoção duma linguagem de desenvolvimento. O Capitão C. Pombeiro enunciou, como linguagens presentes na RSI, COBOL e Visual Basic 6, utilizadas nas aplicações *legacy* de primeira e segunda geração respetivamente. Para o desenvolvimento WEB é utilizado PHP, HTML e JavaScript. Para os novos SI, foi adotada a ferramenta *Service Studio* da empresa OutSystems que gera código .NET. De uma forma transversal a todas as aplicações é utilizada a linguagem SQL por forma a interagir com a base de dados.

Um SI é construído para ir ao encontro duma necessidade concreta. Essa mesma necessidade é levantada pelo AdIAF respetivo, sob a forma dum conjunto estruturado de requisitos. Estes constituem-se como a matéria-prima fundamental para a génese do SI, e deverão ser os mais claros e específicos possíveis. Nesta fase de levantamento de requisitos poderão ser já identificados um conjunto de testes de aceitação no sistema, no entanto, como atesta o Capitão C. Pombeiro, estes não são levantados pela DCSI, nesta, ou em qualquer outra fase do ciclo de desenvolvimento.

A implementação de um SI preconiza a existência dum ciclo de desenvolvimento de *software* bem definido. Tal como referido pelo Capitão C. Pombeiro, a RSI utiliza *Agile* como metodologia de desenvolvimento, assentando numa construção iterativa e bem delineada. Um ciclo de desenvolvimento constitui-se assim como um conjunto destes miniciclos denominados *sprints*. Cada um destes tem a duração de três a quatro semanas, tal como enunciado pela Capitã M. Côrte-Real (entrevista por email, 4 de junho de 2022). No final do *sprint* é apresentado à Entidade Primariamente Responsável (EPR), o resultado deste desenvolvimento, tendo posteriormente acesso ao mesmo em ambiente de qualidade por forma a poder iniciar a fase de testes de aceitação, tal como testemunhado pelo Capitão C. Pombeiro.

Durante o processo de desenvolvimento, os programadores efetuam unicamente testes básicos. A exceção a este fenómeno, assenta nas aplicações construídas em PHP e Visual Basic 6, cujo código fonte está alojado numa ferramenta *open source*, denominada *GitLab*, e que permite a criação de processos automatizados de testagem e distribuição de código. Tal como referido pela Capitã M. Côrte-Real esta ferramenta permite a realização



de testes elementares de análise estática do código. A DCSI, como atesta o Capitão C. Pombeiro, não dispõe de nenhuma ferramenta que permita a automatização de testes.

No atual modelo de desenvolvimento da FA a maioria dos testes é efetuada depois da passagem para o ambiente de qualidade, tal como indicado pelo Major P. Sousa (entrevista por email, 4 de junho de 2022). Os atores responsáveis pela realização dos mesmos poderão estar colocados nos gabinetes dos ADIAF, tal como testemunhado pela Capitão M. Matos (entrevista por email, 4 de junho de 2022) e pelo Coronel L. Cordeiro (entrevista por email, 4 de junho de 2022) ou fora dos mesmos tal como referido pelo Major P. Sousa, sendo neste último caso identificados utilizadores chave, com clara afinidade com o sistema.

No paradigma apresentado e em vigor na FA, não existe ainda a possibilidade de testes automáticos, sendo responsabilidade de vários atores a realização dos mesmos. A automatização, tal como indicado pelo Capitão C. Pombeiro, permitiria identificar precocemente a existência de erros aplicativos no *sprint* atual, garantindo ainda que este não influenciaria negativamente *sprints* passados e em produção. A Capitão M. Côrte-Real realça ainda que o tempo de estágio em ambiente de qualidade diminuiria, assim como o impacto de uma incorreta testagem especialmente em SI complexos, aumentando assim a fiabilidade do sistema e a rapidez de desenvolvimento.

Tendo em consideração a escassez de recursos humanos e o aumento das tarefas existentes por parte dos mesmos, a automatização de testes constituir-se-ia como uma vantagem, tal como referido pela Capitão M. Matos. Segundo o Coronel L. Cordeiro, os testes não poderão ser exclusivamente automáticos pois poderão conflitar com a disponibilidade e afetação do propósito do próprio sistema de informação, podendo ser, no entanto, úteis para a deteção de erros grosseiros.

Atualmente, a DCSI não dispõe de nenhuma *checklist*, para a realização de testes. Segundo a opinião do Capitão C. Pombeiro a criação duma *checklist* em mecanismo de execução automático de testes aumentaria substancialmente a qualidade do *software* produzido. Segundo o testemunho do Major P. Sousa, no seu gabinete não existe uma *checklist* de testagem, sendo pedido a quem realiza os mesmos, um conjunto de testes dirigidos, mas não limitados. Para o mesmo,



[...] a realização de testes de forma totalmente autónoma acarreta o risco de alguma funcionalidade não vir a ser testada, pelo que se considera pertinente definir o mínimo a testar, mas deixar em aberto outros testes que entendam efetuar (*op. cit.*).

No gabinete do ADIAL, existem *checklists* para testagem e que são percorridas antes de qualquer entrada em produtivo. Para a Capitão M. Matos, uma automatização destas acarretaria diversas vantagens,

[...] tarefas repetitivas existentes nesses mesmos testes e com o facto de podermos alocar recursos humanos noutras tarefas, tão pertinentes como os testes, ou abreviar esses mesmos testes. Antes da entrada em produtivo de uma nova versão é sempre efetuado o conjunto de testes da *checklist* para verificar, para além do bom funcionamento das novas alterações, verificar também se nenhuma das funcionalidades anteriores foi afetada. (*op. cit.*).

#### 4.2 Testes

Como observado na dimensão anterior a FA não dispõe até ao momento de ferramentas, nem incorpora processos para a automatização de testes de *software*. Para auxiliar nesta dimensão obtiveram-se os testemunhos dos engenheiros Ricardo Costa e Christopher Pereira, colaboradores de empresas onde já existe uma cultura enraizada para a integração de testes automáticos na produção de *software*.

Um dos principais pressupostos na utilização de testes automáticos consiste em definir concretamente o que deve ser automatizado ou não, tal como refere o Engenheiro C. Pereira (entrevista por email, 4 de junho de 2022), sendo esta escolha um processo moroso, mas necessário, uma vez que uma *framework* de automação exige quase tanta manutenção quanto o respetivo SI que a mesma testa.

Tendo em conta os vários tipos de teste identificados superiormente, procurou-se perceber se os mesmos são utilizados nestas duas organizações, obtendo-se as respostas e testemunhos seguintes:

- Testes de Unidade – estes são adotados por ambas as organizações, tendo o Engenheiro C. Pereira referido que os mesmos são executados pelos programadores no ambiente local e no repositório de código.
- Testes Funcionais – também em ambas as empresas, sendo utilizados como indica o Engenheiro C. Pereira durante a execução de testes a uma nova

funcionalidade, na disponibilização para os vários ambientes e na execução de rotinas diárias.

- Testes de Desempenho – estes são utilizados na empresa OutSystems, e são tipicamente apelidados de testes de carga, tal como referido pelo engenheiro R. Costa (entrevista por email, 4 de junho de 2022), sendo empregues no ambiente de pré-produção, uma vez que este é o ambiente que mais se assemelha ao de produção. Este tipo específico de testes é aplicado maioritariamente antes da passagem duma versão de *software* entre ambientes e pode justificar uma não passagem, se o desempenho observado não for aceitável.
- Testes de Ponta-a-Ponta – este tipo é utilizado apenas na TalkDesk, nos mesmos moldes dos testes funcionais automáticos, tal como atesta o Engenheiro C. Pereira.

Como observado, existem vários tipos diferentes de testes automáticos, cabendo à organização definir qual ou quais os mais adequados à sua realidade, e em situações optar por automatização ao invés de testes manuais. Segundo o Engenheiro R. Costa, na OutSystems, essa decisão é tomada internamente quando se prevê um impacto elevado nas novas funcionalidades desenvolvidas. Por sua vez o Engenheiro C. Pereira preconiza a automatização de testes, quando é necessário testar no mínimo a integração de dois micro serviços.

### **4.3 Recursos Humanos**

A inexistência de recursos humanos com experiência neste tema levou a que se procurasse obter o conhecimento necessário junto das empresas identificadas previamente. As mesmas possuem recursos humanos e uma organização adequada que permitiu a adoção de automatização de testes. Tal como testemunhou o Engenheiro C. Pereira a equipa de Qualidade é a responsável pela definição dos testes a automatizar, podendo articular esforços conjuntamente com a equipa de desenvolvimento.

A criação de uma *framework* de testes automáticos, segundo o Engenheiro C. Pereira, exige uma constante manutenção e sustentação, estando a complexidade deste processo intimamente relacionada com a experiência dos elementos que constituem esta equipa. É aconselhável, segundo o mesmo, existir pelo menos um programador experiente na equipa. O tempo de maturação dum programador de modo a atingir uma proficiência elevada está, como referido pelo Engenheiro C. Pereira, intrinsecamente ligado à sua



experiência em desenvolvimento de *software*. Nesta matéria o Engenheiro R. Costa atesta a sua dificuldade em quantificar o tempo necessário para a proficiência elevada, dependendo sempre dos projetos participados pelo elemento.

A formação, não sendo obrigatória, é fortemente aconselhada pelos dois elementos entrevistados principalmente numa fase inicial da implementação do modelo. Segundo as palavras do Engenheiro C. Pereira,

[...] formação será sempre importante, pois permite-nos estar mais familiarizados com as boas práticas da indústria. Existe igualmente formação específica para a área de automação de testes de *software*, que novamente, é importante pois permite dar a conhecer quais as boas práticas da indústria. Mas, quanto a esta última, ela apenas será útil para uma empresa/entidade que não tenha rigorosamente nada automatizado nem ninguém com a experiência em criar uma *framework* de automação. (*op. cit.*).

Tão importante como a formação e a experiência dos recursos humanos numa organização, está a capacidade da mesma em enraizar uma cultura orientada para a qualidade e para a manutenção e controlo da mesma. Os testemunhos abaixo do Engenheiro C. Pereira atestam nesse mesmo sentido,

[...] Acima de tudo é necessário que exista uma cultura para a Qualidade na entidade/empresa na qual está a ser desenvolvido o produto. É fundamental entender que incluir o processo de garantia de qualidade, do qual fazem parte os testes, é o que permitirá reduzir custos com a manutenção de um sistema, antecipar e evitar a entrada de problemas no ambiente de produtivo (*op. cit.*).

#### **4.4 Síntese conclusiva e resposta à PD1**

Em resposta à PD1, “Quais as vantagens e desvantagens na adoção de um modelo de automatização de testes face aos processos atuais de desenvolvimento de SI na FA?” enumeram-se as vantagens identificadas:

- Identificação precoce de erros aplicacionais;
- Minimização do impacto de novos desenvolvimentos nas funcionalidades do sistema existentes em ambiente de produção;
- Menor tempo em ambiente de qualidade culminando numa maior rapidez de desenvolvimento;
- Menor o risco no SI, fruto duma incorreta testagem. O não acautelar desta situação aumenta com a complexidade do SI;



- Menor tempo despendido pelo utilizador na testagem. Este fator é especialmente importante tendo em conta a escassez de recursos humanos e o avolumar de tarefas nos mesmos;
- Maior fiabilidade do *software*;
- Aumento na qualidade do *software* produzido.

Por sua vez, foram também identificadas um conjunto de desvantagens abaixo descritas:

- Tempo de maturação elevado, consoante a experiência do programador. Este fator pode ser superior, numa fase inicial de adoção do modelo de testagem automática;
- Necessidade de formação inicial;
- Morosidade na definição dos testes automáticos;
- Aumento no tempo de desenvolvimento de *software*, especialmente numa fase inicial do processo;
- Necessidade de efetuar a manutenção e sustentação da *framework* de testes automáticos;
- Impossibilidade de dispensar completamente o fator humano na testagem.



