

Estágio em Quality Assurance: Desenvolvimento de Testes de Integração e Automação para Veículos BMW

Relatório Final do Projeto Integrador

Rafael Schmidt



Licenciatura em Engenharia Informática e Computação

Tutor na U.Porto: Bruno Miguel Carvalhido Lima
Orientador na empresa/Proponente: Pedro Marcelo Andrade Carneiro

21 de Abril de 2023

Conteúdo

1	Introdução	2
1.1	Enquadramento	2
1.2	Objetivos e resultados esperados	2
1.3	Estrutura do relatório	2
2	Metodologia utilizada e principais atividades desenvolvidas	3
2.1	Metodologia utilizada	3
2.2	Intervenientes, papéis e responsabilidades	5
2.3	Atividades desenvolvidas	5
2.3.1	T1: Onboarding	6
2.3.2	T2: Onboarding com a equipa	6
2.3.3	T3: Desenvolvimento da testsuite para MGU22	6
2.3.4	T4: Investigação sobre testes em Android Auto Open Source	6
3	Desenvolvimento dos testes	7
3.1	Requisitos	7
3.2	Arquitetura e tecnologias	7
3.3	Solução desenvolvida	7
3.4	Validação	8
4	Conclusões	8
4.1	Resultados alcançados	8
4.2	Lições aprendidas	9
4.3	Trabalho futuro	10

1 Introdução

1.1 Enquadramento

O presente relatório visa documentar o trabalho realizado durante o estágio do estudante Rafael no âmbito da unidade curricular de Projeto Integrador da Licenciatura em Engenharia Informática e Computação da Faculdade de Engenharia da Universidade do Porto.

O estágio ocorreu na empresa Critical Techworks[12]: uma joint venture entre o BMW Group[5] e a empresa portuguesa Critical Software[10]. A Critical Techworks trabalha no desenvolvimento de software para vários domínios da BMW, desde software para controle de linhas de produção em fábricas até software embarcado de condução autônoma.

1.2 Objetivos e resultados esperados

O objetivo do estágio consistiu no desenvolvimento de testes de integração automatizados no domínio de personalização para a Head Unit MGU22 (Media Graphics Unit 22) do sistema de infoentretenimento dos veículos. Tal módulo, que executa os sistemas operativos iDrive 7 e iDrive 8, é capaz de gerir perfis de utilizadores e gravar as suas definições, como a posição dos bancos, definições de temperatura e até interagir com a aplicação MyBMW[6] para controle remoto de funções do carro.

Testes automatizados de integração para software são desenvolvidos com o objetivo de garantir o correto funcionamento das diferentes partes do sistema de forma conjunta. Esses testes são essenciais para detectar problemas de compatibilidade entre os componentes do software e validar a integração entre eles, com o objetivo de identificar e corrigir possíveis problemas de integração de forma rápida e efetiva antes de o software ser instalado no veículos dos consumidores finais.

Para além do desenvolvimento de testes, foi também planeado o trabalho de investigação para a criação de um novo framework de testes para o iDrive 9, em uma nova Head Unit, baseada em Android OS.

Concretamente, no final do estágio, espera-se que o estagiário tenha adquirido experiência prática em testes de integração automatizados em um ambiente ágil utilizando a linguagem Python, bem como conhecimentos em metodologias ágeis de desenvolvimento de software, com destaque para a metodologia Scrum. Além disso, é esperado que sejam desenvolvidas soft-skills essenciais para a integração no mercado de trabalho, como trabalho em equipa, comunicação profissional, aprendizagem contínua e adaptabilidade.

1.3 Estrutura do relatório

O relatório está dividido, para além da secção atual, em outras três secções principais: Metodologia utilizada e principais atividades desenvolvidas, Desenvolvimento dos testes e Conclusões.

A seção 2 descreve o processo utilizado para alcançar os objetivos estabelecidos de maneira a fornecer detalhes suficientes para que o leitor possa entender claramente o processo pelo qual a solução foi alcançada.

Na seção 3 são apresentados os requisitos e tecnologias utilizados para desenvolver os testes durante o estágio. Além disso, expõe as soluções desenvolvidas e como foram validadas.

