

# Os contratos de desenvolvimento e manutenção de *software*: o caráter duradouro e as influências do *scrum* sobre a tipologia, a cessação e a responsabilidade contratuais

## Vinicius Aquini Gonçalves

Associate Expert do Office of Administration of Justice, Nações Unidas. Doutorando em Direito Econômico e da Tecnologia (Universidade de Coimbra), LL.M. em Direito Internacional e Europeu Econômico (Universidade de Munique), graduado em Direito (UFRGS) e em Administração de Empresas com ênfase em Informática (Universidade Técnica de Munique). [viniciusaquini@gmail.com](mailto:viniciusaquini@gmail.com)  
ORCID: <https://orcid.org/0000-0001-8266-4975>

---

### SUMÁRIO

Introdução

1. O caráter duradouro dos contratos de desenvolvimento e manutenção de *software*
2. *Frameworks* de gerenciamento e organização de projetos de desenvolvimento de *software*
3. Tipologia contratual
4. Cessação do contrato
5. Responsabilidade contratual
6. Considerações finais
7. Referências

## Introdução

O artigo analisa os contratos de desenvolvimento e manutenção de *software* sob medida, contratos que, a princípio, não necessariamente são duradouros, mas que em sua forma mais frequente exigem uma atuação pelo menos por prazo prolongado, a poder estender-se por períodos inclusive superiores a cinco anos. A relação contratual se estabelece, de regra, entre um contratante que busca a prestação de serviços de alta complexidade com mão de obra e *know-how* das quais não dispõe e de uma empresa contratada especializada no desenvolvimento de *software* para a satisfação de requisitos específicos.

Neste contexto, variados são os *frameworks* de gerenciamento e organização de projetos utilizados para o desenvolvimento de *software*, ainda que dois sejam os mais frequentes: o modelo clássico em cascata (do inglês *waterfall model*) e o *scrum* (o maior expoente do chamado desenvolvimento ágil)<sup>1</sup>. Em virtude de não haver uma previsão legislativa específica para tais contratos, muitos deles assumem características de contratos de empreitada ou contratos inominados de prestação de serviços, dependendo das condições convencionadas, o que tem efeitos sobre a qualificação das obrigações contratuais. Portanto, verifica-se grande discricionariedade das partes ao convencionar estes contratos. Devido a esta grande liberdade contratual das partes ao regular esta relação complexa e dependendo do *framework* utilizado pode se verificar interessantes constelações contratuais que surtem efeitos sobre as matérias da cessação e responsabilidade contratuais.

No artigo analisar-se-á em primeiro passo o caráter duradouro deste tipo de contratação. A seguir será introduzida a matéria dos *frameworks* de desenvolvimento de *software* e quais os efeitos que podem ter na busca de um tipo contratual adequado. Nesse sentido, serão pontuados alguns desafios oferecidos pelo *scrum* e possíveis constelações contratuais. Por fim, serão explicitadas quais as influências que o caráter duradouro e os *frameworks* podem ter sobre as matérias da cessação e responsabilidade contratuais e quais alternativas a que podem recorrer as partes ao convencionar estes contratos.

## 1. O caráter duradouro dos contratos de desenvolvimento e manutenção de *software*

Via de regra, o caráter duradouro de um contrato pode derivar tanto da ausência de fixação de um prazo determinado ou da fixação de um longo prazo de

---

1 Cfr. outros *frameworks* de desenvolvimento ágil como *eXtreme Programming* em Koch (2010), p. 114.

Os contratos de desenvolvimento e manutenção de *software*: o caráter duradouro e as influências do *scrum* sobre a tipologia, a cessação e a responsabilidade contratuais \ **Vinicius Aquini Gonçalves**

duração<sup>2</sup>. Há ainda contratos que assumem essa característica pelo fato de o prazo contratual ser sucessivamente dilatado, em razão dos efeitos da renovação automática<sup>3</sup>. Alguns destes contratos renovam-se por tempo certo, outros por tempo indeterminado.

Os contratos de desenvolvimento e manutenção de *software* não são contratos necessariamente duradouros, ainda que possam ser de tal sorte acordados. Muitos desses contratos se iniciam pelo estabelecimento de um contrato-quadro que definirá as linhas gerais do negócio jurídico, tal como a responsabilidade contratual, cláusulas de escolha de foro e direito, condições gerais da prestação, etc. Nestes contratos-quadro vê-se seguidamente ausente a determinação de um prazo, ao mesmo tempo que relegam a especificação das prestações a contratos subjacentes específicos, o que no direito alemão se consubstancia na composição entre *Rahmenvertrag* e *Einzelaufträge*. Devido a esta ausência de termo, frequente é a fixação de uma cláusula resolutiva que se assemelha à denúncia de um contrato duradouro, pois trata-se de uma manifestação unilateral de vontade.

Outrossim, muitos desses contratos passam de uma fase de desenvolvimento para uma fase de operacionalização (do inglês *Operations*)<sup>4</sup>, que se prolonga por tempo indeterminado ou há no contrato uma cláusula de renovação automática. Alguns contratos preveem ainda a concomitância do desenvolvimento e da operacionalização, a partir da prática chamada *DevOps*<sup>5</sup>, geralmente convencionados por um prazo prolongado de execução. Existem também aqueles que têm por objeto exclusivamente a manutenção do *software*, o que em alguns casos pode se estender por um período superior a dez ou vinte anos.

O contrato em si não tem enquadramento legal como contrato duradouro, porém, a sua matéria e características de desenvolvimento, na prática, levam à consideração destes como com um caráter duradouro. A própria natureza da matéria objeto do contrato poderia ser considerada de cunho duradouro, pois diferentemente de uma obra com cunho material que se pode atestar terminada uma vez entregue, devidamente fiscalizada e aceite pelo contratante/dono da obra, o desenvolvimento do *software* não se extingue, ainda que possa ser pausado durante certo período. O programa sempre pode ser expandido, passar a compor novas funcionalidades e, à medida que um novo sistema operacional

---

2 CoELHO (2017), p. 226.

3 CoELHO (2017), p. 232.

4 *IT Operations* engloba tanto o acompanhamento de processos administrativos quanto a prestação de suporte para usuários internos e externos do *software*. Algumas questões que estão relacionadas entre outras são a garantia da qualidade (do inglês *quality assurance*), a administração da infraestrutura e das redes. Cfr. maiores detalhes em BRUSH; HAUGHN (2020).

5 Uma extensa explicação sobre DevOps pode ser encontrada em MICROSOFT AZURE (2020).

ou novas versões da linguagem de programação são lançadas, ou mesmo novos tipos de vírus são desenvolvidos, faz-se necessária sua modernização. Além disso, diferentemente do que demais produtos, tal como um prédio da construção civil, por exemplo, os ciclos de inovação e obsolescência de *softwares* são muito rápidos, o que faz com que estes fiquem rapidamente obsoletos.

Por estas razões, mesmo havendo algumas facetas contratuais que desnaturem o caráter duradouro dos contratos de desenvolvimento e manutenção ou mesmo quando são convencionados por tempo certo, a natureza material da relação subsistente, devido a essa inesgotabilidade do desenvolvimento do *software*, tem um caráter de duração pelo menos prolongada, se não dirá mesmo duradoura.

## **2. Frameworks de gerenciamento e organização de projetos de desenvolvimento de *software***

### **2.1. Modelo em cascata**

O modelo em cascata é um modelo clássico de desenvolvimento de *software* de forma linear, a partir da identificação de fases que são desenvolvidas sequencialmente, para frente<sup>6</sup>. As fases compreendem, de regra, 1) a análise de requisitos, 2) o projeto, 3) a implementação, 4) os testes (validação), 5) a integração e 6) a manutenção do *software*<sup>7</sup>. Em geral, estas fases são desenvolvidas por equipes diferentes. Por exemplo, tão logo a equipe de análise de requisitos termina a análise para o primeiro incremento, inicia-se a análise dos requisitos para o segundo incremento. Um incremento pode ser, por exemplo, uma nova funcionalidade a ser oferecida pelo *software*.