## 5. Boas práticas de outras organizações no que toca à utilização de modelos de automatização de testes no desenvolvimento e manutenção de SI

### 5.1 *Software*

Esta dimensão surge como o resultado dum ciclo de desenvolvimento. Este, pode se constituir como um novo produto a disponibilizar, ou uma alteração a um SI já existente. Como fim último de cada SI, procura-se que o mesmo introduza valor a um determinado processo. Este desiderato só será alcançado caso o *software* produzido vá ao encontro dos requisitos levantados e apresente uma disponibilidade contínua.

O primeiro conceito estruturante deste TII, automatização de testes de SI, apresenta-se como um contributo para o incremento da qualidade do *software* produzido. Segundo as palavras do Engenheiro R. Costa, após a implementação de testes automáticos, a qualidade do *software* aumenta substancialmente, assim como a manutenção do código desenvolvido. A mesma opinião é partilhada pelo Engenheiro C. Pereira,

[...] a introdução de testes automáticos permitiu a redução de tempo na execução de testes rotineiros, garantir que todos os micro serviços funcionam entre si e que quais falhas que sejam levadas até ao ambiente de Produtivo não passem despercebidas. (*op. cit.*).

A OutSystems, segundo o testemunho do Engenheiro R. Costa já incorpora, desde há 3 anos, e de uma forma generalizada, a testagem automática no processo de desenvolvimento de SI. A TalkDesk, no entanto e segundo as palavras do Engenheiro C. Pereira apresenta produtos em que a testagem automática está incorporada desde o início de vida dos mesmos, assim como outros produtos em que não existe uma *framework* de testes automáticos.

Não está identificada uma frequência ou situações específicas para a execução dos testes automáticos. Segundo o testemunho do Engenheiro C. Pereira a testagem automática é executada em diversas ocasiões,

[...] diariamente num *job* que é executado automaticamente todas as noites, para garantir que não terá ido nada para o ambiente de produtivo erradamente. Sempre que são efetuados *deploys* para os ambientes de *Staging*, Qualidade e Produtivo. E por fim, sempre que o QA<sup>4</sup> o entender na execução do seu trabalho. (*op. cit.*).

O tempo de indisponibilidade dos SI, graças à implementação de testes automáticos, diminuiu, como indica o Engenheiro R. Costa, assim como existe uma maior celeridade e

---

<sup>4</sup> Quality Assurance Engineer



grau de confiança na entrega de novas funcionalidades e alterações a funcionalidades existentes.

A média de tempo de indisponibilidade dos SI na TalkDesk, é segundo o Engenheiro C. Pereira de 0 minutos, uma vez que:

[...] esta empresa vende um serviço inteiramente disponibilizado na *Cloud* e com um *Service-Level Agreement* bastante exigente, pelo que todos os processos envolvidos no desenvolvimento e disponibilização dos Sistemas de Informação vão ao encontro de que os mesmos tenham o mínimo de tempo possível de indisponibilidade. (*op. cit.*).

No que toca à escalabilidade, ou capacidade para crescer, dum SI, o Engenheiro R. Costa afirmou que a empresa OutSystems utiliza testes de carga para o avaliar deste conceito.

## 5.2 Gestão

Como foi observado previamente, existem vantagens na implementação de testagem automática. A mudança de paradigma de manual para automática envolve fatores a ter em conta. Antes de mais o apoio das chefias, e sob esta matéria é bastante concludente a opinião partilhada pelo Engenheiro C. Pereira, quando questionado em que fase é mais importante o apoio da chefia,

[...] Desde o início do ciclo de vida de um Sistema de Informação. O terceiro princípio dos testes exposto no *Foundation Level* da certificação da ISTQB diz que “Testar cedo poupa tempo e dinheiro”. E de facto, para encontrar defeitos precocemente, os testes devem ser iniciados o mais cedo possível no ciclo de vida do desenvolvimento do *software*. Testar no início do ciclo de vida do desenvolvimento de *software* ajuda a reduzir ou eliminar alterações caras. Como tal, como a questão do Controlo de Qualidade tem uma vertente muito cultural da entidade/empresa, é fundamental que as chefias apoiem o processo de implementação de testes automáticos. (*op. cit.*).

Um outro fator a ter em conta é a decisão de automatizar ou não a testagem face à complexidade do SI. É unânime a opinião entre os dois entrevistados que poderá não fazer sentido automatizar sistemas pouco complexos. Como refere o Engenheiro R. Costa, numa primeira fase, deve ser automatizado o que é o mais prioritário ou impactante. R. Costa indica que a capacidade de automatização de testes durante cada *sprint* não é infinita, estimando-se nos 30%, e que esse facto deverá ser tido em conta no que deve ou não ser automatizado. Para o Engenheiro C. Pereira, a manutenção de uma *framework* de



automação de testes exige esforço e contante análise face às várias iterações e maturação do produto, devendo ser considerado se se justifica tal esforço para projetos de menor complexidade. Este facto é especialmente realçado por C. Pereira, podendo mesmo ser considerado uma desvantagem se descurado, dado que,

[...] se a *framework* de automação não for alvo de revisões e de manutenção, rapidamente poderão começar a surgir falsos negativos o que levará a um custo de tempo acrescido na análise dos testes que estão a falhar. (*op. cit.*).

Para a implementação de testes automáticos, existem um conjunto alargado de ferramentas disponíveis. A decisão de escolha está condicionada ao facto de o orçamento disponível ser maior ou menor, uma vez que nem todas as ferramentas são gratuitas, algumas exigindo licenciamento específico. O Engenheiro R. Costa indicou que a OutSystems utiliza um misto de ferramentas gratuitas e com licenciamento:

- *BDD Framework*;
- *Selenium*;
- *Appium*;
- *UI Path* (licenciamento específico).

O Engenheiro C. Pereira por sua vez, relatou que na TalkDesk a maioria das ferramentas são gratuitas, existindo apenas uma com licenciamento específico:

- *Selenium*;
- *Selenide*;
- *REST Assured*;
- *Cucumber* (licenciamento específico).

### 5.3 Riscos

Em qualquer processo os riscos estão presentes e a implementação de testes automáticos não foge a esta regra.

O Engenheiro R. Costa refere que, na OutSystems, o risco encontrado na implementação se prende com o tempo de entregas de *software*. No entanto este tempo adicional é uma troca. “Ir um pouco mais devagar para garantir um futuro a médio/longo prazo mais calmo” (R. Costa, *op. cit.*).

Para o Engenheiro C. Pereira o principal risco encontrado prende-se na criação duma *framework* demasiadamente complexa e mal mantida cujos resultados dos testes comecem



a não acrescentar valor ao ciclo de desenvolvimento. A forma de mitigação encontrada, segundo C. Pereira foi através duma contínua avaliação dos resultados dos testes e consequente análise daqueles que poderão estar a originar problemas. A TalkDesk, como testemunha C. Pereira, possui uma estrutura que permite partilhar conhecimento e dificuldades, assim como trabalhar conjuntamente na manutenção da *framework* de testes automáticos.

Paralelamente aos riscos, existem um conjunto de desafios inerentes ao processo de criação de testes automáticos. Para o Engenheiro R. Costa, na OutSystems, estes prendem-se por fazer entender ao cliente as vantagens em se entregar menos funcionalidades, mas devidamente enquadrados com testes automáticos. A melhor forma de ultrapassar este desafio passa, segundo o Engenheiro R. Costa, por demonstrar com factos, que a automatização acrescenta robustez às aplicações, assim como as torna mais fáceis de manter, enquanto auxilia os programadores a detetar e a corrigir *bugs*.

Para o Engenheiro C. Pereira o maior desafio passa por identificar quais os testes a automatizar.

[...] O maior desafio da implementação de testes automáticos é perceber que testes automatizar. É fundamental perceber que áreas do Sistema de Informação se pretende testar de forma automática. No caso da TalkDesk definiu-se que para um teste ser passível de ser automatizado deverá pelo menos existir iteração entre dois micro serviços, o que facilita bastante na altura de triar o que deverá ser automatizado ou não. (*op. cit.*).

#### 5.4 Síntese conclusiva e resposta à PD2

Em resposta à PD2, “Quais as boas práticas de outras organizações no que toca à utilização de modelos de automatização de testes no desenvolvimento e manutenção de SI?” enumeram-se as mesmas identificadas:

- Definição concreta da real necessidade de testagem automática;
- Perceber claramente que testes devem ser automatizados. Numa fase inicial escolher os processos mais prioritários e impactantes;
- Criação de uma *framework* de testagem adequada, sem excessiva complexidade que dificulte a manutenção e ponha em causa as vantagens da automatização;
- Manutenção e revisão constante da *framework* de automação;



- Avaliação dos resultados dos testes e consequente análise daqueles que poderão originar problemas;
- Criação duma estrutura que permita a partilha de conhecimento e dificuldades sentidas;
- Definição duma frequência ou situações específicas para a execução dos testes automáticos. Esta irá variar consoante o SI;
- Envolvimento das chefias e garantia do seu apoio no processo de adoção de modelos de automatização de testes;
- Criação dum mecanismo de controlo de qualidade na organização;
- Escolher qual ou quais as ferramentas de implementação de testes automáticos que mais se adequam à realidade da organização;



## **6. Aplicabilidade de um modelo de automatização de testes no desenvolvimento e manutenção de SI na FA**

Em resposta à PP: “Será que a introdução de automatização de testes no desenvolvimento e manutenção de SI na FA cria valor?” conclui-se que a automatização de testes é um modelo que acarreta vantagens às organizações, razão pela qual estas optam por introduzi-lo nos seus ciclos de desenvolvimento de *software*.

A FA, ainda não tem implementado nenhum modelo de automatização de testes no processo de desenvolvimento de SI, identificando todos os possíveis erros aplicativos através dum modelo de testagem manual. Esta é efetuada pelo programador numa primeira fase e por um conjunto de utilizadores específicos numa fase posterior.

A possibilidade de deteção precoce de erros aplicativos minimiza os riscos de uma incorreta programação ou testagem manual, garantindo uma maior qualidade e fiabilidade no produto final. Apesar do fator humano na testagem não poder ser eliminado, este modelo permite reduzir a testagem manual, contribuindo para uma melhoria na gestão de recursos, permitindo alocar os mesmos noutras tarefas que não sejam possíveis de automatizar. A introdução deste novo paradigma de testagem acarreta riscos que necessitam de ser mitigados para se alcançarem as vantagens previamente enunciadas deste modelo. O estudo prévio das duas organizações, OutSystems e TalkDesk permitiu identificar um conjunto de boas práticas que a serem consideradas e seguidas, contribuirão para esta diminuição do risco.

Conclui-se ainda que a FA poderá usufruir destas vantagens elencadas, refletindo as mesmas no *software* produzido, que será mais robusto e fiável permitindo uma maior satisfação e utilização por parte do utilizador final. A testagem automática permitirá diminuir tempos de testagem manuais, otimizando e simplificando este processo. O processo de manutenção de sistemas também sairá facilitado, garantindo que novos desenvolvimentos não afetam sistemas em produção. A inexperiência neste novo paradigma poderá ser colmatada com formação adequada permitindo dotar as equipas de desenvolvimento da DCSI com o conhecimento adequado para tirar partido deste modelo, de forma a ser acrescentado ainda mais valor ao *software* produzido.



## 7. Conclusões

A DCSI desenvolve SI que visam colmatar as necessidades do ramo nas suas variadas áreas de operação. Estes produtos aplicativos pretendem-se que sejam o mais fiéis possível aos requisitos levantados, por forma a que se constituam como uma ferramenta útil e que acrescentem valor aos processos institucionais da FA. Por entre as várias etapas dum ciclo de desenvolvimento, aquela que nunca poderá ser negligenciada é a fase de testagem. Esta poderá ser manual, automática ou um modelo misto. A DCSI utiliza na génese dos seus sistemas o modelo de testagem manual.