Finalmente, na seção 4, são descritas as principais conclusões do trabalho realizado. Nessa seção, são expostos os resultados obtidos, apresentando recomendações para trabalho futuro.

2 Metodologia utilizada e principais atividades desenvolvidas

2.1 Metodologia utilizada

Na Critical Techworks, é implementada uma cultura ágil, utilizando cerimônias e ferramentas do framework Scrum[9]. Nesse sentido, a empresa divide os seus funcionários em pequenos grupos, tipicamente variando de 5 a 8 profissionais, formando equipas. Estas equipas são autônomas, o que significa que tem poder de decidir internamente a dinâmica de sua organização e divisão de tarefas. Em uma visão mais alto nível, um conjunto de equipas que trabalha no mesmo domínio forma a uma Unidade e o conjunto de unidades compõe a empresa.

É difundido na empresa o lema “inspect and adapt”, que se traduz “inspecionar e adaptar”. O significado desse lema é aplicado na interpretação do Guia do Scrum[8], utilizando o framework como um kit de ferramentas a ser utilizado da maneira que as equipas julgarem ser mais eficiente para trazer melhorias ao produto em que trabalham.

A equipa que recebeu o estagiário durante o estágio, era composta por 6 elementos inicialmente. Nesta equipa, o trabalho foi desenvolvido iterativamente com sprints bissemanais. Diariamente, ocorreram reuniões chamadas Daily Meetings, onde cada participante da equipa falava sobre as tarefas realizadas no dia anterior e o que planeava fazer naquele dia. Para além disso, mais três cerimônias do Scrum realizadas: Sprint Review, na qual o trabalho planeado na sprint é revisitado e analisado durante uma reunião com toda a equipa, Sprint Retrospective, quando é discutido o desenvolvimento da sprint, o que correu bem e o que poderia ter corrido melhor, e por fim o Sprint Planning, que é a primeira cerimônia de uma sprint, onde são definidas e planeadas as tarefas que uma equipa se compromete a realizar no período de duas semanas.

Para o controle e desenvolvimento das tarefas, foram utilizadas ferramentas de entrega contínua como GitHub[7], Jenkins[11], JIRA[1] e Gerrit[4]. Os testes foram desenvolvidos utilizando a linguagem Python e JavaScript e o sistema operativo Linux, utilizando o método “black box”[2], que é uma técnica de teste de software em que o testador examina a funcionalidade de um sistema sem conhecer a sua estrutura interna. Desta forma, o testador avalia o sistema

com base nas entradas e saídas esperadas, sem ter acesso ao código-fonte ou à arquitetura subjacente. O objetivo é garantir que o sistema corresponde aos requisitos e as expectativas do utilizador final.

O fluxo de desenvolvimento dos testes decorreu de acordo com o diagrama da figura 1:

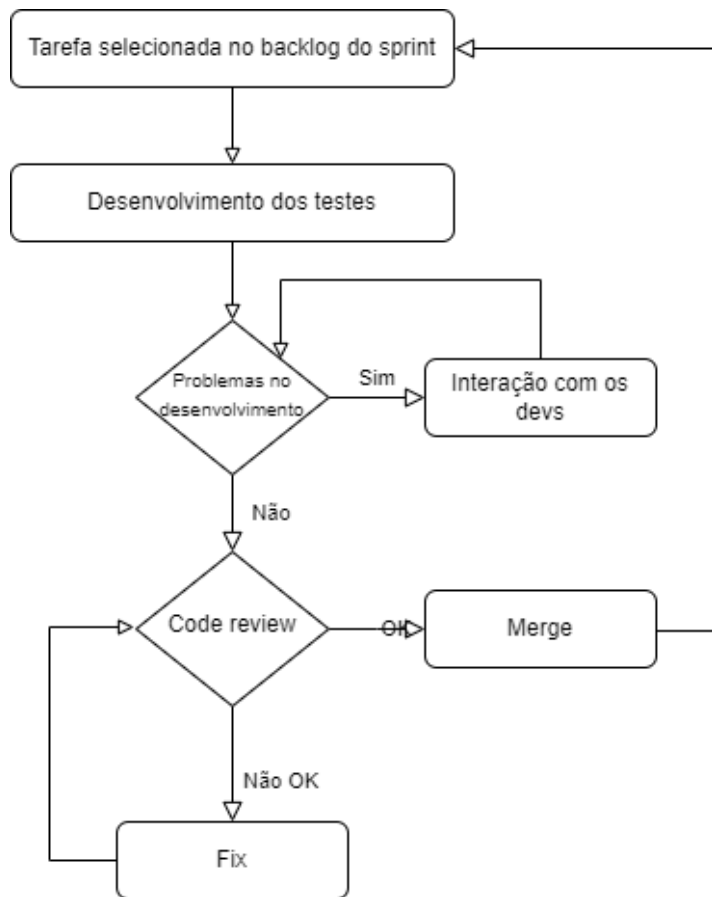


Figura 1: Fluxograma do desenvolvimento dos testes

Todas as equipas possuem uma “Definition of Done”, que é uma métrica definida e utilizada pelas equipas para haver um consenso sobre a completude de uma tarefa. No contexto do estágio, para considerar um teste como feito, era necessário a aprovação e a revisão do código por dois outros membros da equipa. Após isso, o código estava pronto para ser anexado ao repositório de código principal.

Durante o desenvolvimento dos testes, foi necessário interagir o mínimo possível com os desenvolvedores das funcionalidades, uma vez que os testes

elaborados seguiram o modelo “black box”, detalhado anteriormente.

Portanto, os desenvolvedores disponibilizavam os casos de testes, adicionando-os ao backlog e estes eram atribuídos aos testers da equipa que trabalhavam em sua implementação.

2.2 Intervenientes, papéis e responsabilidades

Na equipa que o estagiário integrou, foram respeitados os papéis tradicionais de uma equipa Scrum, sendo eles:

- Product Owner[3] (Marta Segadães): O Product Owner é responsável por definir a visão do produto e priorizar o backlog do produto. Ele trabalha em estreita colaboração com a equipa de desenvolvimento para garantir que o produto corresponde às necessidades dos utilizadores e do negócio.
- Scrum Master (Marcos Silva): O Scrum Master é responsável por garantir que a equipa Scrum cumpre as práticas do Scrum e resolve impedimentos que possam surgir. É um facilitador que ajuda a equipa a trabalhar de forma mais eficiente e eficaz. No caso da equipa que o estagiário integrou, o Scrum Master atuou também como desenvolvedor, selecionando e implementando, além de ser responsável por garantir a prática do Scrum.
- Equipa de Desenvolvimento de testes (Marcos Silva, Pedro Carneiro, Pedro Rocha, Radhika Moran e Rafael Schmidt): A equipa de desenvolvimento é composta por profissionais que trabalham juntos para entregar um produto de alta qualidade. Eles são responsáveis por implementar tarefas e cumprir a “Definition of Done”. A equipa de desenvolvimento é auto-organizada e multidisciplinar, o que significa que ela tem todas as competências necessárias para concluir as tarefas. Na equipa do estagiário, o desenvolvimento esteve voltado para testes automatizados de integração.
- Equipa de desenvolvimento de funcionalidades (Mohamed Garvi, Francisco Arrabaça e João Chambel): Possuem as mesmas características da equipa de desenvolvimento de testes, mas seu foco está em desenvolver novas funcionalidades para o sistema de personalização da central de fomentamento de veículos BMW.

O orientador do estágio foi o Pedro Carneiro, membro da equipa de desenvolvimento de testes. Para controlar o progresso do estagiário, foram utilizadas as Daily Meetings anteriormente citadas, com a finalidade de auxiliar, acompanhar o desenvolvimento das tarefas e planear as atividades de cada dia de trabalho.