De acordo com este modelo é imprescindível um longo planejamento anterior ao início de fato da execução do projeto, o que possibilita realizar uma melhor apreciação dos custos e ajuste da remuneração do projeto, além de viabilizar ao contratante definir uma longa lista de requisitos para o *software* a serem satisfeitos pelo prestador, bem como de um cronograma para a execução do projeto com diversas etapas, marcos do projeto, a serem atingidos em prazos determinados. Com base neste planejamento, alguns contratos preveem, caso não sejam atingidos os marcos do projeto, a obrigação de o contratado pagar valores de acordo com a gravidade de sua mora, os quais são fixados em cláusulas penais.

---

6 LUTZ; BACH (2017), p. 3016. Cfr. outras variações do modelo em cascata como o modelo V em KREMER (2010), pp. 284-285; sobre outro paradigma clássico, ITIL, SÖBBING (2010), pp. 222-224.

7 SARRE, in AUER-REINSDORFF; CONRAD (2019), Teil A, §1, Nm. 26.

Os contratos de desenvolvimento e manutenção de *software*: o caráter duradouro e as influências do *scrum* sobre a tipologia, a cessação e a responsabilidade contratuais \ **Vinicius Aquini Gonçalves**

Na forma do modelo em cascata, os ciclos, isto é, o tempo que leva até certo incremento ser entregue ao contratante, tomam em torno de três a seis meses, em casos extremos podendo levar um ano. A comunicação e a interação com o cliente/contratante não são, de regra, muito frequentes, mas se há – no meio de uma iteração – a necessidade de adequação dos requisitos, o contratante pode fazer um pedido de alteração formalizado, do inglês *change request*, o que em geral não deve acontecer com alta frequência<sup>8</sup>.

Decorrente destas características, o modelo apresenta algumas desvantagens inerentes. Devido à baixa interação com o contratante durante a implementação, muitas vezes, ao final do ciclo constata-se que as funcionalidades desenvolvidas não correspondem às expectativas do contratante. Além disso, habitualmente os marcos do projeto definidos não são atingidos, pois é bastante difícil mensurar com precisão e tanta precedência a demanda de recursos/ tempo de cada incremento<sup>9</sup>. No decorrer do desenvolvimento é usual a existência de um *trade-off* entre desenvolver mais testes, o que assegura um melhor funcionamento do *software*, ou desenvolver mais funcionalidades. As cláusulas penais importam em alto risco para o contratado. Muitos dos contratantes impõem um preço fixo do projeto ao contratado que, principalmente no caso de ocorrerem muitos pedidos de alteração, torna-se extremamente pouco atraente para o contratado. Por último, devido ao fato de os ciclos serem tão longos, corre-se o risco de no momento de o ciclo ser findado, os incrementos do programa desenvolvidos não serem mais suficientemente modernos e existirem já no mercado soluções com melhor desempenho, uma vez que se trata de um mercado com alta velocidade no sentido inovação/obsolescência<sup>10</sup>.

## 2.2. Desenvolvimento ágil: *scrum*

*Scrum*, por sua vez, é o modelo mais popular de desenvolvimento ágil. Os *frameworks* de desenvolvimento ágil foram criados sob a ciência de que os projetos de desenvolvimento de programas apresentam grandes inconstâncias, uma vez que há incessante alteração e extensão das especificações e dos requisitos a serem satisfeitos e funcionalidades que o programa deve oferecer. Diferentemente do modelo em cascata, modelos ágeis não são lineares, mas iterativos e incrementativos, possibilitando ampla comunicação com o cliente/contratante. Alguns dos preceitos que orientam o desenvolvimento ágil são a auto-organização, a crossfuncionalidade das equipes, e o fornecimento de

---

8 FUCHS *et al.* (2012), p. 428.

9 SARRE, in: AUER-REINSDORFF; CONRAD (2019), Teil A, §1, Nm. 30.

10 SARRE, in: AUER-REINSDORFF; CONRAD (2019), Teil A, §1, Nm. 31.

um serviço mais orientado para o cliente. O modelo *scrum* implementa esses preceitos de formas que demonstraremos a seguir. Para entender os problemas jurídicos inerentes à utilização do *scrum*, faz-se necessário primeiramente explicá-lo, a partir da pormenorização de suas diferentes características.

### 2.2.1. Visão do produto

Trata-se do ponto de partida de um projeto de desenvolvimento sob a égide deste modelo. O contratante apresenta a sua visão e com o auxílio do contratado vai iterativamente expandir essa visão em objetivos primordiais e conexos. A visão é descrita em um documento que é parte integrante do contrato de desenvolvimento<sup>11</sup>. Nesta visão são descritas as características gerais do produto/*software* a ser desenvolvido e as suas finalidades, em especial econômicas. A visão, entretanto, não é estática, podendo ser adequada durante o desenvolvimento, de acordo com a superveniência mesmo de inovações tecnológicas, funcionalidades que podem melhor atender às necessidades do contratante. Portanto, não há, anterior ao início do projeto, a criação de uma longa lista de requisitos a serem satisfeitos e funcionalidades a serem oferecidas pelo *software*<sup>12</sup>. Economiza-se assim em muito o tempo despendido na fase pré-contratual, incorporando-se a análise de requisitos na fase de execução do contrato.

### 2.2.2. Procedimento em *sprints*

*Sprint*, a também chamada iteração, é o período de tempo, em geral de duas a quatro semanas, em que são desenvolvidos incrementos do *software*<sup>13</sup>. Cada *sprint* compreende a realização de quatro tipos de reuniões<sup>14</sup>.

No início realiza-se uma reunião de planejamento em que se define quais são as tarefas constantes no *product backlog*, uma listagem de todas as funcionalidades e requisitos, que devem ser desenvolvidas na iteração seguinte e quais são os objetivos desta iteração, como por exemplo a entrega de uma ou mais funcionalidades. Caso os objetivos não sejam alcançados na iteração, há uma redefinição e repriorização dos objetivos<sup>15</sup>. A duração do *sprint*, no entanto, não é alterada.

Todos os dias se realiza o *daily scrum*, uma reunião de quinze minutos que, além de ser uma oportunidade de os membros da equipe informarem os demais

---

11 HOEREN; PINELLI (2018), p. 201; SCHWABER; SUTHERLAND (2017), p. 5.

12 FUCHS *et al.* (2012), p. 428.

13 HOEREN; PINELLI (2018), p. 201.

14 HOEREN; PINELLI (2018), p. 201.

15 SCHWABER; SUTHERLAND (2017), p. 10; HOEREN; PINELLI (2018), p. 201.

Os contratos de desenvolvimento e manutenção de *software*: o caráter duradouro e as influências do *scrum* sobre a tipologia, a cessação e a responsabilidade contratuais \ **Vinicius Aquini Gonçalves**

sobre o *status* do que estão desenvolvendo, é o momento em que se faz o planejamento do dia<sup>16</sup>.

O *sprint review* ocorre no final do *sprint* e é a reunião em que o contratante e/ou o dono do produto participa, na qual os resultados são apresentados e se demonstra as novas funcionalidades que foram implementadas e se informa sobre o *status* atual de execução do contrato<sup>17</sup>. Nessa oportunidade há uma certa entrega parcial do produto/obra desenvolvida.

Por último, há ainda a retrospectiva que é uma reunião entre os membros da equipe de desenvolvimento e o *scrum master* para reavaliar dificuldades e êxitos no decurso do *sprint* encerrado<sup>18</sup>.

### 2.2.3. Papéis

O modelo *scrum* prevê a existência de três papéis: dono do produto (*product owner*), *scrum master* e equipe de desenvolvimento. Todos estão comprometidos com o desenvolvimento do produto, ainda que desempenhem funções diferentes. Não necessariamente o contratado preenche todas essas posições. Em regra, o contratante assume a posição de dono do produto, enquanto o contratado assume as funções de *scrum master* e compõe a equipe de desenvolvimento.