O presente TII teve por objeto de estudo a possibilidade de integração de um modelo de automatização de testes no desenvolvimento e manutenção de SI na FA.

Ao nível do procedimento metodológico, esta investigação caracteriza-se por um raciocínio indutivo, assente numa estratégia de investigação qualitativa. Em relação ao desenho de pesquisa utilizado, este enquadra-se num estudo de caso.

A estratégia de investigação qualitativa decorreu através de uma análise documental, assim como através da análise de conteúdo de sete entrevistas semiestruturadas. Destas, cinco foram efetuadas a entidades relevantes no ciclo de desenvolvimento de *software* em utilização na FA, e duas a colaboradores que exercem funções em organizações que já detêm modelos de testagem automática, possuindo conhecimento e experiência neste paradigma.

Tendo como ponto de partida o OG, “Analisar a aplicabilidade de um modelo de automatização de testes no desenvolvimento e manutenção de SI na FA”, definiu-se a PP, “Será que a introdução de automatização de testes no desenvolvimento e manutenção de SI na FA cria valor?”, assim como dois OE. O OE1, “Analisar as vantagens e desvantagens de um modelo de automatização de testes face aos processos atuais de desenvolvimento de SI na FA” e o OE2 “Analisar as práticas de outras organizações na utilização de modelos de automatização de testes no desenvolvimento e manutenção de SI”, cuja análise permitiu obter resposta às PD.

Por forma a responder à PD1, “Quais as vantagens e desvantagens na adoção de um modelo de automatização de testes face aos processos atuais de desenvolvimento de SI na FA?”, realizaram-se entrevistas a cinco entidades na FA de modo a entender as dimensões do conceito “Automatização de testes de SI”.



Conclui-se que a testagem automática tem como vantagens identificar precocemente erros aplicacionais e minimizar o impacto de novos desenvolvimentos em sistemas já em produção. Foi classificado como vantajosa a diminuição de tempo em ambiente de qualidade, refletindo-se numa maior rapidez de desenvolvimento. Associado a este paradigma, o fator humano surge como minimizado no processo, permitindo uma gestão mais eficaz deste recurso escasso. Como vantagens, concluiu-se ainda que uma testagem automática possibilita a construção de *software* mais fiável e com mais qualidade.

No entanto foi identificada a existência dum conjunto de desvantagens que terão de ser controladas. Nomeadamente, a necessidade de formação inicial e um elevado tempo de maturação que poderá depender consoante a experiência do programador. Foi observado ainda que o tempo de desenvolvimento poderá ser superior e que será necessário acautelar a manutenção das *frameworks* de testes automáticos. O fator humano mesmo minimizado não poderá ser totalmente dispensado, devendo existir alguns testes manuais.

Como resposta à PD2, “Quais as boas práticas de outras organizações no que toca à utilização de modelos de automatização de testes no desenvolvimento e manutenção de SI?”, analisaram-se duas entrevistas a entidades civis por forma a entender as dimensões do segundo conceito “Integração de modelos de automatização de testes de SI”.

Da sua análise, identificaram-se como boas práticas a necessidade de identificar claramente o que deve ser testado, assim como definir se existir uma real necessidade de testagem automática. No que toca à *framework* de testes desenvolvida, esta deverá ser adequada, sem ser excessiva, devendo ser despendido tempo suficiente para manutenção da mesma. O envolvimento das chefias deverá ser alcançado, tendo em vista a adoção de um modelo de automatização de testagem. Foi observado que para alcançar qualidade, deverá ser criado um mecanismo de controlo de qualidade na organização, assim como uma estrutura de partilha de conhecimento e dificuldades.

Pode-se então concluir que como resposta à PP, a introdução de um modelo de automatização de testes no desenvolvimento e manutenção cria valor e é uma mais-valia para a FA. Este modelo, não substituindo totalmente o fator humano, acarreta vantagens e desafios, permitindo explorar um novo paradigma, contribuindo para um *software* de qualidade, que se pretende ser o mais robusto e fiável possível.



Como contributo para o conhecimento, este estudo vem demonstrar a importância que a testagem automática poderá ter nos modelos de desenvolvimento de SI na FA, desde que mantidos sob controlo os riscos duma utilização incorreta da automatização.

Como limitação do estudo a não resposta por parte dos demais ADIAF, cujos contributos e experiências poderiam ser uma mais-valia, na obtenção de respostas e na criação de conhecimento. Recomenda-se como estudo futuro a análise desta implementação de forma a poder responder com métricas objetivas, quais as mais valias reais deste modelo.

Recomenda-se à DCSI a criação de um plano para implementar e operacionalizar o modelo de automatização de testes automáticos. Este, deverá contemplar formação, identificação e seleção de ferramenta de implementação de testes automáticos, assim como de processos críticos a automatizar num SI que possa servir como prova de conceito para o modelo em estudo.

Não existe uma forma única de adoção de um modelo desta natureza o que pode ser classificado como uma dificuldade na implementação do mesmo. Uma das claras dificuldades que poderá constituir-se como uma limitação, passará por identificar nitidamente que projeto poderá ser alvo de testagem automática, ou se por sua vez deverá ser mantido o modelo tradicional de testagem manual única. Esta limitação será diminuída à medida que as equipas de desenvolvimento da DCSI ganhem experiência e maturação no novo modelo.

Através deste TII apresentou-se um novo paradigma no que à testagem de *software* diz respeito. A adoção permitirá certamente aumentar o nível na qualidade e de fiabilidade dos produtos finais, assim como introduzir metodologias modernas em todo o modelo de desenvolvimento de SI. Este passo será toda uma mais-valia para a organização, que contará cada vez mais com SI robustos que contribuam para o sucesso da sua missão.



## Referências bibliográficas

- Amaral, E. (2021). *Primeiros passos para a Automatização de Testes de Software*. Retirado em 12 de março de 2022, de <https://www.computerworld.com.pt/2021/06/15/primeiros-passos-para-a-automatizacao-de-testes-de-software>
- Axelrod, A. (2018). *Complete Guide to Test: Techniques, Practices, and Patterns for Building and Maintaining Effective Software Projects*, Berkeley: Apress
- Carvalho, M. (2010). *Automatização de Testes de Software*. (Relatório de Estágio para obtenção do Grau de Mestre em Automação e Comunicações em Sistemas de Energia). Instituto Politécnico de Coimbra, Coimbra: Autor.
- Divisão de Comunicações e Sistemas de Informação. (2015). *Plano Diretor dos Sistemas de Informação da Força Aérea*. Lisboa: Autor.
- Instituto Universitário Militar. (setembro de 2020). NEP/INV - 003 (O). *Estrutura e Regras de Citação e Referenciação de Trabalhos Escritos a Realizar no IUM*. Pedrouços, Lisboa, Portugal: Autor.
- Instituto Universitário Militar. (setembro de 2020). NEP/INV - 001 (O). *Trabalhos de Investigação*. Pedrouços, Lisboa, Portugal: Autor.
- MCLAFA 305-2. (2013). *Organização e Normas de Funcionamento da Direção de Comunicações e Sistemas de Informação*. Lisboa: Força Aérea, Comando da Logística.
- Outsystems. (2022). *Test Automation in the Delivery Lifecycle*. Retirado em 21 de março de 2022, de [https://success.outsystems.com/Documentation/11/Managing\\_the\\_Applications\\_Lifecycle/Test\\_Automation\\_in\\_the\\_Delivery\\_Lifecycle](https://success.outsystems.com/Documentation/11/Managing_the_Applications_Lifecycle/Test_Automation_in_the_Delivery_Lifecycle)
- Pittet, S. (s.d.). *The different types of software testing*. Retirado em 14 de março de 2022, de <https://www.atlassian.com/continuous-delivery/software-testing/types-of-software-testing>
- Ribeiro, A. (2016). *A relevância dos sistemas de informação para os negócios*. Retirado em 20 de março de 2022, de <https://www.itchannel.pt/news/negocios/a-relevancia-dos-sistemas-de-informacao-para-os-negocios>
- Rossel, S. (2017). *Continuous Integration, Delivery, and Deployment*, Berkeley: Packt Publishing



- Santos, L.A.B., & Lima, J.M.M. (Coord.) (2019). *Orientações metodológicas para a elaboração de trabalhos de investigação (2.ª ed., revista e atualizada)*. Cadernos do IUM, 8. Lisboa: Instituto Universitário Militar
- Silva, M., & Dallilo, F. (2019). *Uma visão geral sobre automação de testes*. Retirado em 11 de março de 2022, de <https://www.researchgate.net/publication/341141370>
- Tian, J. (2005). *Software Quality Engineering: Testing, Quality Assurance, and Quantifiable Improvement*, Dallas: IEEE Computer Society Press
- Walter, M. (2017). *Test Automation, 20 Years After*. Obtido em 11 de março de 2022, de <https://www.agilealliance.org/resources/experience-reports/test-automation-20-years-after/>



## Apêndice A– Modelo de análise

| TEMA                                                                                                                                         | AUTOMATIZAÇÃO DE TESTES NO DESENVOLVIMENTO E MANUTENÇÃO DE SISTEMAS DE INFORMAÇÃO NA FORÇA AÉREA                                                          |                                                        |                  |                          |                                                        |
|----------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------|------------------|--------------------------|--------------------------------------------------------|
| Objetivo Geral                                                                                                                               | Analisar a aplicabilidade de um modelo de automatização de testes no desenvolvimento e manutenção de SI na FA                                             |                                                        |                  |                          |                                                        |
| Pergunta de Partida                                                                                                                          | Será que a introdução de automatização de testes no desenvolvimento e manutenção de SI na FA cria valor?                                                  |                                                        |                  |                          |                                                        |
| Objetivos Específicos                                                                                                                        | Perguntas Derivadas                                                                                                                                       | Conceitos                                              | Dimensões        | Indicadores              | Técnicas de recolha de dados                           |
| OE1 – Analisar as vantagens e desvantagens de um modelo de automatização de testes face aos processos atuais de desenvolvimento de SI na FA. | PD1 - Quais as vantagens e desvantagens na adoção de um modelo de automatização de testes face aos processos atuais de desenvolvimento de SI na FA?       | Automatização de testes de SI                          | Implementação    | Ferramentas              | Análise Documental<br><br>Entrevistas Semiestruturadas |
|                                                                                                                                              |                                                                                                                                                           |                                                        |                  | Requisitos               |                                                        |
|                                                                                                                                              |                                                                                                                                                           |                                                        |                  | Ciclo de Desenvolvimento |                                                        |
|                                                                                                                                              |                                                                                                                                                           |                                                        | Testes           | Tipologias               |                                                        |
|                                                                                                                                              |                                                                                                                                                           |                                                        | Recursos Humanos | Cursos Formação          |                                                        |
|                                                                                                                                              |                                                                                                                                                           |                                                        |                  | Experiência              |                                                        |
|                                                                                                                                              |                                                                                                                                                           |                                                        |                  | Funções                  |                                                        |
| OE2 – Analisar as práticas de outras organizações na utilização de modelos de automatização de testes no desenvolvimento e manutenção de SI. | PD2 – Quais as boas práticas de outras organizações no que toca à utilização de modelos de automatização de testes no desenvolvimento e manutenção de SI? | Integração de modelos de automatização de testes de SI | Software         | Qualidade                |                                                        |
|                                                                                                                                              |                                                                                                                                                           |                                                        |                  | Escalabilidade           |                                                        |
|                                                                                                                                              |                                                                                                                                                           |                                                        | Gestão           | Económica                |                                                        |
|                                                                                                                                              |                                                                                                                                                           |                                                        |                  | Mudança                  |                                                        |
|                                                                                                                                              |                                                                                                                                                           |                                                        | Riscos           | Identificação            |                                                        |
|                                                                                                                                              | Desafios                                                                                                                                                  |                                                        |                  |                          |                                                        |



## **Apêndice B – Guião de entrevistas semiestruturadas**

De forma a dar resposta aos indicadores identificados, foram identificadas 37 questões distintas que foram agrupadas consoante as funções e conhecimentos dos elementos, em guiões de entrevistas. De seguida apresentam-se as questões colocadas a cada entrevistado.