2.3 Atividades desenvolvidas

As atividades desenvolvidas estão ilustradas pelo diagrama de Gantt[14] da figura 2:

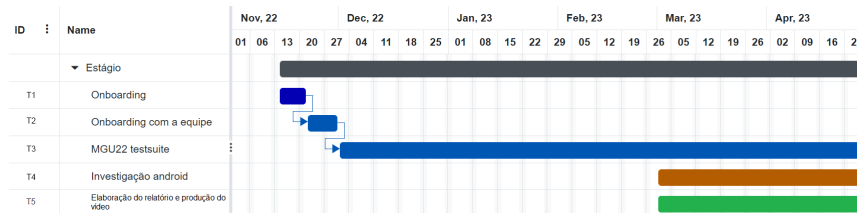


Figura 2: Diagrama de Gantt com as atividades desenvolvidas

2.3.1 T1: Onboarding

Durante a primeira semana de estágio, foram organizadas sessões com o intuito de apresentar aos novos membros da Critical os ideais da empresa, a dinâmica de trabalho e como as equipas são estruturadas. Houve também workshops com foco nas práticas do Scrum com um especialista, com dinâmicas de grupo para simular e elucidar sobre os papéis de cada membro de uma equipa. Esta etapa foi fundamental para o progresso no estágio, visto que através das dinâmicas de grupo foram resolvidos exemplos reais de problemas cotidianos de uma equipa de desenvolvimento de software.

2.3.2 T2: Onboarding com a equipa

Na semana subsequente, foi realizada a integração do estagiário com a equipa. Nesta altura, foram explicadas e instaladas as ferramentas para o desenvolvimento dos testes. Também foi solicitado o acesso às plataformas de versionamento e entrega de código. Durante esta fase, foram feitas sessões com os “Product Owners” para adquirir contexto sobre o produto testado. Durante esta fase foram ainda realizadas também dinâmicas para conhecer os membros da equipa, suas responsabilidades e desenvolver alguma intimidade com estes.

2.3.3 T3: Desenvolvimento da testsuite para MGU22

Neste período, foi feita a portabilidade de testes da Head Unit anterior para a uma versão mais recente. As tarefas incluíram analisar os casos de teste, no contexto da MGU21, estimar o esforço para realizá-los, adaptar para a MGU22 e desenvolver potenciais funções auxiliares para incrementar a usabilidade da framework de testes para casos futuros. Foi utilizado o método “black box testing”, sem conhecimento sobre como a implementação das funcionalidades foi feita, apenas seu comportamento esperado.

2.3.4 T4: Investigação sobre testes em Android Auto Open Source

Estudo para a fase de cobertura inicial do novo sistema de infotainment feito em Android Auto Open Source. Incluiu a construção inicial de um framework de testes, analisar os requisitos, planear e validar os casos de teste e estimar o

esforço para sua implementação. Durante esta fase foram feitas reuniões com outras equipas que já trabalhavam para desenvolver software para este sistema operativo e houve uma intensa troca de informações sobre as metodologias e as tecnologias utilizadas.

3 Desenvolvimento dos testes

3.1 Requisitos

Os requisitos funcionais descrevem as funções ou recursos que o sistema deve ter para atender aos objetivos do utilizador final. No contexto dos testes de integração, é importante testar se as funcionalidades de cada componente individual do sistema funcionam corretamente e se as interações entre esses componentes ocorrem de maneira adequada.

Por outro lado, os requisitos não funcionais concentram-se em como o sistema se comporta em relação a fatores como desempenho, segurança, usabilidade e escalabilidade. Esses requisitos são igualmente importantes nos testes de integração, já que o desempenho e a segurança da interação entre os diferentes componentes podem ser afetados por esses fatores.

3.2 Arquitetura e tecnologias

Os testes de integração foram implementados na linguagem de programação Python. Foi criado um framework de testes que é capaz de interagir, enviando entradas e capturando as saídas da unidade central dos veículos. Os testes são scripts, divididos por etapas, em que cada etapa consiste em uma descrição e uma ação. Para verificar o comportamento do sistema, são realizadas asserções no código, que é interrompido caso estas não sejam as esperadas.

Para simular respostas de serviços externos, foram utilizados Mocks[13] desenvolvidos na linguagem JavaScript. Os Mocks são criados com o objetivo de fornecer um comportamento simulado que imita o comportamento de um objeto real, mas sem a complexidade do objeto real, permitindo que os desenvolvedores testem outras partes do sistema sem depender desses objetos.