#### 2.2.3.1. Dono do produto

O dono do produto, ou *product owner*, é responsável pela coordenação do trabalho da equipe de desenvolvimento através da elaboração de requisitos que devem ser satisfeitos pelo programa (*user stories*) e que compõem o *product backlog*, o qual pode ser dinamicamente alterado e estendido durante o desenvolvimento do projeto<sup>19</sup>. O dono do produto também atua como ponto de comunicação entre a equipe de desenvolvimento e os demais *stakeholders*<sup>20</sup>, sendo responsável por priorizar as tarefas elencadas no *product backlog* e assegurar que a sua execução se dê de acordo com a visão definida na fase pré-contratual.

---

16 SCHWABER; SUTHERLAND (2017), p. 13.

17 SCHWABER; SUTHERLAND (2017), p. 13; SÖBBING (2014), p. 217.

18 SCHWABER; SUTHERLAND (2017), p. 14.

19 BORTZ (2018), p. 290; SCHWABER; SUTHERLAND (2017), p. 6; LITSCHEN; YACOUBI (2017), p. 487.

20 *Stakeholders* podem ser nesse caso investidores, outros funcionários do contratante, empresas terceiras, etc. Cfr. SARRE, in AUER-REINSDORFF; CONRAD (2019), Teil A, §1, Nm. 60.

### 2.2.3.2. Equipe de desenvolvimento

A equipe de desenvolvimento é responsável por entregar ao final de cada *sprint* os incrementos do produto finalizados. É dizer, todas as tarefas de implementação, testes, *reviews* são de encargo desta equipe<sup>21</sup>. A equipe organiza-se com autonomia, sem gerência externa direta, e define como melhor executará as tarefas elencadas no planejamento do *sprint*<sup>22</sup>. Esta equipe pode ser mista, isto é, dela fazendo parte funcionários do contratante, funcionários da empresa contratada e em alguns casos ainda outros prestadores de serviço<sup>23</sup>.

### 2.2.3.3. Scrum master

Muitas vezes a posição do *scrum master* é preenchida por um membro da equipe de desenvolvimento, ainda que não seja ideal. Algumas vezes, estabelece-se um contrato em separado de prestação de serviço entre as partes quando o contratado preenche a função do *scrum master*<sup>24</sup>. O papel do *scrum master* é semelhante ao de um *coach* ou moderador, que assegura que tanto o contratante quanto a equipe de desenvolvimento possam trabalhar cooperativamente de acordo com o plano estabelecido, resolvendo eventuais conflitos<sup>25</sup>. De regra, o *scrum master* assume também a responsabilidade por organizar e moderar todas reuniões mais importantes do projeto<sup>26</sup>.

## 2.2.4. Remuneração

No tocante à remuneração há interesses divergentes das partes. Ao passo que o contratante busca a fixação de um preço máximo ou mesmo global, o contratado busca ser pago dinamicamente pelo tempo despendido<sup>27</sup>. Devido à dinamicidade do projeto e à dificuldade de construção de um cronograma exato – uma vez que o contratante constantemente pode aumentar a dimensão da prestação contratada –, a fixação de um preço global pode ser excessivamente onerosa para o contratado/prestador. No caso de o preço fixo restar acordado entre as partes, o contratante tende a ofertar um valor que compreenda uma

---

21 HOEREN; PINELLI (2018), p. 200; BORTZ (2018), p. 290.

22 SÖBBING (2014), p. 216.

23 BORTZ (2018), p. 290; SCHWABER; SUTHERLAND (2017), p. 7.

24 BORTZ (2018), p. 290.

25 LITSCHEN; YACOUBI (2017), p. 487; HOEREN; PINELLI (2018), p. 200.

26 BORTZ, 2018.

27 VON SCHENCK (2019), p. 139.

Os contratos de desenvolvimento e manutenção de *software*: o caráter duradouro e as influências do *scrum* sobre a tipologia, a cessação e a responsabilidade contratuais \ **Vinicius Aquini Gonçalves**

margem de segurança, isto é, um pouco acima da expectativa, a fim de evitar que seu retorno econômico seja demasiado erodido<sup>28</sup>. Saliente-se não ser, por estas razões, a remuneração global muito adequada ao desenvolvimento por *scrum*.

### **2.2.5. Conclusão: *definition of done***

A conclusão de um projeto destes é um tanto difícil de constatar, uma vez que, conforme supramencionado, geralmente não se termina de desenvolver um programa. De regra os *softwares* individualizados ou sob medida são constantemente expandidos e seguem sendo desenvolvidos por longo espaço de tempo e após o decurso de certo tempo ainda ocorrem as chamadas migrações, do *software* escrito na versão 1.0 para a versão 2.0 da linguagem de programação (por exemplo, da versão Java 8 para Java 10), a qual oferece maiores recursos.

A conclusão de um projeto de acordo com o *scrum* se daria, de certa forma, quando o *backlog* fosse esvaziado, o que no caso é um tanto difícil, pois este em geral jamais se esgota. Na maioria dos casos, a relação jurídica entre as partes esgota-se quando o contratante resolve assumir a manutenção do programa e continuar o desenvolvimento por sua conta, ainda que com menor velocidade e, muitas vezes, profissionalismo. Comum também é a finalização do projeto porque é fixado contratualmente o fim da vigência do contrato ou quando, no momento da renovação contratual, o contratante abre a possibilidade para que outros prestadores façam oferta de preço. Não fazendo o atual contratado a melhor oferta, o contrato é encerrado e outra empresa passa a assumir o desenvolvimento e a execução do projeto.

Embora a conclusão de um projeto desta natureza seja de difícil constatação – com exceção dos casos acima mencionados –, durante a sua execução são previstas conclusões parciais dos chamados incrementos. O modelo *scrum* prevê que seja elaborada pelas partes uma *definition of done*, quando um incremento estaria encerrado definitivamente. Essa definição ajuda não somente o planejamento dos *sprints*, mas também a mensurar quantos incrementos podem ser atingidos em uma iteração. Muitas vezes o incremento só é atestado como *done* à medida que toda a documentação<sup>29</sup> estiver escrita, o que possibilita que terceiros entendam o funcionamento do código escrito e possam alterá-lo, e que este tenha uma boa cobertura de testes, isto é, seja assegurado que o código funciona para um número “x” de situações fáticas.

---

28 VON SCHENCK (2019), p. 139.

29 Cfr. com maior aprofundamento sobre a documentação em SARRE, in: AUER-REINSDORFF; CONRAD (2019), Teil A, §1, VI, Nm. 514 e ss.

### 3. Tipologia contratual

A questão fundamental que permeia a discussão da tipologia contratual está relacionada ao caráter das obrigações subjacentes, se obrigações de fim ou de meio<sup>30</sup>. Desta discussão decorre a problemática de quem assume o risco do projeto, se contratante ou contratado<sup>31</sup>. Os contratos de desenvolvimento e manutenção de *software* levitam fundamentalmente entre duas classificações, contratos de serviços inominados e contratos de empreitada<sup>32</sup>.

A distinção principal entre os contratos de prestação de serviços inominados e os de empreitada é o fato de na empreitada ao final o resultado do trabalho constituir uma obra<sup>33</sup>. Em diversos ordenamentos jurídicos diferencia-se os dois tipos contratuais a partir de uma distinção entre um serviço, simples, e a obra, isto é, se o objeto da prestação é o serviço em si mesmo ou o resultado<sup>34</sup>. Ao passo que o contrato de empreitada encontra regulação específica nos Arts. 1207 e ss. CC, o contrato de serviços inominado ou atípico orienta-se pelas disposições relativas ao contrato de mandato (Art. 1156 CC).

Faz-se aqui repetidamente a remissão ao direito alemão, que vê uma dicotomia entre dois tipos contratuais: *Dienstleistungsvertrag* (contrato de prestação de serviços) e *Werkvertrag* (contrato de empreitada). Não há como no direito português uma relação de especificidade entre os dois tipos contratuais (conforme previsto no Art. 1155 CC), pois trata-se de tipos distintos com alguns elementos comuns, não sendo o *Werkvertrag* visto como uma modalidade de *Dienstleistungsvertrag*<sup>35</sup>.