### **Guião de entrevista para o ADIAO, Coronel Luís Cordeiro e respetivas respostas**

#### **Q1 - Como define os Sistemas de Informação desenvolvidos pela FA em termos de fiabilidade e de taxa de disponibilidade?**

Os projetos de Sistemas de Informação atravessam uma fase de maturação e ganho de credibilidade, nomeadamente nos diferentes momentos da análise, desenvolvimento, testes, aceitação e *initial operational capability* (IOC).

Os Sistemas de Informação desenvolvidos pela Força Aérea são projetados em função dos requisitos identificados para o apoio às diversas missões, o que lhes confere, teoricamente, um produto final que contempla não só as necessidades, mas também um fluxo de interação e usabilidade muito próximos da realidade.

Atualmente, os referidos Sistemas de Informação têm um grau de confiança elevado, quer ao nível da sua qualidade como também da sua credibilidade.

A taxa de disponibilidade dos referidos sistemas é alta e apenas é comprometida em curtos momentos decorrentes de ações de suporte e manutenção programadas ou de eventuais limitações nas comunicações.

#### **Q2 - Seguem alguma metodologia de desenvolvimento de software? Se sim, qual e desde quando?**

A metodologia AGILE, (*framework* SCRUM), já explorada na Força Aérea há alguns anos, aproxima os gestores de projeto e utilizadores finais das áreas afetas ao desenvolvimento e contribui para uma maior confiança e crédito aos Sistemas de Informação das referidas áreas. Destaca-se, em particular, um maior envolvimento dos utilizadores em todos os momentos, desde o levantamento de requisitos e fases de desenvolvimento até aos testes de aceitação.

#### **Q3 - Durante o ciclo de desenvolvimento/sustentação são efetuados algum tipo de testes estruturados manuais ou automáticos?**

Apenas temos conhecimento de testes manuais realizados pelos gestores e utilizadores finais durante o ciclo de desenvolvimento/sustentação.

#### **Q4 - Quais as mais valias do desenvolvimento de Sistemas de Informação pela FA comparativamente a Sistemas de Informação adquiridos?**

Tal como respondido na Q1, os Sistemas de Informação desenvolvidos pela Força Aérea são projetados à medida dos requisitos necessários para o apoio à missão e asseguram disponibilidade técnica. O que confere, teoricamente, um produto que contemple não só as necessidades, mas um fluxo de interação e usabilidade muito próximo da realidade assim como um suporte mais célere e à medida, contribuindo para um grau de confiança elevado.

#### **Q5 - Quais os maiores problemas do desenvolvimento de Sistemas de Informação pela FA comparativamente a Sistemas de Informação adquiridos?**

O maior problema tem a ver com a falta de capacidade de resposta face às inúmeras solicitações para desenvolvimento de novos Sistemas de Informação, bem como a evolução e suporte dos existentes. As crescentes necessidades de automatização de processos, a falta de recursos humanos internos especializados e o foco nos sistemas críticos para a missão primária da Força Aérea, constituem fatores que causam constrangimentos aos avanços tecnológicos necessários.

#### **Q6 - Com que frequência (diariamente, semanalmente, mensalmente) são detetados erros nos Sistemas de Informação já em ambiente produtivo?**

A resposta não é assertiva, pois depende muito do grau de maturidade do sistema, da complexidade e criticidade do mesmo, assim como do número de interações.

Os Sistemas de Informação mais críticos para a missão primária da Força Aérea tendem a estabilizar com os melhoramentos introduzidos e a frequência da deteção de erros diminui gradualmente.

#### **Q7 - Qual o impacto destes erros em ambiente produtivo?**

A resposta também não é assertiva pelas mesmas razões já explanadas na resposta anterior.

Ainda assim, atualmente, o impacto dos referidos erros tem sido menor.

#### **Q8 - Quanto tempo demora em média para ser corrigido?**

Na correção de erros de cariz mais complexo, o tempo de demora tem sido considerável pelas razões explanadas, em parte, na resposta à questão Q5.



**Q9 - Durante o levantamento de requisitos são identificados testes de aceitação que o Sistema de Informação deve passar?**

Nem sempre, por vezes os testes de aceitação ocorrem quando um grupo de requisitos que contribuem para a resposta à mesma necessidade são alcançados.

**Q10 - Em que fase do desenvolvimento são feitos testes?**

Tal como respondido na Q10, “por vezes os testes de aceitação ocorrem quando um grupo de requisitos que contribuem para a resposta à mesma necessidade são alcançados”.

Por outro lado, existem também casos de SI em que os testes são feitos apenas no final do seu desenvolvimento o que provoca, por vezes, grande impacto face ao diferente entendimento de alguns processos entre a equipa de desenvolvimento e o utilizador final.

**Q11 - Antes de entrada em produtivo, os últimos testes são efetuados por quem? Considera que se estes fossem feitos de forma autónoma seria vantajoso?**

Os testes são primariamente realizados pelas áreas afetas aos gestores dos projetos com o apoio, no momento ou à posteriori, de utilizadores finais pré-identificados.

Não nos parece que os testes realizados de forma autónoma sejam vantajosos. Podem conflitar com a disponibilidade e afetação ao propósito dos próprios Sistemas de Informação.

Na nossa opinião, a vantagem da realização de testes de forma autónoma serve apenas para identificação de erros grosseiros.

**Q12 - Existem alguma *checklist* de testes a efetuar antes da entrada de cada versão em produtivo? Considera que se estes fossem feitos de forma autónoma seria vantajoso?**

Os testes a realizar seguem um modelo idêntico a *checklist*.

A realização de testes aleatórios é vantajosa, pois não cumpre com métodos pré-estabelecidos e que podem estar comprometidos e podem contribuir para a deteção de erros não espetáveis assim como a identificação de outras modalidades de interação mais *friendly*.

**Guião de entrevista para o Gabinete do ADIAL, Capitão Marta Matos e respetivas respostas**

**Q1 - Como define os Sistemas de Informação desenvolvidos pela FA em termos de fiabilidade e de taxa de disponibilidade?**

Os sistemas de informação (SI) são uma pedra angular e de grande importância em qualquer área de negócio, dado o seu impacto na sustentabilidade e evolução do mesmo. Qualquer organização requer SI que sustentem os seus processos de natureza operacional, logística e de gestão dos seus recursos humanos. Embora com diferentes requisitos em função da atividade, a desmaterialização documental, a centralização e partilha da informação são indispensáveis.

No entendimento deste Gabinete (GabADIAL), acreditamos que a FA ainda tem um longo caminho a percorrer neste domínio, e o facto de estarmos sob o domínio da Defesa, obrigando-nos indubitavelmente a crescer na mesma medida em que o nosso negócio vai evoluindo, desde operarmos tecnologicamente com aeronaves mais sofisticadas (integrando possíveis ajustes em sede de SI), passando por termos um quadro de indicadores que preconizam as necessidades logísticas para daqui a 5 anos, entre outras.

Tendo em consideração, que qualquer SI tem de ter um elevado comprometimento da gestão de topo, todo o trabalho que é realizado desde o indivíduo que está na unidade a carregar uma obra de assistência, ao gestor que programa as cartas de inspeção no SI, passando pelos militares no ADIAL que ministram também formação (por forma a que todos os *stakeholders* introduzam corretamente os dados em sistema), e não esquecendo por exemplo os militares que programam na DCSI, TODOS são importantes para que qualquer SI funcione em função do negócio.

Os SI da competência do ADIAL tem, por isso, uma taxa de fiabilidade e de disponibilidade elevada. Ainda assim, acredito que podemos melhorar, pois o foco na eficiência e melhoria contínua, a disponibilidade e a qualidade da informação para apoio à decisão são fatores de sucesso críticos e incontornáveis.

**Q2 - Seguem alguma metodologia de desenvolvimento de *software*? Se sim, qual e desde quando?**

Um dos objetivos deste GabADIAL será a adoção da ISO 20000. Estamos a falar de suporte de 1º nível e desenvolvimento, assim como o ato de documentar toda a gestão de projeto. Isto é importante, na medida, em que tudo o que é feito fica documentado, e são seguidas regras.

O desenvolvimento de *software*, contempla de forma lata a gestão de projeto.

A metodologia passa inicialmente por um pedido formal ao ADIAL. Qualquer pedido quando entra no ADIAL, deve-se consubstanciar como aplicável a toda a organização e deve seguir o princípio de acrescentar valor à mesma, o qual será apresentado ao Grupo Coordenador dos Sistemas de Informação, que deverá aprová-lo.



Verificando-se estas três condições, a criação de um processo ou desenvolvimento de software:

- Inicia-se com o estado de arte do processo, o chamado AS IS, onde o ADIAL pretende mapear o processo tal como está a funcionar na FA;
- Simultaneamente é importante perceber junto dos utilizadores, normalmente das várias Unidades da FA como funciona o processo, ou como (caso seja algo que não exista) se pretende que funcione;
- O ADIAL desenha o respetivo processo futuro, designado por TO BE;
- É nesta fase que em reuniões entre o ADIAL e os utilizadores, por forma a aferir que o projeto servirá na generalidade os propósitos dos utilizadores, em que se fazem correções/alterações de algumas funcionalidades/processos;
- Escrevem-se os requisitos, para serem enviados posteriormente à DCSI;
- Em reuniões entre o ADIAL e a DCSI, serão afinados pormenores do que foi projetado;
- Os requisitos são alvo de desenvolvimento pela equipa de programação, e colocados já em formato de funcionalidades no módulo de qualidade (QLA), para serem testadas pelo ADIAL;
- Depois das funcionalidades serem devidamente testadas e sem erros, são migradas para o módulo de produção, de forma a ficarem disponíveis para o utilizador.

Como vimos pela metodologia acima descrita, é absolutamente essencial a profissionalização dos militares que trabalham nesta área, sendo a formação um pilar essencial.

**Q3 - Durante o ciclo de desenvolvimento/sustentação são efetuados algum tipo de testes estruturados manuais ou automáticos?**

Como já evidenciado na resposta à Q2, os testes efetuados, em qualquer um dos ciclos, desenvolvimento ou sustentação, é efetuado manualmente pelos militares que se encontram colocados no ADIAL, utilizando a sua experiência de SI ou módulos anteriores. Neste momento, existem 4 militares, para esse desiderato.

**Q4 - Quais as mais valias do desenvolvimento de Sistemas de Informação pela FA comparativamente a Sistemas de Informação adquiridos?**

As mais valias no desenvolvimento de SI serão sempre para o core da FA, e para os seus recursos humanos. Naturalmente, que será sempre a gestão de topo a definir as prioridades. No entanto, caso seja um SI com impacto para os objetivos estratégicos da organização, e com data limite de projeto, terá de ser sempre equacionado a aquisição via outsourcing, analisando financeiramente o caso. O desenvolvimento no ADIAL é apenas mais um vértice das nossas funções, acumulando a formação às Unidades e respetiva sustentação dos diversos SI, portanto, quando se opta por fazer projetos internamente, como se tem feito até agora, com os quantitativos que o serviço dispõe, a tónica temporal não pode ser prioritária. Neste sentido, qualquer desenvolvimento de projeto levará mais tempo a ser concretizado.

**Q5 - Quais os maiores problemas do desenvolvimento de Sistemas de Informação pela FA comparativamente a Sistemas de Informação adquiridos?**

Os SI não desenvolvidos internamente, mas utilizados pela FA, e no qual o ADIAL têm a gestão (SIG, ACINGOV, entre outros), faz-se apenas a sua sustentação, basicamente a gestão de acessos. Já em SI, desenvolvidos pela FA, e da gestão do ADIAL, existe todo um trabalho de desenvolvimento/melhoria, sustentação, averiguação de erros que temos de dar resposta, muitas vezes em tempo útil.