3.3 Solução desenvolvida

A quando da integração do estagiário, a testsuite contava com 26 testes de integração. No final do estágio, contando com o esforço coletivo da equipa, foram implementados 48 novos testes, sendo 26 da responsabilidade do estagiário, totalizando 76 testes em abril de 2023. Os testes cobrem funcionalidades básicas, como a criação de utilizadores, até casos mais complexos, como a comunicação com o back-end e “testes de stress”, que consistem em realizar a mesma tarefa diversas vezes e observar o comportamento das interfaces.

Além disso, durante os sprints, foram desenvolvidas tarefas de melhoria do framework, como o download automático de imagens com novas funcionalidades,

melhorias na documentação dos casos de teste e escrita de documentação para futuros membros da equipa.

As soluções desenvolvidas demonstraram eficácia na detecção de problemas e contribuíram para a melhoria do processo de desenvolvimento de software, aumentando a confiabilidade do sistema e reduzindo riscos.

No âmbito da investigação para o desenvolvimento do framework baseado em Android, foi criada uma documentação para as futuras etapas de seu desenvolvimento, com todo o conhecimento partilhado por outras equipas através de reuniões. Foi também desenvolvido uma versão inicial do framework, que faz interações básicas com o software. Entretanto, não foi possível avançar mais em seu desenvolvimento devido à saída de um dos membros mais experientes da equipa.

3.4 Validação

Durante os seis meses de estágio, a testsuite foi corrida diversas vezes. Na maioria das situações, não foram detectados bugs nas versões testadas. Entretanto, uma única vez ocorreu um comportamento inesperado e, após uma reunião com os desenvolvedores, foi concluído que se tratava de um bug e foi transformado em um ticket para ser corrigido.

Apesar de parecer um baixo número de erros detetado, é importante perceber que a camada de testes de integração é a última a ser executada quando uma funcionalidade é implementada. A política da empresa consiste também no desenvolvimento de testes unitários, que filtram a maioria dos potenciais erros no código.

A equipa de testes trabalhou em conjunto com os desenvolvedores em alguns momentos para estudar novos casos de testes e trocar sugestões e feedbacks e, diversas vezes, o desenvolvimento dos testes obteve feedback positivo.

4 Conclusões

4.1 Resultados alcançados

Durante o estágio, foram alcançados resultados significativos em relação aos objetivos estabelecidos. O estagiário adquiriu experiência prática em testes de integração no modelo “black box” automatizados, utilizando a linguagem Python. Além disso, foram adquiridos conhecimentos em metodologias ágeis de desenvolvimento de software, com destaque para a metodologia Scrum. Os testes de integração foram realizados de forma eficiente e eficaz, contribuindo para a identificação de falhas e garantindo a qualidade do software. Durante o período de estágio, o estagiário também desenvolveu soft-skills essenciais, como trabalho em equipa, comunicação profissional, aprendizagem contínua e adaptabilidade, que são fundamentais para a integração bem-sucedida no mercado de trabalho. Os resultados alcançados demonstram o crescimento profissional e a capacidade do estagiário em aplicar os conhecimentos adquiridos em um am-

biente de trabalho real, contribuindo para o sucesso dos projetos em que esteve envolvido.

Na tabela 1 está ilustrada a contribuição do estudante em termos quantitativo para a base de testes. É possível observar que o estagiário foi responsável pelo desenvolvimento de mais de 1/3 dos testes finais.

Data	Total de testes	Testes feitos por Rafael Schmidt	% do total
06/12/2022	26	0	0
26/04/2023	74	26	35

Tabela 1: Amostragem dos testes desenvolvidos entre Dezembro e Abril

4.2 Lições aprendidas

Diversas lições foram aprendidas, contribuindo para o desenvolvimento profissional do estagiário. Uma das principais lições foi a importância dos testes de integração no processo de desenvolvimento de software. Foi possível compreender como os testes automatizados podem identificar problemas de integração entre os componentes do sistema, garantindo a estabilidade e a funcionalidade adequada. Além disso, a experiência com o Scrum, permitiu compreender a importância da colaboração e comunicação eficaz em equipes de desenvolvimento. O estagiário aprendeu também sobre a importância da adaptabilidade e da aprendizagem contínua, pois o ambiente de trabalho exige constante atualização e a capacidade de se adaptar a novas situações e desafios. Essas lições foram valiosas para o crescimento profissional e serão levadas como lições aprendidas para futuras oportunidades no mercado de trabalho.