Algumas considerações sobre os contratos de empreitada e de prestação de serviços inominado são necessárias para situar a matéria. Posteriormente, passa-se a delinear as influências da utilização do *scrum* sobre a tipologia contratual.

---

30 KARGER (2004), p. 209.

31 KOCH *et al.* (2019), p. 708.

32 Ressalte-se que há outras alternativas de enquadramento que, por parecerem-nos demasiado inadequadas, não serão aqui tratadas, como a qualificação como contratos de compra e venda e contratos societários. Cfr. esta discussão a nível do direito alemão em KARGER (2004), pp. 209-210. No mesmo sentido sobre a inadequação de outros tipos contratuais cfr. KOCH *et al.* (2019), p. 708. O direito alemão ainda conhece a figura de um tipo contratual diverso, isto é, dos contratos de cessão de mão de obra, *Arbeitnehmerüberlassung*, que são bastante comuns no contexto do desenvolvimento de *software*. Cfr. com maior profundidade em KARSTEN *et al.* (2016), pp. 172 e ss. Cfr. ainda as últimas alterações da *Arbeitnehmerüberlassungsgesetz* de 1.4.2017 em LITSCHEN; YACOUBI (2017), pp. 484 e ss.

33 MENEZES LEITÃO (2019), p. 498.

34 ALMEIDA (2016), p. 156.

35 O direito alemão ainda conhece uma distinção entre o contrato de empreitada, *Werkvertrag*, e o contrato de fornecimento, *Werklieferungsvertrag*, que diz respeito ao fornecimento de coisas móveis/mercadorias a serem fabricadas ou produzidas e que se orienta pelo direito da compra e venda.

Os contratos de desenvolvimento e manutenção de *software*: o caráter duradouro e as influências do scrum sobre a tipologia, a cessação e a responsabilidade contratuais \ Vinicius Aquini Gonçalves

### 3.1. Contrato de empreitada

Usualmente, os contratos de desenvolvimento/implementação de *software* eram identificados pela doutrina e judiciário como contratos de empreitada<sup>36</sup>. Essa classificação é mais fácil de visualizar em um contrato sob a utilização do modelo em cascata, uma vez que é de início definido um grande catálogo de especificações que devem ser desenvolvidas/entregues de acordo com um cronograma definido e que ainda se baseiam de regra na remuneração por preço global.

Nesta tipologia contratual, se ao final a “obra” não corresponde ao que foi encomendado/contratado, deve o dono da obra/contratante denunciar os vícios ao empreiteiro/contratado (Art. 1220 CC) para fins de que este elimine os defeitos (Art. 1221 CC) ou, se constatar algum defeito essencial durante o cumprimento contratual, pode exigir a resolução (Art. 1222 CC) e, em alguns casos, exercer o seu direito à indenização (Art. 1223 CC). Durante a execução do contrato pode ainda o dono da obra/contratante às suas custas fiscalizar a execução da empreitada (Art. 1209 CC), por perito técnico especializado.

Ao final deve o dono da obra aceitá-la, o que importa na transferência da propriedade sobre a obra (Art. 1212 CC). É dizer, a aceitação é parte imprescindível nesse tipo contratual. Exige-se que o dono da obra, antes da aceitação, verifique se a obra respeita as condições convencionadas e sem vícios (Art. 1218 CC). Ressalve-se, no entanto que, mesmo sob a tipologia de empreitada, a transferência da propriedade do projeto de manutenção e desenvolvimento do *software* não ocorre somente ao final do projeto, uma vez que há entregas parciais durante a execução do contrato. Isso, porque, embora a transferência da propriedade posterior à aceitação seja mais adequada a empreitadas que tenham por objeto uma construção de bem corpóreo, o mesmo não ocorre no caso ora em tela<sup>37</sup>.

Destaque-se que os contratos de empreitada não são contratos de execução instantânea, mas sim prolongada, ainda que interesse ao dono da obra de regra somente a obra final. O tempo de execução é um fator delimitador do conteúdo da obrigação, especialmente quando a possibilidade de entregas parciais é levada em conta<sup>38</sup>. No caso dos contratos de desenvolvimento de *software*, o tempo é um elemento essencial, tendo em vista que inclusive a contraprestação pecuniária ocorre muitas vezes atrelada ao tempo, e ao contratante interessar quais são os incrementos que estarão sendo entregues até certa data. Isso, pois, a despeito de uma obra cuja fruição inicia-se somente após a

---

36 Cfr. menção em Acórdão TR de Lisboa de 08.09.2015 (Maria do Rosário Morgado).

37 MENEZES LEITÃO (2019), p. 524; ALMEIDA (2016), p. 157.

38 MENEZES LEITÃO (2019), p. 504.

entrega final, a fruição do *software* dá-se durante o período em que muitos dos incrementos estão a ser desenvolvidos. O DL 252/94 sobre a proteção jurídica de programas de computador<sup>39</sup> ainda é claro no Art. 3 (3) ao reputar a titularidade de direitos ao destinatário do programa, quando o *software* for desenvolvido sob encomenda.

Relativamente ao objeto da obra, há a discussão doutrinária acerca da adequação do contrato de empreitada para a execução de obras intelectuais. Parte da doutrina se posiciona contrariamente, porque seria este mais destinado às obras corpóreas, sendo o contrato mais adequado um contrato de encomenda de obra intelectual, previsto no Art. 14 CDADC<sup>40</sup>. De acordo com esta linha, muitos dos dispositivos do contrato de empreitada, como os relativos ao regime de fiscalização (Art. 1209 CC), da transferência de propriedade (Art. 1212 CC), das alterações (Arts. 1214 e ss. CC) e dos defeitos da obra (Arts. 1218 e ss. CC), não seriam necessariamente compatíveis com a obra intelectual<sup>41</sup>. A natureza incorpórea do resultado, no entanto, seria irrelevante para a qualificação do resultado como obra, não impedindo ainda se recorrer ao regime da propriedade intelectual<sup>42</sup>.

Na jurisprudência portuguesa recorrente é a tese da não possibilidade de qualificar o desenvolvimento de *software*, enquanto obra não corpórea/imaterial, no regime dos Arts. 1207 e ss. CC<sup>43</sup>. No entanto, deve-se dizer que já em 1983, o STJ reconheceu a possibilidade de o contrato de empreitada ter por objeto uma obra eminentemente intelectual<sup>44</sup>. Outros acórdãos do mesmo tribunal vão, entretanto, em sentido diverso<sup>45</sup>. Em verdade, deve-se dizer que o ponto ainda é gerador de alguma controvérsia<sup>46</sup>.

Em algumas constelações, quando o contrato de desenvolvimento de *software* não envolve o desenvolvimento de *software* individual para o contratante, mas, por exemplo, a customização e adaptação de um *software* standardizado, como os sistemas integrados de gestão empresarial oferecidos por grandes empresas de tecnologia como SAP, Microsoft e Sales Force, o contrato pode

---

39 Que transpõe a Diretiva n.º 91/250/CEE, do Conselho, de 14 de maio. Cfr. para maior aprofundamento a questão em PEREIRA (2011).

40 ALMEIDA (2016), pp. 156-157. Ainda que tipificado o contrato de encomenda de obra intelectual no Art. 14 CDADC, não há uma regulamentação exaustiva como é o caso do contrato de empreitada previsto no Código Civil.

41 MENEZES LEITÃO (2019), p. 505.

42 ALMEIDA (2016), p. 158.

43 Cfr. por exemplo o Acórdão TR de Lisboa de 04.07.2019 (António Santos).

44 Acórdão do STJ de 03.11.1983 (Santos Silveira).

45 Acórdão do STJ de 21.11.2006 (Sebastião Póvoas).

46 ALMEIDA LEITE (2016), p. 37.

Os contratos de desenvolvimento e manutenção de *software*: o caráter duradouro e as influências do *scrum* sobre a tipologia, a cessação e a responsabilidade contratuais \ **Vinicius Aquini Gonçalves**

adesivamente receber características de um contrato de compra e venda<sup>47</sup>. Um contrato que envolva somente a aquisição e instalação de “equipamento informático” poderia ainda ser qualificado como um contrato misto de compra e venda e de empreitada<sup>48</sup>. Nesse caso, pode o equipamento envolver também um conteúdo corpóreo significativo, como servidores entre outros.