**Q6 - Com que frequência (diariamente, semanalmente, mensalmente) são detetados erros nos Sistemas de Informação já em ambiente produtivo?**

Muito raramente, aparecem erros em produção.

**Q7 - Qual o impacto destes erros em ambiente produtivo?**

Quando existem, O SI continua operacional, afetando normalmente apenas uma funcionalidade específica.

**Q8 - Quanto tempo demora em média para ser corrigido?**

Depende do tipo de erro, no entanto a maioria é corrigido no próprio dia.

**Q9 - Durante o levantamento de requisitos são identificados testes de aceitação que o Sistema de Informação deve passar?**

Sim. Conforme respondido em Q2, qualquer gestão de projeto deve documentar os testes a serem efetuados pelo gestor de projetos.

**Q10 - Em que fase do desenvolvimento são feitos testes?**

Conforme referido em Q2, os requisitos preconizados pelo ADIAL, são alvo de programação pela DCSI, e colocados já em formato de funcionalidades no módulo de qualidade (QLA), de forma a poderem ser testadas pelo ADIAL.

**Q11 - Antes de entrada em produtivo, os últimos testes são efetuados por quem? Considera que se estes fossem feitos de forma autónoma seria vantajoso?**

Os testes são efetuados pelo ADIAL. Tendo em consideração as múltiplas tarefas executadas no ADIAL, e sabendo dos constrangimentos em termos de efetivos neste serviço, seria claramente vantajoso.



**Q12 - Existem alguma *checklist* de testes a efetuar antes da entrada de cada versão em produtivo? Considera que se estes fossem feitos de forma autónoma seria vantajoso?**

Antes da entrada de qualquer versão em produtivo, existe uma *checklist* a percorrer. A vantagem prende-se com tarefas repetitivas existentes nesses mesmos testes e com o facto de podermos alocar recursos humanos noutras tarefas, tão pertinentes como os testes, ou abreviar esses mesmos testes. Antes da entrada em produtivo de uma nova versão é sempre efetuado o conjunto de testes da *checklist* para verificar, para além do bom funcionamento das novas alterações, verificar também se nenhuma das funcionalidades anteriores foi afetada.

**Guião de entrevista para o ADIAFIN, Major Paulo Sousa e respetivas respostas**

**Q1 - Como define os Sistemas de Informação desenvolvidos pela FA em termos de fiabilidade e de taxa de disponibilidade?**

Considero que os sistemas desenvolvidos pela FA apresentam uma elevada fiabilidade e disponibilidade

**Q2 - Seguem alguma metodologia de desenvolvimento de *software*? Se sim, qual e desde quando?**

Na maioria dos casos é seguido o procedimento definido no RFA 391-1 (A) - Regulamento de gestão da informação da Força Aérea, para a definição da pertinência e das prioridades no desenvolvimento de *software*. Após aprovação superior para o desenvolvimento, com base em requisitos macro, são elaborados os requisitos específicos, analíticos, pelos diversos órgãos envolvidos. Esta metodologia tem vindo a ser afinada e, no caso particular do desenvolvimento de um novo sistema de informação de apoio à faturação, atual, nota-se que existe um maior foco na metodologia.

**Q3 - Durante o ciclo de desenvolvimento/sustentação são efetuados algum tipo de testes estruturados manuais ou automáticos?**

Sim, após qualquer desenvolvimento este é colocado em ambiente de testes para validação pelos utilizadores. Só depois desta validação é que se efetua a passagem para ambiente produtivo.

**Q4 - Quais as mais valias do desenvolvimento de Sistemas de Informação pela FA comparativamente a Sistemas de Informação adquiridos?**

A principal vantagem é a maior customização e facilidade na assistência.

**Q5 - Quais os maiores problemas do desenvolvimento de Sistemas de Informação pela FA comparativamente a Sistemas de Informação adquiridos?**

A falta de pessoal para responder a todas as solicitações, o que coloca diversos sistemas em espera.

**Q6 - Com que frequência (diariamente, semanalmente, mensalmente) são detetados erros nos Sistemas de Informação já em ambiente produtivo?**

Depende da maturidade do sistema. Se em início de exploração temos erros detetados semanalmente. À medida que nos distanciamos temporalmente do início de exploração a frequência dos erros diminui.

**Q7 - Qual o impacto destes erros em ambiente produtivo?**

Depende do tipo de erro e do sistema em operação. Podemos ter erros que são contornáveis através de processo ou suportáveis até à sua resolução, assim como há casos em que o erro origina falha no cumprimento de obrigações legais/contratuais, de maior gravidade, e que exigem rápida resolução.

**Q8 - Quanto tempo demora em média para ser corrigido?**

Depende do tipo de erro, da respetiva complexidade e do seu impacto no cumprimento de obrigações legais/contratuais. Conforme referido na questão anterior, há erros que podem ser contornados com alteração de processos e/ou procedimentos e, portanto, aceita-se que haja um elevado tempo de correção. Outros há que implicam resolução, o mais rápido possível, por implicarem impacto na missão da Força Aérea.

**Q9 - Durante o levantamento de requisitos são identificados testes de aceitação que o Sistema de Informação deve passar?**

Da minha experiência a definição dos testes de aceitação na FAP é efetuado ad hoc, deixando ao critério do “dono” do programa a definição desses testes.

**Q10 - Em que fase do desenvolvimento são feitos testes?**

A maior parte dos testes é efetuada após passagem de ambiente de desenvolvimento para ambiente de qualidade/testes. Durante a fase de programação podem ser feitos alguns testes para aferir da concordância com os requisitos definidos.

**Q11 - Antes de entrada em produtivo, os últimos testes são efetuados por quem? Considera que se estes fossem feitos de forma autónoma seria vantajoso?**

Os últimos testes são efetuados pelos utilizadores do sistema, através da definição de alguns Key Users. Sim, é vantajoso que os utilizadores testem livremente todas as funcionalidades do sistema para minimizar os erros a detetar na exploração futura em produtivo.



**Q12 - Existem alguma checklist de testes a efetuar antes da entrada de cada versão em produtivo? Considera que se estes fossem feitos de forma autónoma seria vantajoso?**

Quando se efetuam os testes é solicitado aos Key Users para testar “em especial” determinadas funcionalidades ou tipos de registo, mas não se limita os testes que podem fazer. A realização de testes de forma totalmente autónoma acarreta o risco de alguma funcionalidade não vir a ser testada, pelo que se considera pertinente definir o mínimo a testar, mas deixar em aberto outros testes que entendam efetuar.

**Guião de entrevista para o chefe da Sub-Repartição de Desenvolvimento de Sistemas de Informação, Capitão Carlos Pombeiro e respetivas respostas**

**Q1 - Como define os Sistemas de Informação desenvolvidos pela FA em termos de fiabilidade e de taxa de disponibilidade?**

Os SI da FA são desenvolvidos segundo os requisitos elencados pelos ADIAF de cada área. De acordo com estes inputs assim são construídos e colocados em ambiente de qualidade para testes aplicacionais. Quando maior o tempo investido no processo de testagem, maior a fiabilidade que o sistema desenvolvido vai ao encontro das necessidades do ramo e da área de negócio respetiva. Tendo em conta a situação atual, os sistemas que se encontram em produção à mais tempo dispõem duma elevada fiabilidade pois a sua operação e tempo em testagem já permitiu depurar muitos dos possíveis erros de codificação que são intrínsecos ao processo de desenvolvimento. Respondendo diretamente à sua questão e tendo em consideração a realidade atual, a FA dispõe de sistemas com elevada fiabilidade fruto da sua longevidade em operação. Os sistemas mais recentes possuem a fiabilidade possível, fruto da testagem efetuada pelo EPR e da operação diária. Globalmente a fiabilidade dos sistemas desenvolvidos é bastante boa como atestam as opiniões diárias de utilizadores e EPRS, assim como a taxa de utilização.

No que toca a disponibilidade a taxa respetiva (não dispondo de números) é bastante elevada, uma vez que os sistemas estão continuamente disponíveis, exceção feita em algumas atualizações do software que impliquem paragem, por exemplo, mudança na estrutura de dados, ou em problemas ou *updates* da infraestrutura aplicacional ou de base de dados que impliquem paragem.

**Q2 - Seguem alguma metodologia de desenvolvimento de software? Se sim, qual e desde quando?**

A DCSI utiliza a metodologia AGILE desde 2011, apostando num desenvolvimento iterativo permitindo disponibilizar software ao utilizador final mais rapidamente, envolvendo continuamente os EPR e utilizadores neste processo, garantindo que o software produzido tem mais qualidade e vai ao encontro das reais necessidades da organização.

**Q3 - Durante o ciclo de desenvolvimento/sustentação são efetuados algum tipo de testes estruturados manuais ou automáticos?**

No ciclo de desenvolvimento e tendo em conta a situação atual os testes iniciais são feitos pelos programadores, (o próprio ou em regime de *cross testing*). Quando terminados, o *software* é disponibilizado em ambiente de qualidade para que os EPR efetuem testes mais dedicados e exaustivos possíveis, tendo como objetivo garantir que o *software* faz aquilo que se espera dele. A forma como cada EPR faz os testes é da sua responsabilidade, não intervindo a DCSI na forma ou estrutura dos mesmos.

Infelizmente a DCSI não realiza nenhum tipo de testes automáticos.

**Q4 - Quais as mais valias do desenvolvimento de Sistemas de Informação pela FA comparativamente a Sistemas de Informação adquiridos?**

Existem consideráveis ganhos no desenvolvimento de *software* pelo próprio ramo:

- A garantia que o sistema de informação é construído à medida das necessidades do ramo
- Inexistência de custos de desenvolvimento ou de manutenção do sistema de informação
- A permanente disponibilidade (caso superiormente decidido) para evoluir o sistema sem custos adicionais
- Evolução tecnológica do ramo garantindo a atualização dos sistemas de informação face à constante volatilidade e inovação do mundo informático
- Independência tecnológica e contratual face a empresas civis de desenvolvimento
- A existência de recursos humanos com capacidade técnica fazer face às necessidades de desenvolvimento do ramo

**Q5 - Quais os maiores problemas do desenvolvimento de Sistemas de Informação pela FA comparativamente a Sistemas de Informação adquiridos?**

Apesar de ser uma vantagem o desenvolvimento interno, este modelo também tem questões a rever, nomeadamente:

- Falta de recursos humanos



- Falta de recursos humanos qualificados. É cada vez mais difícil recrutar militares com formação na área das tecnologias de informação. Esta dificuldade tem sido colmatada com recrutamento de militares sem formação mínima para o desenvolvimento, o que aumenta o tempo de preparação dum militar para estar minimamente apto para trabalhar como programador de sistemas de informação
- Falta de formação interna
- Inexistência clara de definição de prioridades superiormente definidas (o que assistimos diariamente é que tudo é prioritário)
- Demasiadas lacunas de sistemas, o que impossibilita, conjuntamente com os fatores acima mencionados, conseguir dar uma resposta cabal às reais necessidades da organização
- Projetos inacabados, ou seja, um sistema inicia a sua operação, mas nunca termina efetivamente o seu desenvolvimento, fruto das constantes mudanças e alterações pedidas.
- Dificuldades na definição clara de requisitos por parte dos EPR. Esta dificuldade é especialmente sentida quando há rotação de militares nestes gabinetes.

**Q6 - Com que frequência (diariamente, semanalmente, mensalmente) são detetados erros nos Sistemas de Informação já em ambiente produtivo?**

Não existe uma matriz ou uma estatística referente a erros detetados nos sistemas de informação, mas garantidamente todas as semanas são detetadas situações ou erros nos sistemas de informação FAP. Estas situações são mais comuns nos sistemas mais recentes ou pontualmente em alguns sistemas *Legacy*.