Outra lição aprendida durante o estágio foi a importância da documentação de código e de processos. Ficou claro que a documentação adequada do código fonte é fundamental para facilitar a compreensão e manutenção do software. Além disso, a documentação de processos também se mostrou essencial para garantir a consistência das atividades desenvolvidas. Ao documentar os procedimentos e fluxos de trabalho, é possível assegurar a eficiência, qualidade e padronização das tarefas realizadas, além de permitir a fácil integração de novos membros nas equipes.

Por fim, a lição mais importante foi o valor dos testes no mercado de trabalho. Através dos testes, é possível identificar e corrigir falhas antes que o software seja instalado nos veículos de utilizadores, evitando custos e impactos negativos para as organizações. Um estudante, durante o seu curso acadêmico, pode não dar tanto valor aos testes de software, focando principalmente no desenvolvimento das funcionalidades. No entanto, ao ingressar no mercado de trabalho, percebe-se rapidamente a importância dos testes. Eles não garantem apenas a qualidade do produto final, mas também proporcionam confiança ao cliente, fortalecem a reputação da empresa e contribuem para a redução de riscos e problemas futuros. Assim, valorizar os testes de software como uma parte integral do processo de desenvolvimento é essencial para se destacar e ter sucesso profissionalmente.

4.3 Trabalho futuro

O trabalho futuro da equipa será de investigação para o desenvolvimento de um novo framework de testes para as centrais de infotainment que serão disponibilizadas no futuro com base em Android Auto. Outro aspecto importante é a continuidade da documentação, tanto do código como dos processos, a fim de garantir a sua manutenção e facilidade de compreensão para a equipa. Por fim, a participação em cursos e formações relacionados com testes de integração, metodologias ágeis e novas tecnologias pode ampliar ainda mais o conhecimento e as habilidades necessárias para futuros desafios.

Referências

- [1] Atlassian. Jira. <https://www.atlassian.com/software/jira>, 2023. [Online; acedido 21 de abril de 2023].
- [2] Boris Beizer. *Black-box testing: techniques for functional testing of software and systems*. John Wiley & Sons, Inc., 1995.
- [3] Gary R Gemmill and David L Wilemon. The product manager as an influence agent. *Journal of Marketing*, 36(1):26–30, 1972.
- [4] Google. Gerrit. <https://www.gerritcodereview.com/>, 2023. [Online; acedido 21 de abril de 2023].
- [5] BMW Group. Bmw group. <https://www.bmwgroup.com/en.html>, 2023. [Online; acedido 30 de março de 2023].
- [6] BMW Group. My bmw. https://play.google.com/store/apps/details?id=de.bmw.connected.mobile20.row&hl=pt_PT, 2023. [Online; acedido 30 de março de 2023].
- [7] GitHub Inc. Github. <https://github.com/>, 2023. [Online; acedido 21 de abril de 2023].
- [8] Jeff Sutherland Ken Schwaber. Scrum guide. <https://scrumguides.org/scrum-guide.html>, 2020. [Online; acedido 30 de maio de 2023].
- [9] Ken Schwaber. Scrum development process. In *Business Object Design and Implementation: OOPSLA '95 Workshop Proceedings 16 October 1995, Austin, Texas*, pages 117–134. Springer, 1997.
- [10] Critical Software. Critical software. <https://criticalsoftware.com/>, 2023. [Online; acedido 30 de março de 2023].
- [11] Open source. Jenkins. <https://www.jenkins.io/>, 2023. [Online; acedido 21 de abril de 2023].

- [12] Critical Techworks. Critical techworks. <https://www.criticaltechworks.com/>, 2023. [Online; acedido 30 de março de 2023].
- [13] Dave Thomas and Andy Hunt. Mock objects. *IEEE Software*, 19(3):22–24, 2002.
- [14] James M Wilson. Gantt charts: A centenary appreciation. *European Journal of Operational Research*, 149(2):430–437, 2003.