Deve-se ainda dizer que o regime dos contratos de empreitada em Portugal não é tão abrangente como a empreitada nos direitos alemão e francês (*louage d’ouvrage*, Arts. 1710 e ss. CC francês, e *Werkvertrag*, §631 e ss. BGB), visto que nos dois ordenamentos é corrente a utilização para obras de bens não corpóreos e, por essa razão dispõem os ordenamentos de dispositivos específicos. Além disso, ainda que alguns dos contratos de desenvolvimento de *software* tenham alguma componente residual corpórea como a disponibilização e configuração de infraestrutura (como por exemplo a aquisição e operacionalização de *hardware*), está-se em face de um objeto contratual primordialmente não corpóreo.

Embora se reconheça que o quadro legislativo e a jurisprudência levem em muitos casos à desclassificação de contratos de criação de obras intelectuais como de empreitada, defende-se aqui que, excepcionalmente, muitos elementos lhe possam ser transportados por analogia, uma vez que os contratos têm geralmente uma natureza mista<sup>49</sup>.

### 3.2. Contrato de prestação de serviços inominados

Na ausência de uma definição de prestação e serviço no Código Civil português, está-se em face de conceitos cuja interpretação deve ser em *lato sensu*. No mesmo sentido, a Diretiva n.º 91/250/CEE, relativa aos serviços no mercado interno, oferece uma definição bastante ampla no Art. 4, I, de serviço como qualquer atividade econômica não assalariada prestada “geralmente” mediante remuneração. Portanto, sequer a remuneração é um elemento constituinte.

O contrato de prestação de serviços, de acordo com o Art. 1154 CC, descreve um contrato em que uma das partes proporciona a outra o resultado do seu trabalho, seja intelectual ou manual. A distinção, se estamos em face de um contrato inominado de prestação de serviços ou de empreitada, exige uma análise do contrato. Ainda que na definição de contratos de prestação de serviços

---

47 Acórdão TR de Lisboa de 04.07.2019 (António Santos), 9584/16.0T8LRS.L1-6. Cfr. também de maneira mais aprofundada sobre as considerações da compra e venda de *software* em PEREIRA (1999), pp. 930 e s.

48 Acórdão do STJ de 05.07.1994 (Carlos Caldas).

49 BATISTA FERREIRA (2015), p. 85.

mencione-se genericamente um resultado, a diferença está no elemento central da obrigação, isto é, se é o serviço a ser prestado ou a entrega de uma obra<sup>50</sup>. Ao passo que no contrato de serviço inominado a prestação adequada do serviço é devida, no contrato de empreitada deve lograr o empreiteiro determinado sucesso de acordo com as especificações previamente definidas<sup>51</sup>. Não se assevera, entretanto, que não seja possível fixar em contrato de serviços inominado critérios mais exaustivos para delimitar a prestação, que ao final importem na assunção de riscos pelo “sucesso” do *software* por parte do contratado.

Como o contrato de desenvolvimento e manutenção de *software* sob medida não é uma modalidade especificamente regulamentada no direito português, este poderia orientar-se, com adaptações específicas, pelas disposições do contrato de mandato dos Arts. 1157 e ss. CC<sup>52</sup>. A aplicação das normas do contrato de mandato não impede o recurso, por analogia, a dispositivos do contrato de empreitada. Dependendo do fixado entre as partes, pode a natureza contratual como contrato de prestação de serviços inominado restar incontrovertida<sup>53</sup>. Observa-se que carece o direito pátrio de um tipo contratual genérico para a prestação de serviços, uma vez que o contrato de mandato se mostra em grande medida muito específico.

De toda forma, pode-se fazer recurso a algumas disposições do contrato de mandato, das quais decorrem obrigações para o contratado, mais especificamente, de executar ou praticar atos segundo as instruções do ou por conta do contratante, prestar ao último informações, comunicar a execução, prestar contas, entre outros (Art. 1161 CC). Na execução, o contratado tem certa discricionariedade, quando razoável supor que a conduta seria aprovada pelo contratante, podendo deixar de executar o contrato ou não seguir exatamente certas especificações do contratante (Art. 1162 CC). Nos contratos de desenvolvimento de *software* esta disposição poderia ter efeitos relevantes, pois frequente é o desconhecimento do contratante da matéria devido a grande especificidade e complexidade da utilização de determinadas tecnologias, ou seja, haveria mais liberdade para o contratado em agir, ainda que de acordo com os interesses e a assim chamada “visão” do contratante.

---

50 Cfr. analogicamente BAUMGÄRTNER, in: BAMBERGER *et al.* (2020), §611, Nm. 10; SÖBBING (2010), p. 225; WITTE (2010), p. 47.

51 *Baumgärtner*, in: BAUMGÄRTNER, in: BAMBERGER *et al.* (2020), §611, Nm. 10; sobre a assunção de riscos pelo contratado WITTE (2010), p. 47.

52 De acordo com a jurisprudência do STJ, “a prestação de serviço é o genus ao qual são, subsidiariamente, aplicáveis as regras do mandato (species)”. Acórdão do STJ de 21.11.2006 (Sebastião Póvoas).

53 Nesse sentido cfr. Acórdão TR de Lisboa de 08.09.2015 (Maria do Rosário Morgado).

Os contratos de desenvolvimento e manutenção de *software*: o caráter duradouro e as influências do *scrum* sobre a tipologia, a cessação e a responsabilidade contratuais \ Vinicius Aquini Gonçalves

### 3.3. Alguns desafios relativos ao *scrum*

Embora defenda-se aqui a utilização do *scrum* como *framework* de organização e gerenciamento do projeto, este modelo gera certos desafios à tipificação dos contratos de desenvolvimento e manutenção do programa sob medida e à identificação dos efeitos dela decorrentes. Passa-se aqui a referir estes desafios:

1. Como não há uma fase pré-contratual extensa, com a definição da grande maioria dos requisitos e funcionalidades, torna-se difícil a constatação se o produto ao final corresponde ao que foi inicialmente contratado<sup>54</sup>. Com o fato de haver uma constante alteração do projeto e preenchimento do *product backlog*, a matéria contratada é muito dinâmica, ou seja, não há grande previsibilidade no início de como o programa será ao final do contrato. Mesmo a visão do produto pode ser alterada à medida que são desenvolvidas funcionalidades outras, as quais não sejam adstritas ao objeto essencial do *software* a ser desenvolvido<sup>55</sup>. Parte da doutrina, no entanto, não vislumbra nesse ponto um óbice à qualificação deste contrato como de empreitada, pois continuaria a ser devida uma prestação específica, ainda que concretizada durante a execução do contrato<sup>56</sup>.

2. Ainda no tocante à empreitada, a própria transferência da propriedade mediante a aceitação não se dá dessa forma, pois ainda que no *sprint review* haja uma apresentação dos incrementos ao contratante, não necessariamente o contratante nessa oportunidade fiscaliza o produto. Além disso, alguns dos módulos desenvolvidos no *software* podem não ser utilizados por um longo espaço de tempo. Por exemplo, o programa funciona para uma série de lógicas/situações, na realidade há outras situações que não foram adequadamente programadas. Somente depois de alguns meses quando se busca a utilização do *software* para uma situação específica, o defeito aparece, isto é, uma espécie de vício oculto. Por estas razões, a aceitação ocorre sem uma real possibilidade de análise *software* para uma grande porção de eventuais vícios.