**Q7 - Qual o impacto destes erros em ambiente produtivo?**

O impacto é elevado dependendo da situação ou do erro detetado. Por vezes ocorrem erros que derivam de uma incorreta utilização dos sistemas e que originam situações indesejáveis que obrigam a intervenção do programador. Sempre que algo foge do normal funcionamento esperado existe um impacto que não pode ser negligenciado e que deve ser corrigido, pois compromete a missão de determinado sistema podendo colocar em causa a sua utilização.

**Q8 - Quanto tempo demora em média para ser corrigido?**

O tempo deverá ser sempre o mínimo possível. Nem sempre se conseguem obter os tempos desejados por falta de mão de obra disponível, ou com conhecimento para corrigir determinado erro. Na RSI procura-se responder o mais rapidamente possível e corrigir os erros que surgem. Sempre que um erro impeça a disponibilidade dum sistema, esse erro é imediatamente tratado passando à frente de eventuais tarefas a decorrer. O tempo para corrigir dependerá do erro em si e do conhecimento do programador para colmatar o mesmo.

**Q9 - Durante o levantamento de requisitos são identificados testes de aceitação que o Sistema de Informação deve passar?**

Infelizmente estes testes ainda não são levantados, nem durante os requisitos, nem em nenhuma outra fase do desenvolvimento.

**Q10 - Em que fase do desenvolvimento são feitos testes?**

A RSI utiliza AGILE no desenvolvimento. Esta metodologia preconiza o desenvolvimento iterativo de sistemas. Em cada ciclo iterativo, denominado *sprint* os programadores desenvolvem um conjunto de tarefas que apresentam ao EPR no final de cada ciclo. Depois desta reunião, ou demo, o EPR terá acesso às mesmas em ambiente de qualidade e é nesta fase que irá efetuar os testes que ache necessários antes de autorizar a passagem destes desenvolvimentos para produção.

**Q11 - Antes de entrada em produtivo, os últimos testes são efetuados por quem? Considera que se estes fossem feitos de forma autónoma seria vantajoso?**

Antes da entrada em produtivo os últimos testes são efetuados pelo EPR a quem foi apresentado o resultado do último ciclo de desenvolvimento. Existem claras vantagens na existência de testes automáticos, pois permitiria identificar precocemente a existência de erros aplicacionais, quer ao nível de novas funcionalidades, quer na garantia que de um novo ciclo não influenciaria negativamente os ciclos anteriores. No entanto a existência de testes automáticos não dispensará nunca a necessidade de testes por parte do EPR.

**Q12 - Existem alguma *checklist* de testes a efetuar antes da entrada de cada versão em produtivo? Considera que se estes fossem feitos de forma autónoma seria vantajoso?**

Por parte da DCSI não existe nenhuma *checklist* de testes a efetuar antes da entrada em produtivo numa nova versão. Mas pelos factos acima enunciados será de todo vantajoso criar-se uma *checklist* e um mecanismo de execução de testes automáticos. A qualidade do software aumentaria substancialmente.

**Q13 - Quais as linguagens de desenvolvimento usadas?**

A DCSI utiliza essencialmente COBOL e Visual Basic 6 nas suas aplicações *legacy* de primeira e segunda geração respetivamente. Para o desenvolvimento WEB é utilizado PHP. Para os novos sistemas de informação em ambiente PLUS é utilizada a ferramenta *Service Studio* da empresa OutSystems que gera



código .NET. Transversalmente às várias aplicações é utilizada a linguagem SQL para interagir com a base de dados.

**Q14 - Quais as ferramentas/programas utilizados para a automatização de testes?**

A DCSI não dispõe de qualquer ferramenta para a automatização de testes.

## **Guião de entrevista para o chefe da Sub-Repartição de Administração de Sistemas de Informação, Capitão Maria Côrte-Real e respetivas respostas**

**Q1 - Como define os Sistemas de Informação desenvolvidos pela FA em termos de fiabilidade e de taxa de disponibilidade?**

Não temos nenhuma forma de medir efetivamente quer a fiabilidade quer a taxa de disponibilidade. O que posso acrescentar será em termos de perceção que tenho desses aspetos.

No que diz respeito à fiabilidade penso que as aplicações desenvolvidas na RSI têm, por parte da maioria dos utilizadores, uma taxa de confiabilidade e credibilidade elevada. Embora por vezes sejam necessárias algumas correções, acredito que a maioria acredita e confia nos sistemas que usa.

Relativamente à taxa de disponibilidade, sem querer lançar nenhum valor específico, penso que é bastante elevada também.

**Q2 - Seguem alguma metodologia de desenvolvimento de software? Se sim, qual e desde quando?**

Desde 2011 que utilizamos a metodologia Agile com recurso à *framework* Scrum e em 2019 começámos a utilizar também o *Kanban*. Atualmente, em alguns projetos, usamos uma combinação dos dois.

**Q3 - Durante o ciclo de desenvolvimento/sustentação são efetuados algum tipo de testes estruturados manuais ou automáticos?**

Não são efetuados testes estruturados durante o desenvolvimento, apenas é testado se o *software* tem o comportamento esperado de acordo com o que foi pedido.

**Q4 - Quais as mais valias do desenvolvimento de Sistemas de Informação pela FA comparativamente a Sistemas de Informação adquiridos?**

Os Sistemas de Informação desenvolvidos pela FA são feitos à medida dos utilizados, ou pelo menos são desenvolvidos com os requisitos especificados pelas EPR. Existe também a vantagem de em qualquer momento poderem alterar ou acrescentar requisitos, tendo como único impacto para o utilizador o prazo de entrega.

**Q5 - Quais os maiores problemas do desenvolvimento de Sistemas de Informação pela FA comparativamente a Sistemas de Informação adquiridos?**

Na minha perspetiva existem dois grandes problemas associados ao desenvolvimento na FA. Por um lado, o facto de as EPRs não possuírem formação que as habilite a saber definir requisitos, efetuar testes e a terem uma perspetiva do produto final que pretendem, o que leva a constantes alterações e correções, e por outro a escassez de recursos humanos na parte do desenvolvimento e a faltas de especialização desses mesmos recursos.

**Q6 - Com que frequência (diariamente, semanalmente, mensalmente) são detetados erros nos Sistemas de Informação já em ambiente produtivo?**

Depende dos sistemas, em sistemas mais maduros pode até ser anualmente, em sistemas que ainda estão a ser desenvolvidos ou alterados, pode ocorrer semanalmente ou mensalmente.

**Q7 - Qual o impacto destes erros em ambiente produtivo?**

Dependendo dos erros, podem ter maior ou menor impacto, mas são poucas as situações em que tem sido necessário efetuar *hotfixes* para corrigir um problema impactante em produção.

**Q8 - Quanto tempo demora em média para ser corrigido?**

Se for uma correção que possa ir no *sprint* seguinte demora em média 3 semanas, caso seja algo urgente por norma é no próprio dia. Essa avaliação é feita pela EPR da aplicação.

**Q9 - Durante o levantamento de requisitos são identificados testes de aceitação que o Sistema de Informação deve passar?**

Infelizmente não.

**Q10 - Em que fase do desenvolvimento são feitos testes?**

No Scrum o desenvolvimento é dividido em períodos de 3 a 4 semanas e após esse período o software passa para o ambiente de qualidade onde a EPR efetua os testes. Durante o desenvolvimento propriamente dito, os testes são efetuados pelos programadores que estão a desenvolver as funcionalidades, mas serão testes básicos.

Apenas nas aplicações em PHP e Visual Basic, que têm o seu código alojado no *GitLab* e o *deploy* das mesmas é efetuado nessa ferramenta, temos criados *pipelines*, que para além de permitirem fazer *builds* e *deploys* de forma automática, permitem efetuar testes elementares, mas apenas de análise estática do código.



**Q11 - Antes de entrada em produtivo, os últimos testes são efetuados por quem? Considera que se estes fossem feitos de forma autónoma seria vantajoso?**

Antes de entrar em produção os testes são efetuados em qualidade pelas respetivas EPR e de uma forma manual. Penso que existia toda a vantagem em testes automáticos, principalmente em aplicações mais complexas em que o impacto de uma alteração pode não ser testado de forma correta em todos os pontos que possam sofrer impacto. Também se os testes fossem automáticos o tempo que uma aplicação demora em qualidade antes de transitar para produção seria muito inferior.

**Q12 - Existem alguma *checklist* de testes a efetuar antes da entrada de cada versão em produtivo? Considera que se estes fossem feitos de forma autónoma seria vantajoso?**

A existir uma *checklist* será do lado das EPR. Mas mais uma vez considero que seria vantajoso que os testes fossem automáticos, quer pela fiabilidade quer pela rapidez.

**Q13 - Quais as linguagens de desenvolvimento usadas?**

Outsystems, PHP, JavaScript, HTML, Visual Basic, Cobol, C# e PL/SQL.

**Q14 - Quais as ferramentas/programas utilizados para a automatização de testes?**

O único que usamos atualmente, e tal como referi na resposta à questão 10, é o *GitLab*.

## **Guião de entrevista para o Engenheiro Ricardo Costa, *Lead Architect* nos *Professional Services* da OutSystems e respetivas respostas**

**Q2 - Seguem alguma metodologia de desenvolvimento de *software*? Se sim, qual e desde quando?**

No que toca a metodologias é difícil seguir apenas uma. Sendo assim, atualmente segue-se muito *DevOps*, *Agile*, *CI/CD*, *Domain Driven Design (DDD)*, *Behavior Driven Development (BDD)*, *Test Driven Development (TDD)*.

Estas metodologias estão também muito interligadas. Há quem defenda que *DevOps* é *Agile* levado ao extremo. O que quer que isso signifique. Há quem também defenda, por exemplo, que  $BDD = DDD + TDD$ , mas esta fórmula será sempre algo redutora para abranger tudo aquilo que representa o BDD para equipas de entrega de *software*.

Portanto e para simplificar vamos assumir que as metodologias/práticas mais importantes usadas são *Agile*, *DDD*, *BDD* e *CI/CD*.

**Q3 - Durante o ciclo de desenvolvimento/sustentação são efetuados algum tipo de testes estruturados manuais ou automáticos?**

Durante o ciclo de desenvolvimento são seguidas as práticas de BDD. Sendo assim tudo é feito a pensar em especificações executáveis. Estas especificações executáveis podem ser de alto nível, quando os cenários partem de *business goals/end user features*, e de baixo nível, partindo de testes unitários, mas respeitando a sintaxe *Gherkin*.

A decisão de automatizar ou não o critério de aceitação das *User Stories* é feito nas sessões de 3 amigos (QA, Business e Dev). Optando por automatizar, podemos depois automatizar em termos de *backend*/lógica de negócio ou em termos de UI (recorrendo a ferramentas baseadas em Selenium).

Tudo o que é automatizado contribui depois também para o *CI/CD*, em que o *deployment* é feito para um ambiente intermédio entre Dev e QA e se todos os testes passarem o *deployment* é automático para QA.

**Q9 - Durante o levantamento de requisitos são identificados testes de aceitação que o Sistema de Informação deve passar?**

Sim. O *Behavior Driven Development* gira muito a volta disso.

**Q10 - Em que fase do desenvolvimento são feitos testes?**

O mais cedo possível. Não é seguida uma abordagem purista, desenvolvimento do teste, o teste falha, desenvolvimento do código, o teste passa e repete, mas é feito o chamado *Shift Left* em termos de testes. O que significa que as *User Stories* não cumprem o *Definition of Ready* se não estiverem desenvolvidos os testes automáticos. Os testes automáticos, como referido anteriormente, são acordados nas sessões de 3 amigos. Sessões estas que antecedem a inclusão das *User Stories* nos *sprints* de desenvolvimento.