3. Modernamente ainda são aplicados dois conceitos ao desenvolvimento do programa que vêm aliados à utilização do *scrum*, nomeadamente a integração contínua e a entrega contínua, do inglês *continuous integration* e *continuous delivery*. Integração contínua preceitua que haja a integração de todas as partes desenvolvidas (inclusive incrementos) pelos diferentes membros da equipe ou mesmo de equipes diferentes várias vezes ao dia<sup>57</sup>. Entrega contínua, por outro lado, preceitua a divisão do programa em vários pequenos ciclos para

---

54 SCHNEIDER (2010), pp. 19-20.

55 SCHNEIDER (2010), p. 21.

56 KREMER (2010), p. 285.

57 SCHMIDT, in: AUJER-REINSDORFF; CONRAD (2019), Teil A, §1, IV, Nm. 245; RED HAT (2020).

fins de assegurar que o *software* em qualquer momento possa ser entregue, isto é, que o produto desenvolvido possa vir a ser usado pelo destinatário final do programa a qualquer momento<sup>58</sup>. Decorrente desses dois conceitos, há entregas muitas vezes diárias do *software*. Portanto, torna-se difícil identificar uma aceitação conforme o Art. 1212 CC, porque o programa passa quase que imediatamente a ser utilizado. Poder-se-ia falar em um tipo de aceitação tácita ou por ato concludente, no entanto, muitas vezes o incremento é ativado sem que haja qualquer conhecimento do contratante antes da realização da finalização do *sprint*.

4. Uma das principais maneiras de viabilizar que o *software* escrito possa ser desenvolvido e compreendido por terceiros é a documentação, isto é, referências explicativas que podem tanto estar no corpo do código como em documento separado. A finalização da documentação também é vista como um dos elementos essenciais para a conclusão do projeto. Tendencialmente a documentação pelo *scrum* passa a ser menor, pois muitas vezes ao final do *sprint* falta tempo para escrevê-la e já se inicia o próximo *sprint*<sup>59</sup>. Por essa razão tornou-se mais frequente falar em práticas de *clean code*, para tornar o *software* compreensível sem que seja necessária muita documentação. A entrega de uma documentação correta muitas vezes é incluída nos deveres de diligência do contratado, sob a égide de um contrato de empreitada, a documentação deveria ser parte integrante do resultado a ser entregue, a sua ausência constituindo um vício.

### 3.4. Constelações contratuais

Conforme constata-se dos elementos supramencionados, não é possível uma tipificação clara dos contratos de desenvolvimento e manutenção de programas sob medida. Trata-se então de especificar contratualmente quais os elementos que este contrato deve assumir, tanto de um contrato de serviços inominado mais próximo do mandato quanto de um contrato de empreitada. De tal sorte algumas constelações contratuais são possíveis.

As partes poderiam convencionar a entrega de um resultado, a partir das especificações contratuais, o que se assemelha à empreitada<sup>60</sup>. No caso do *scrum* seria indispensável compartimentar a execução do contrato. Neste caso, far-se-ia necessário haver aceitações parciais que poderiam ser de acordo com o decurso dos *sprints*. Portanto, poder-se-ia firmar um contrato-quadro

---

58 SCHMIDT, in: AUER-REINSDORFF; CONRAD (2019), Teil A, §1, IV, Nm. 246; RED HAT (2020).

59 SCHNEIDER (2010).

60 MENEZES LEITÃO (2019), p. 500.

Os contratos de desenvolvimento e manutenção de *software*: o caráter duradouro e as influências do *scrum* sobre a tipologia, a cessação e a responsabilidade contratuais \ **Vinicius Aquini Gonçalves**

mais próximo de um contrato de serviços genérico em que constassem expressas as linhas gerais da prestação, acompanhado por pequenos contratos de empreitada para cada *sprint*. A vantagem para o contratante seria que o contratado assumiria nesse caso os riscos do “sucesso” do projeto, ainda que compartimentado.

Também é facultado às partes a convenção de um contrato amplamente ancorado no contrato de empreitada, o que, conforme dissertado anteriormente, não parece compatibilizar-se com o *scrum*, mas talvez com o modelo em cascata<sup>61</sup>. No caso da aplicação deste tipo contratual, ao final da execução do contrato, o contratante poderia fiscalizar e atestar se o *software* satisfaz as especificações predefinidas e, em caso contrário, exigir o saneamento dos vícios pelo contratado às custas deste último ou a minoração do preço, entre outros<sup>62</sup>.

Saliente-se, entretanto, que existem alguns projetos com um caráter mais científico, em especial quando se trata do desenvolvimento de novos algoritmos e mesmo da utilização de procedimentos da inteligência artificial, para os quais os riscos em relação ao resultado, independente do modelo utilizado, não poderiam ser assumidos pelo contratado, sob pena de onerosidade excessiva. Nestes projetos ainda que haja uma chance de sucesso, é possível que ao final os algoritmos ou métodos utilizados não logrem satisfazer a visão com a qual foi iniciado o projeto. Portanto, aqui em especial deve-se tender à qualificação das obrigações como obrigações de meio e aproximar este contrato de um contrato de prestação de serviços inominado, em que seja devida diligência profissional e não um resultado específico.

Além disso, poder-se-ia ainda convencionar que nem todas as prestações fossem desenvolvidas e gerenciadas de maneira ágil, por exemplo, as componentes de manutenção e mesmo a atividade do *scrum master* poderiam ser facilmente desmembradas, assim possibilitando a combinação de tipos contratuais, do que decorre certa “customização” da relação jurídica e uma diferenciação da tutela jurídica<sup>63</sup>. Dessa forma, o *scrum master* e os membros responsáveis pela manutenção do *software* poderiam ser exclusivamente responsáveis por agir com a diligência profissional adequada e os demais membros da equipe de desenvolvimento pela entrega de resultados especificados.

Outra alternativa seria introduzir a componente duradoura em um contrato-quadro para possibilitar a cessação por manifestação unilateral da vontade, acompanhado por contratos específicos por tempo determinado. Poder-se-ia

---

61 KOCH *et al.* (2019), p. 708. Nos casos em que o desenvolvimento é reduzido, isto é, mais adstrito a uma configuração, torna-se mais fácil fazer o recurso ao contrato de empreita. Cfr. Acórdão do TR de Lisboa de 18.09.2008 (Farinha Alves).

62 KOCH *et al.* (2019), p. 708.

63 HOEREN; PINELLI (2018), p. 200.

ainda convencionar um contrato em que o contratante assumia todos os riscos do negócio e o cunho da prestação mais se assemelhe a uma cessão de mão de obra por parte do contratado, neste caso sendo a total gerência do projeto resguardada ao contratante<sup>64</sup>. A seguir, conferir-se-ão quais os efeitos que tais constelações podem ter sobre a cessação e a repartição da responsabilidade.

#### 4. Cessação do contrato

Dependendo da constelação contratual convencionada entre as partes, diferentes são as alternativas de cessação destes contratos. Em geral, os meios de cessação vêm convencionados no contrato de implementação e manutenção de *software* sob medida. Ao passo que os contratos de manutenção pura são normalmente celebrados por tempo determinado, os contratos de desenvolvimento, especialmente quando compostos por um contrato-quadro e contratos específicos, podem ser celebrados com uma componente por tempo indeterminado.

No caso dos contratos por tempo determinado pode-se estar em face de contratos com cláusulas de renovação automática ou não. Para fins de cessar a renovação faz-se necessária a oposição do contratante ou do contratado para fins de ativar a caducidade. No caso dos contratos por tempo determinado sem renovação automática, dá-se a caducidade, ou seja, a cessação automática com o decurso do prazo sem que necessária a manifestação de vontade das partes<sup>65</sup>.

Outros contratos, cujo cunho mais se assemelha a um contrato de empreitada, dão-se por encerrados uma vez entregue a obra. No caso dos contratos que utilizam o modelo em cascata, uma entrega “final” da obra seria até possível. No caso dos contratos por *scrum*, na ausência de exaurimento do *backlog*, não é possível atestar uma entrega final. Nesse caso seria necessário compartimentar as obrigações em obrigações parciais e fixar para o final de cada *sprint* a possibilidade de cessação do contrato.

Alguns contratos mesmo por tempo determinado poderiam ainda prever a cessação por manifestação unilateral da vontade de uma das partes, isto é, por uma espécie de denúncia<sup>66</sup>. Não se trata aqui de uma razão por fato superveniente, pois esta também poderia ocorrer sem justa causa. Os contratos

---

64 Cfr. os efeitos de uma cessão de mão de obra sobre a utilização de *scrum* em HEISE; FRIEDL (2019), pp. 130-132.

65 Cfr. analogicamente MONTEIRO (2009), p. 132.

66 Ainda que a denúncia seja um instrumento privativo dos contratos duradouros, cfr. COELHO (2017), p. 228; MARTINEZ (2015), p. 230.

Os contratos de desenvolvimento e manutenção de *software*: o caráter duradouro e as influências do *scrum* sobre a tipologia, a cessação e a responsabilidade contratuais \ **Vinicius Aquini Gonçalves**

duradouros de implementação, por outro lado, em regra preveem uma cláusula de cessação por denúncia. A denúncia produz efeitos *ex nunc*, o que no caso destes contratos de implementação é muitas vezes postergada do momento em que declarada pela fixação de um aviso prévio<sup>67</sup>. Por exemplo, não rara é a fixação de prazos de três meses ou mais a partir da declaração, podendo chegar a um ano.

A grande vantagem destes contratos para o contratante é a possibilidade de aumentar e diminuir flexivelmente os recursos utilizados na execução do projeto, isto é, o contratante tem, na prática, a possibilidade de aumentar e diminuir o tamanho das equipes de desenvolvimento com certa flexibilidade. Por essa razão, muitas vezes o contratante, mesmo que pudesse contratar funcionários permanentes para desenvolver o programa, decide contratar uma empresa especializada, que de regra lhe custa mais<sup>68</sup>. Talvez se pudesse falar neste caso não de uma cessação universal, mas parcial do contrato em relação a membros específicos da equipe.

A presença de cláusulas de cessação no contrato não obsta a possibilidade de uma convenção superveniente entre as partes, a qual deve reunir os requisitos gerais de validade de um negócio jurídico<sup>69</sup>. De regra, a cessação por acordo mútuo prescinde de aviso prévio e tem efeitos imediatos<sup>70</sup>.

Por último, a resolução, ainda que também ocorrendo a partir de uma manifestação uniliteral da vontade, diferentemente da denúncia exige fundamentação, podendo ter efeitos imediatos, isto é, sem a concessão de aviso prévio à parte oposta<sup>71</sup>. Dois fundamentos podem ser transportados por analogia do contrato de agência [Art. 30, *a*) e *b*), DL 178/86], nomeadamente: 1) quando a outra parte reiterada e gravemente não cumprir as suas obrigações, tornando a subsistência do vínculo contratual inexigível; ou 2) se ocorrerem circunstâncias que tornem impossível ou prejudiquem gravemente a realização do fim contratual<sup>72</sup>. O incumprimento no caso dos contratos ora em tela pode de acordo com a primeira casuística, por exemplo, tanto ocorrer por falta ou atraso grave reiterado do pagamento das prestações pelo contratante, quanto pela comprovada

---

67 COELHO (2017), p. 230.

68 Neste atual contexto do coronavírus, por exemplo, muitos contratantes reduziram, com efeitos imediatos ou com aviso prévio de 20-30 dias, a mão de obra contratada em 50% ou mais. Disto denota o poder económico muitas vezes do contratante que, seguidamente, a despeito das disposições contratuais, age com excessiva discricionariedade. Saliente-se que a cessação por manifestação unilateral quando livre colide com os princípios do *pacta sunt servanda* e da segurança jurídica. Cfr. em relação à denúncia em MARTINEZ (2015), p. 233.

69 Cfr. analogicamente MONTEIRO (2009), p. 131.

70 MONTEIRO (2009), p. 131.

71 MONTEIRO (2009), p. 142.

72 MONTEIRO (2009), p. 143.

incapacidade técnica dos agentes do contratado em satisfazer o contrato, pelo reiterado descumprimento do cronograma estabelecido, ou ainda pela constatação de vícios no *software* que impossibilitem a sua utilização, não havendo uma reação adequada por parte do contratado. No tocante à segunda alternativa, poderia o fundamento decorrer, por exemplo, da falência de uma das partes ou ainda na emergência de uma causa de força maior que prejudique substancialmente a liquidez ou a mobilização de recursos por uma das partes, como é caso de muitas empresas hoje no contexto da crise do coronavírus.

## 5. Responsabilidade contratual

Não se trata aqui de uma análise aprofundada da responsabilidade contratual, dos deveres de contratados e contratantes, mas sim de situar exemplificativamente algumas dificuldades decorrentes da complexidade deste tipo de relação jurídica. Saliente-se, pois, que os limites da responsabilidade contratual nos contratos ora em tela pertencem ao direito dispositivo. Disto decorre que tanto componentes subjetivas quanto objetivas desta responsabilidade podem ser convencionadas. Neste sentido, restrições da responsabilidade, por exemplo, em relação a danos causados por negligência leve seriam possíveis, bem como convencionar a responsabilidade objetiva. Assim, passa-se à exposição de alguns desafios.

Destarte, dependendo da constelação contratual, vê-se possível a identificação de obrigações de resultado por parte do contratado, sendo exigível do contratado a eliminação de defeitos, no caso dos programas, as falhas, também chamadas *bugs*<sup>73</sup>. Considere-se também que pode a ausência de documentação ser identificada como um defeito do *software*, pois inviabiliza que terceiros compreendam partes do *software* ou que possam dar seguimento ao seu desenvolvimento. Não sendo estes eliminados ainda poder-se-ia recorrer à minoração do preço ou a resolução do contrato, conforme supramencionado<sup>74</sup>. Não se faria aqui, no entanto, renúncia a eventuais perdas e danos<sup>75</sup>. Acentue-se que defeitos graves que inviabilizam totalmente a utilização do programa podem gerar para o contratante prejuízos significativos, em especial quando o *software* é vital para o exercício da atividade econômica do contratante.

Relativamente aos casos em que não se está em face de um defeito evidente, mas de um *software* que não satisfaz a visão do contratante, se o contrato prevê a utilização de *scrum* com a eliminação de uma fase pré-contratual

---

73 LUTZ; BACH (2017), p. 3016.

74 SCHUSTER (2019), pp. 347-350.

75 SCHUSTER (2019), p. 350.

Os contratos de desenvolvimento e manutenção de *software*: o caráter duradouro e as influências do scrum sobre a tipologia, a cessação e a responsabilidade contratuais \ **Vinicius Aquini Gonçalves**

extensa de planejamento, torna-se difícil confrontar se ao final o produto corresponde ou não às especificações do contratante ensejando diretas a reparações. Neste contexto, seria possível identificar algum direito a reparações se repetidamente o contratado não cumprisse o planejamento definido nos *sprints*.

No caso de se estar em face de um contrato com maior preponderância de obrigações de meio, isto é, quando a gerência do projeto seja de total autonomia do contratante e a prestação de serviço mais se aproxime de uma cessão de mão de obra, não necessariamente seria exigível do contratante o saneamento de eventuais defeitos às custas do contratado. Claramente poderia o contratante engajar os agentes do contratado na solução do problema, mas o risco do negócio seria assumido pelo contratante.

Além disso, com o fato de frequentemente as equipes de desenvolvimento serem mistas, não necessariamente é possível atestar quem é responsável pelo desenvolvimento de um certo incremento. Na existência de dois contratados, muito dificilmente seria possível individualizar a culpa ou mesmo verificar o nexo de causalidade. Ainda mais complexo seria o caso de funcionários do contratante fazerem parte das equipes de desenvolvimento. Em geral o contratante busca uma empresa especializada porque ou não dispõe de mão de obra suficiente ou a sua mão de obra não é suficientemente qualificada. No último caso faz-se difícil a assunção da responsabilidade por eventual defeito do produto se evidente que o nível da prestação do contratado em muito supera à qualidade da prestação dos funcionários do contratante.