**Q11 - Antes de entrada em produtivo, os últimos testes são efetuados por quem? Considera que se estes fossem feitos de forma autónoma seria vantajoso? Porquê?**

Os “últimos” testes não é algo propriamente aplicável no contexto em que atualmente se trabalha. Todos os testes foram documentados e automatizados e correm nos momentos definidos, muitos deles diariamente e inclusive com geração de *reports*.

É sempre vantajoso fazer o tal *Shift Left* em termos de *testing*. Quanto mais cedo melhor. Menos *rework* haverá como consequência de falhas por falta de testes, menos replaneamento, menor impacto nas várias equipas que participam em todas as fases do ciclo de vida de desenvolvimento de uma Aplicação/Produto.



Até mesmo em termos daquilo que são as expectativas do Negócio. Por exemplo, se algo se “descobre” já muito próximo de um *Go Live*, pode comprometer a data planeada para o mesmo. E até mesmo no dia a dia, quando fazendo uso de práticas CI/CD, em que estas só são possíveis se houver grande confiança que vamos apanhar de forma atempada alterações que partam certas funcionalidades.

**Q12 - Existem alguma *checklist* de testes a efetuar antes da entrada de cada versão em produtivo? Considera que se estes fossem feitos de forma autónoma seria vantajoso? Porquê?**

A *checklist* foi construída durante os ciclos de *grooming* das *User Stories* e durante os *sprints* de desenvolvimento. Automatizando tudo o que sejam testes a funcionalidades cujo impacto tenha sido identificado como elevado será sempre vantajoso.

Tudo o que é manual está mais sujeito a falha e é mais difícil de repetir principalmente quando as cadências de entrega são elevadas, por exemplo diariamente.

**Q13 - Quais as linguagens de desenvolvimento usadas?**

OutSystems, .Net, Javascript

**Q14 - Quais as ferramentas/programas utilizados para a automatização de testes?**

BDD Framework (OutSystems) Server and Client versions, Selenium, Appium, UI Path (UI Testing)

**Q15 - Como caracteriza o processo de definição de testes automáticos a implementar num Sistema de Informação?**

A minha sugestão é seguir o que está definido em termos de *Behavior Driven Development*

**Q16 - Têm testes de Unidade automáticos? Se sim, em que situações são utilizados?**

Sim, mas sempre orientados a *behavior*. Ou seja, especificação executáveis de *low level*. Usados em testes de funcionalidades de negócio, teste de certos funcionamentos de APIs (quando expostas), etc.

**Q17 - Têm testes Funcionais automáticos? Se sim, em que situações são utilizados?**

Cada caso é um caso, mas temos sim. Lá está respeitando o que for decidido nas sessões de 3 amigos e o que a boa prática definida pelo BDD.

**Q18 - Têm testes de Desempenho automáticos? Se sim, em que situações são utilizados?**

Tipicamente chamamos a estes testes, testes de carga. Feitos em ambiente de Pré-Produção porque é o ambiente mais semelhante a Produção. São utilizados maioritariamente em situações pré *Go Live*. E até podem servir para justificar decisões de *No Go*. Decisões onde o *Go Live* é adiado porque o desempenho não é o aceitável.

**Q19 - Têm testes de Ponta-a-Ponta automáticos? Se sim, em que situações são utilizados?**

Depende do que caracterizamos como Ponta-a-Ponta (ou em Inglês *End-to-End*). Mas geralmente não temos.

**Q20 - Em que situações optam por testes automáticos em vez de testes manuais? Existe alguma situação em que são usados complementarmente?**

Quando as funcionalidades têm um impacto considerado pelos 3 amigos alto.

**Q21 - É necessária ou aconselhada formação específica para a implementação de testes automáticos?**

Sim. Dependendo das equipas nomeadas para entregar um determinado Projeto/Produto essa formação pode ser interna ou contratada.

**Q22 - É necessária ou aconselhada formação específica para a definição de testes de aceitação?**

Sim. Formação nas práticas BDD é um *must have*.

**Q23 - Qual o nível de experiência dos elementos que definem e que implementam os testes automáticos?**

Depende do *budget*.

**Q24 - Quanto tempo demora a um elemento ficar proficiente?**

Depende dos Projetos em que esse elemento já participou. Não é algo que eu consiga quantificar.

**Q25 - As definições de testes automáticos são feitas pelas equipas de desenvolvimento ou por equipas à parte dedicadas?**

Em conjunto pelos 3 amigos, QA + DEV + Negócio

**Q26 - Com que frequência e em que situações é repetida a execução dos testes automáticos?**

Muitos são diariamente, outros só antes do *Go Live* inicial ou antes de *Go Live* de funcionalidades mais abrangentes.

**Q27 - Há quanto tempo implementaram testes automáticos no processo de desenvolvimento de Sistemas de Informação?**

No Ecossistema OutSystems a prática de forma generalizada é relativamente recente, uns 3 anos. Embora já muitos a seguissem muito antes disso.



**Q28 - Desde a implementação de testes automáticos houve um acréscimo ou uma redução do tempo de indisponibilidade dos Sistemas de Informação?**

Um decréscimo claro. E a própria entrega de novas funcionalidades e alterações a funcionalidades existentes é feita de forma mais célere e com um maior grau de confiança.

**Q29 - Qual a média de tempo de indisponibilidade dos Sistemas de Informação?**

Não sei dizer.

**Q30 - Como avalia a qualidade dos Sistemas de Informação após a implementação de testes automáticos?**

Aumenta substancialmente. E o próprio código desenvolvido fica muito mais fácil de manter.

**Q31 - Para avaliar a escalabilidade de um Sistema de Informação são efetuados alguns testes em específico?**

Testes de carga, como referido anteriormente.

**Q32 - Em que fase é mais importante o apoio das chefias durante o processo de implementação de testes automáticos?**

Neste contexto as equipas querem-se autónomas e o conceito de chefia é um pouco diferente. Mas o processo está sempre condicionado a *Budget* e a forma como o Projeto é vendido. *Budget* menor e prazos mais apertados normalmente irá levar a que testes fiquem por automatizar.

**Q33 - Considera que a complexidade do Sistema de Informação deve ser tida em conta para a decisão de implementação de testes automáticos ou não? Porquê?**

Claro. Poderá não fazer sentido investir esforço de desenvolvimento na automatização de testes sobre funcionalidades pouco complexas, ou seja, com pouco impacto nas funcionalidades auto nível que efetivamente acrescentar valor ao utilizador final.

A capacidade de automatização de testes durante cada sprint não é infinita. Tipicamente é de 30% em média e isso tem de ser tido em conta no que deve ser automatizado. Automatizando primeiro o que é mais prioritário/impactante.

**Q34 - Quais as diferenças, vantagens e desvantagens entre sistemas com testes automáticos e com testes manuais?**

Mais fáceis de manter, mais robustos, maior rapidez na deteção e correção de bugs.

**Q35 - Das ferramentas utilizadas para a implementação de testes automáticos, são utilizadas ferramentas com licenciamento específico ou gratuitas? Quais?**

Ambas. Já referidas anteriormente. *BDD Framework (OutSystems) Server and Client versions, Selenium, Appium, UI Path (UI Testing)*

**Q36 - Quais os riscos encontrados durante e após a implementação de testes automáticos? De que forma foram mitigados?**

O risco está mais ao nível do tempo para as entregas. Mas é uma troca. Ir um pouco mais devagar para garantir um futuro a médio/longo prazo mais calmo. Ou seja, a capacidade de desenvolvimento é um pouco mais reduzida para se garantir que as práticas de BDD são melhor seguidas.

**Q37 - Quais os maiores desafios durante e após a implementação de testes automáticos? Que medidas foram tomadas para os ultrapassar?**

Fazer o negócio perceber as vantagens de entregar menos, mas devidamente enquadrado com testes automáticos.

As medidas a tomar passam por demonstrar com factos/dados que efetivamente a automatização de testes tornam as aplicações/produtos mais fáceis de manter e mais robustos e ajudam as Equipas de desenvolvimento/manutenção a detetar e a corrigir bugs.

**Guião de entrevista para o Engenheiro Cristopher Pereira, Quality Automation Engineer na TalkDesk e respetivas respostas.**

**Q2 - Seguem alguma metodologia de desenvolvimento de software? Se sim, qual e desde quando?**

Sim seguimos. Aplicamos a metodologia AGILE. Tanto quanto tenho conhecimento, sempre foi utilizada esta metodologia desde a fundação da Talkdesk como empresa.

**Q3 - Durante o ciclo de desenvolvimento/sustentação são efetuados algum tipo de testes estruturados manuais ou automáticos?**

O processo de testes começa tão ou mais cedo que o processo de desenvolvimento. Cada equipa AGILE tem um QA alocado que fica responsável por definir o plano de testes, bem como desenhar e executar os testes que considere relevantes. Todo este processo é liderado pelo QA da equipa, que envolverá os restantes elementos da equipa sempre que for pertinente.



**Q9 - Durante o levantamento de requisitos são identificados testes de aceitação que o Sistema de Informação deve passar?**

Para melhor expor este processo, é fundamental perceber antes como se desenrola o ciclo de desenvolvimento na Talkdesk.

Por ano fiscal são consideradas 4 *major releases*, *Spring*, *Summer*, *Fall* e *Winter*. Para cada *release* o *Product Owner* da equipa, com base nas reuniões com os clientes e restantes POs, determina quais os Épicos e/ou *User Stories* a incluir na *release*. Este trabalho, define então qual objetivo de entrega de produto para a *release*.

Depois de estar identificado o que se pretende entregar na *release*, a equipa começa a trabalhar no refinamento de cada épico, de onde saem as tarefas e sub-tarefas a executar, com as devidas descrições do “What”, “Why”, “How” e claro os critérios de aceitação.

**Q10 - Em que fase do desenvolvimento são feitos testes?**

Os testes iniciam-se tão cedo quanto possível. Numa primeira abordagem, são criados testes unitários pelos próprios programadores, que poderão ser ou não acompanhados por um QA no seu desenvolvimento. Paralelamente dá-se início à avaliação da pertinência da inclusão na *framework* de automação de testes e se necessário a criação do mesmo. Por fim, após a entrega da tarefa pelo programador, o QA executa os testes definidos no plano de testes.

**Q11 - Antes de entrada em produtivo, os últimos testes são efetuados por quem? Considera que se estes fossem feitos de forma autónoma seria vantajoso? Porquê?**

Os últimos testes são sempre realizados pelo QA da equipa, assim como é o QA que realiza os *deploys* para o ambiente de produção. Quanto à questão da automação, é fundamental perceber que a mesma não é o “Santo Gral” da garantia de qualidade de um software. É uma forma de realizar testes útil, principalmente na execução de testes de regressão e de *smoke tests*, uma vez que estes testes são realizados com muita frequência e acabam por ser tarefas rotineiras. No entanto, nem todos estes testes são suscetíveis de serem automatizados. De forma a perceber quais os testes mais importantes que efetivamente deverão ser priorizados no processo de automação, é necessário responder a uma questão à seguinte questão: “Qual a coisa mais importante que o nosso sistema faz e o que deveremos testar?”. Sendo esta uma questão bastante lata, a mesma poderá ser refinada em questões mais concretas:

Quais são os utilizadores principais do sistema? Qual o valor acrescentado que este sistema traz aos utilizadores? Quais os *outputs* que produzem esse valor? Se um desses *outputs* falhar, ainda que momentaneamente em produtivo e que sem ninguém para além da equipa de desenvolvimento/suporte perceber, qual deles criará mais danos? Que utilizador é mais provável que possa igualmente detetar a falha, quando e como? Qual foi a característica mais importante que deu início ao produto? O que é que os utilizadores faziam antes da existência deste novo sistema?

Estas são apenas algumas questões a ter em mente quando se pretende definir quais os primeiros testes a automatizar.

Ainda assim, sim, ter uma *framework* de automação de testes é vantajoso, pois permite reduzir o tempo da execução de testes rotineiros com tarefas repetitivas, libertando assim quem executa os testes para realizar outras tarefas.

**Q12 - Existem alguma *checklist* de testes a efetuar antes da entrada de cada versão em produtivo? Considera que se estes fossem feitos de forma autónoma seria vantajoso? Porquê?**

Não existe nenhuma *checklist* de testes a efetuar. Cada equipa tem o seu processo de QA e *deploy* para produtivo muito bem documentado, ainda que este mesmo processo seja na sua grande maioria igual a todas as equipas.

Salvo raras exceções todos os testes a realizar estão automatizados. A nossa *framework* de testes incluem a execução de testes de regressão, *smoke tests*, testes de integração e testes *end-to-end*. Considero vantajoso na medida em que permite verificar se a nova funcionalidade introduzida no sistema não teve qualquer impacto no correto funcionamento do sistema.

**Q13 - Quais as linguagens de desenvolvimento usadas?**

As linguagens de desenvolvimento utilizadas na Talkdesk diferem bastante. Utilizamos desde Java, a Python, C#, R, entre outras.

**Q14 - Quais as ferramentas/programas utilizados para a automatização de testes?**

Para o desenvolvimento da *framework* de automação de testes, utilizamos *Java*, *Selenium*, *Selenide*, *REST Assured* e *Cucumber*.

Utilizamos ainda o *SonarQube* para a inspeção contínua da qualidade do código.

**Q15 - Como caracteriza o processo de definição de testes automáticos a implementar num Sistema de Informação?**

Tal como referi anteriormente, o processo de definição dos testes automáticos a implementar exige uma extensa reflexão sobre o que deve ou não ser automatizado. E como tal acaba por ser ainda um processo



moroso, bem como é sempre preciso ter em mente que uma *framework* de automação exige quase tanta manutenção quanto o sistema de informação que a mesma testa.

**Q16 - Têm testes de Unidade automáticos? Se sim, em que situações são utilizados?**

Sim temos. São geralmente executados pelos programadores no ambiente local e posteriormente quando é feita a criação de um novo *Pull Request* da nova funcionalidade no repositório de código.

**Q17 - Têm testes Funcionais automáticos? Se sim, em que situações são utilizados?**

Sim temos. São executados em diversas ocasiões. Durante a execução de testes a uma nova funcionalidade, nas diversas fases de *deploy* para os ambientes de *Staging*, Qualidade e Produtivo, e na execução uma rotina diária que é executada automaticamente todas as noites.

**Q18 - Têm testes de Desempenho automáticos? Se sim, em que situações são utilizados?**

Não temos.

**Q19 - Têm testes de Ponta-a-Ponta automáticos? Se sim, em que situações são utilizados?**

Sim temos. São executados em diversas ocasiões. Durante a execução de testes a uma nova funcionalidade, nas diversas fases de *deploy* para os ambientes de *Staging*, Qualidade e Produtivo, e na execução uma rotina diária que é executada automaticamente todas as noites.

**Q20 - Em que situações optam por testes automáticos em vez de testes manuais? Existe alguma situação em que são usados complementarmente?**

Geralmente um dos critérios utilizados para a criação de testes automáticos é que o teste a criar teste no mínimo a integração de dois micro serviços. Se formos testar apenas um componente do sistema, verificamos sempre é possível integrar o teste nos testes unitários. Sim existem, geralmente quando uma nova funcionalidade é adicionada ao sistema.

**Q21 - É necessária ou aconselhada formação específica para a implementação de testes automáticos?**

Acima de tudo é necessário que exista uma cultura para a Qualidade na entidade/empresa na qual está a ser desenvolvido o produto. É fundamental entender que incluir o processo de garantia de qualidade, do qual fazem parte os testes, é o que permitirá reduzir custos com a manutenção de um sistema, antecipar e evitar a entrada de problemas no ambiente de produtivo, etc.

Formação será sempre importante, pois permite-nos estar mais familiarizados com quais as boas práticas da indústria. Existe igualmente formação específica para a área de automação de testes de *software*, que novamente, é importante pois permite dar a conhecer quais as boas práticas da indústria. Mas, quanto a esta última, ela apenas será útil para uma empresa/entidade que não tenha rigorosamente nada automatizado nem ninguém com a experiência em criar uma *framework* de automação.

**Q22 - É necessária ou aconselhada formação específica para a definição de testes de aceitação?**

Não considero que seja necessária formação específica. Considero novamente que é preciso sim, uma cultura para o Controlo de Qualidade e ter uma boa equipa de QA dedicada exclusivamente ao processo.

**Q23 - Qual o nível de experiência dos elementos que definem e que implementam os testes automáticos?**

Uma *framework* de testes automáticos exige uma constante manutenção e sustentação, pelo que a complexidade deste processo estará em muito ligado à experiência dos elementos da equipa responsável pela mesma. No entanto, diria que deverá haver pelo menos um programador experiente na equipa.

**Q24 - Quanto tempo demora a um elemento ficar proficiente?**

Pessoalmente, considero que o tempo que um elemento demorará a ficar proficiente na implementação de testes está intrinsecamente ligado à experiência que o mesmo tem em desenvolvimento de *software*. Quanto mais experiente for, mais rapidamente ficará proficiente.

**Q25 - A definição de testes automáticos são feitas pelas equipas de desenvolvimento ou por equipas à parte dedicadas?**

Na Talkdesk, a definição dos testes a automatizar é da inteira responsabilidade do QA. O mesmo poderá/deverá trabalhar em conjunto com a equipa de desenvolvimento, mas a responsabilidade da sua definição e implementação será do QA.

**Q26 - Com que frequência e em que situações é repetida a execução dos testes automáticos?**

Os testes automáticos são executados em diversas ocasiões. Diariamente num *job* que é executado automaticamente todas as noites, para garantir que não terá ido nada para o ambiente de produtivo erradamente. Sempre que são efetuados *deploys* para os ambientes de *Staging*, Qualidade e Produtivo. E por fim, sempre que o QA entender na execução do seu trabalho.

**Q27 - Há quanto tempo implementaram testes automáticos no processo de desenvolvimento de Sistemas de Informação?**

No produto no qual estou envolvido, desde o início de vida do mesmo. Sei, no entanto, da existência de outros produtos na empresa que não tem uma *framework* de testes automáticos.



**Q28 - Desde a implementação de testes automáticos houve um acréscimo ou uma redução do tempo de indisponibilidade dos Sistemas de Informação?**

Não tenho informação suficiente para responder a esta questão. Mas desde que estou na empresa posso afirmar que nunca houve indisponibilidade do Sistema de Informação por entrada de um *bug* para o ambiente de Produtivo. Mas aí entra uma boa estrutura de CI/CD implementada na empresa. Das poucas vezes que foi feito um *deploy* para Produtivo de um defeito, rapidamente o mesmo foi detetado e revertida versão do sistema para a versão anteriormente em Produtivo.

**Q29 - Qual a média de tempo de indisponibilidade dos Sistemas de Informação?**

Com base no tempo que estou na empresa a média será de 0 minutos. Mas há aqui um aspeto fundamental a referir, a empresa é uma empresa que vende um serviço inteiramente disponibilizado na *Cloud* e com um SLA bastante exigente, pelo que todos os processos envolvidos no desenvolvimento e disponibilização dos Sistemas de Informação vão ao encontro de que os mesmos tenham o mínimo de tempo possível de indisponibilidade.

**Q30 - Como avalia a qualidade dos Sistemas de Informação após a implementação de testes automáticos?**

Qualquer Sistema de Informação que inclua um bom processo de Controlo de Qualidade no seu ciclo de vida, terá sempre uma boa qualidade. A introdução de testes automáticos permitiu a redução de tempo na execução de testes rotineiros, garantir que todos os micro serviços funcionam entre si e que quais falhas que sejam levadas até ao ambiente de Produtivo não passem despercebidas.

**Q31 - Para avaliar a escalabilidade de um Sistema de Informação são efetuados alguns testes em específico?**

Não. Este trabalho é feito diretamente pelas equipas de desenvolvimento e cada um utiliza os *benchmarks* que considerar convenientes.

**Q32 - Em que fase é mais importante o apoio das chefias durante o processo de implementação de testes automáticos?**

Desde o início do ciclo de vida de um Sistema de Informação. O terceiro princípio dos testes exposto no *Foundation Level* da certificação da ISTQB diz que “Testar cedo poupa tempo e dinheiro”. E de facto, para encontrar defeitos precocemente, os testes devem ser iniciados o mais cedo possível no ciclo de vida do desenvolvimento do *software*. Testar no início do ciclo de vida do desenvolvimento de *software* ajuda a reduzir ou eliminar alterações caras. Como tal, como a questão do Controlo de Qualidade tem uma vertente muito cultural da entidade/empresa, é fundamental que as chefias apoiem o processo de implementação de testes automáticos.

**Q33 - Considera que a complexidade do Sistema de Informação deve ser tida em conta para a decisão de implementação de testes automáticos ou não? Porquê?**

Sim, totalmente. Manter uma *framework* de automação de testes exige esforço e constante análise para fazer face às várias iterações e maturação do produto. Pelo que deverá ser considerado se se justifica tal esforço para projetos de menor complexidade.

**Q34 - Quais as diferenças, vantagens e desvantagens entre sistemas com testes automáticos e com testes manuais?**

As principais diferenças prendem-se com a necessidade de se empenhar tempo no desenvolvimento dos testes automáticos a executar. A vantagem é possibilitar a execução de forma mais rápida e eficiente de testes. Essencialmente a automatização dos testes poderá ser vista como uma forma de otimizar um processo numa realidade onde cada vez mais se pretende entregar produto a curto prazo e iterativamente. Uma desvantagem da automatização de testes é que se a *framework* de automação não for alvo de revisões e de manutenção, rapidamente poderão começar a surgir falsos negativos o que levará a um custo de tempo acrescido na análise dos testes que estão a falhar.

**Q35 - Das ferramentas utilizadas para a implementação de testes automáticos, são utilizadas ferramentas com licenciamento específico ou gratuitas? Quais?**

A maioria das ferramentas utilizadas na Talkdesk para o desenvolvimento de testes automáticos tem todas licenças gratuitas.

O *Selenium* encontra-se ao abrigo da licença *Apache License 2.0*, o *Selenide* ao abrigo da licença *MIT License*, o *REST Assured* encontra-se ao abrigo da licença *Apache License 2.0*.

Já o *Cucumber* como está integrado com o JIRA (ferramenta de monitorização de tarefas e gestão de projetos) necessita de licenciamento específico.

**Q36 - Quais os riscos encontrados durante e após a implementação de testes automáticos? De que forma foram mitigados?**

O maior risco é criar-se uma *framework* de tal forma complexa e mal mantida que o resultado dos testes comece a não acrescentar valor ao ciclo de desenvolvimento. A forma como se mitigou este risco foi através da constante avaliação de resultados dos testes e consequente análise dos testes que possam estar a originar



problemas. Apesar de cada equipa de desenvolvimento ter o seu QA atribuído, existe uma estrutura própria de QA da qual fazem parte todos os QA. Esta estrutura responde perante uma estrutura hierárquica própria da empresa e permite entre várias coisas partilhar conhecimento e dificuldades com os restantes QAs, assim como trabalhar em conjunto na manutenção da *framework* de testes automáticos.

**Q37 - Quais os maiores desafios durante e após a implementação de testes automáticos? Que medidas foram tomadas para os ultrapassar?**

O maior desafio da implementação de testes automáticos é perceber que testes automatizar. É fundamental perceber que áreas do Sistema de Informação se pretende testar de forma automática. No caso da Talkdesk definiu-se que para um teste ser passível de ser automatizado deverá pelo menos existir interação entre dois micro-serviços, o que facilita bastante na altura de triar o que deverá ser automatizado ou não. Depois, existe uma estrutura de QA da qual todos os QA fazem parte, onde é possível expor as principais dificuldades e trabalhar em conjunto para a obtenção de uma solução conveniente aos problemas expostos.