Ademais, ainda que seja praxe modularizar o programa e tentar reduzir as chamadas *dependencies*<sup>76</sup>, de regra o *software* funciona como um todo, ou seja, a funcionalidade que o contratante A está desenvolvendo agora pode vir a causar defeitos na funcionalidade ou incremento que desenvolveu contratante B. Portanto, faz-se difícil delimitar a prestação, em especial quando se está em face de um produto que não foi desde o início desenvolvido pelo mesmo contratado. Muitas vezes o contratado assume o desenvolvimento de uma aplicação, saneia-a e desenvolve outras funcionalidades sobre esta<sup>77</sup>.

Outra questão estaria relacionada ao fato de o programa sob medida utilizar componentes ou interfaces com programas estandarizados, mais especificamente, de outras empresas que não foram desenvolvidos exclusivamente para o contratante. Nesse caso, o defeito pode ser originado no *software* estandarizado e não no *software* desenvolvido sob medida<sup>78</sup>. Nesta situação, poderia ser o defeito de responsabilidade do contrato se a decisão de utilização do

---

76 Quando há muita e forte interligação entre pacotes diferentes cujas funcionalidades podem ser inteiramente distintas.

77 Cfr. neste contexto a distinção entre *greenfield* e *brownfield engineering* em WADE (2018).

78 RÖTTGEN (2017), pp. 191-192.

referido programa seja de autoria do contratado e se o *software* estandardizado utilizado seja conhecidamente defeituoso ou incompatível para o fim a que se destina<sup>79</sup>.

Por último, com a utilização da integração e da entrega contínuas, outrora mencionadas, está se em face da incorporação e entrega contínua das novas funcionalidades, que passam diretamente a poder ser utilizadas pelo usuário final do *software*. Nesse sentido, faz-se complicada a identificação de quando podem ter surgido os defeitos em questão, isto é, dificultando a identificação de prazos prescritivos.

## 6. Considerações finais

Diante destas reflexões observa-se estar na presença de relações de grande complexidade, a qual se inicia na origem técnica da matéria e desencadeia efeitos jurídicos relevantes. Sintetiza-se ora as principais constatações aqui expostas.

Os contratos de desenvolvimento e manutenção de *software*, ainda que não sejam contratos originariamente duradouros, assumem em muitas constelações esta faceta, não raro sendo a fixação da duração por tempo indeterminado em contratos-quadro a serem especificados em diversos contratos de curta ou média duração. Neste caso, o caráter duradouro também decorre do cunho material da relação, por ser o programa objeto de constantes alterações, de regra, não se esgotando o seu desenvolvimento.

Os *frameworks* de gerenciamento e organização dos projetos, sejam clássicos ou ágeis, têm influências sobre a qualificação destes contratos e das obrigações neles previstas. Ao passo que um projeto de acordo com o modelo em cascata, com algumas exceções, muito se aproxima de um contrato de empreitada, um projeto de acordo com o *scrum* poderia estar mais próximo de um contrato de prestação de serviços inominado (e por consequência do contrato de mandato), uma vez que muito ausente uma longa fase pré-contratual com a extensa definição de requisitos e especificações e com a possibilidade de grande ingerência do contratante na execução do contrato. Pontuou-se ainda as dificuldades de satisfazer alguns requisitos de um contrato de empreitada, como a questão da aceitação por parte do contratante diante de um programa que constantemente está sendo atualizado e entregue, e ainda muitas vezes a impossibilidade de o contratante fazer uma fiscalização adequada deste programa.

---

79 RÖTTGEN (2017), p. 192.

Os contratos de desenvolvimento e manutenção de *software*: o caráter duradouro e as influências do *scrum* sobre a tipologia, a cessação e a responsabilidade contratuais \ **Vinicius Aquini Gonçalves**

Face à qualificação complexa, muitas outras constelações contratuais seriam possíveis. De um lado, podendo-se aproximar o contrato a um contrato de empreitada, com obrigações de resultado, em especial quando utilizado um modelo em cascata e sendo o contratado responsável substancialmente pelo desenvolvimento. De outro lado, um contrato mais genérico de prestação de serviços e ancorado no contrato de mandato seria mais adequado para os casos em que o contratante esteja muito fortemente envolvido no desenvolvimento do programa, até mesmo compondo as equipes de desenvolvimento. Frisou-se ainda a possibilidade de diferenciar a tutela das obrigações dos papéis de um projeto de acordo com o *scrum*, bem como de incorporar ou não o caráter duradouro ao contrato.

Destas constelações também se derivariam efeitos para a cessação contratual, sendo no caso de um caráter duradouro a cessação por manifestação unilateral de uma das partes possível, isto é, por denúncia. Evidenciou-se as questões da caducidade de contratos por tempo determinado e da sua ativação decorrente da oposição em contratos de renovação automática. No tocante à resolução referiu-se à possibilidade de transpor por analogia os fundamentos do Art. 30 do DL 178/86 sobre a resolução em razão de circunstâncias endógenas ou exógenas ao contrato.

No tocante à responsabilidade contratual, esta de cunho dispositivo, dependendo da constelação contratual convencionada poderia tanto relegar ao contratado ou ao contratante o risco do negócio. Assim sendo, possível o contratado ser responsável pela entrega de um programa de acordo com as especificações do contratante, podendo os vícios ensejar o direito ao saneamento, à minoração de preços, bem como perdas e danos, ou somente por agir com a devida diligência profissional e de acordo com as instruções do contratante. No último caso, havendo grande envolvimento do contratante na execução do projeto, este assumiria o risco do negócio.

Concluindo, diante da ausência de um tipo contratual específico, resta grande discricionariedade às partes para configurar a relação contratual de modo que melhor corresponda à maneira como a relação fática será estabelecida. Assim sendo, procurou-se aqui ponderar as vantagens e desvantagens do enquadramento nestes dois tipos contratuais comumente utilizados, a fim de prover bases para a discussão acadêmica e para análise de riscos ante a adoção de um modelo contratual para negócios jurídicos de desenvolvimento e manutenção de programas, na prática. Devido à rara judicialização destas matérias, tampouco há uma jurisprudência vasta que oriente estas discussões, restando assim a desejar que algumas destas matérias sejam no futuro mais intensamente debatidas nos tribunais ou ainda que haja uma evolução legislativa que venha a atender as especificidades deste negócio jurídico.

## 7. Referências

### Bibliográficas

- ALMEIDA, Carlos Ferreira de, 2016, *Contratos II: Conteúdo. Contratos de Troca*, 4.<sup>a</sup> ed., Almedina, Coimbra.
- ALMEIDA LEITE, Rui Pedro Rodrigues, 2016, *Do Conceito de Obra no Contrato de Empreitada: Reflexões Doutrinárias e Jurisprudenciais* (Dissertação de Mestrado) – Faculdade de Direito da Universidade Católica Portuguesa, Porto, in <https://repositorio.ucp.pt/bitstream/10400.14/21975/1/Rui%20Pedro%20Leite%20-%20Dissertacao%20de%20Mestrado%20-%202016.pdf> (20.11.2020).
- AUER-REINSDORFF, Astrid; CONRAD, Isabell (Ed.), 2019, *Handbuch IT- und Datenschutzrecht*. 3.<sup>a</sup> ed., C. H. Beck, Munique.
- BAMBERGER, Heinz Georg, *et alii*. (Ed.), 2020, *BeckOK BGB*. 53.<sup>a</sup> ed., C. H. Beck, Munique.
- BATISTA FERREIRA, Manuel Ribeiro, 2015, *Noção de Obra no Contrato de Empreitada: Contributos da Doutrina e da Jurisprudência* (Dissertação de Mestrado) – Escola Superior de Tecnologia e Gestão, Politécnico do Porto, Porto, in <https://core.ac.uk/download/pdf/302867842.pdf> (20.11.2020).
- BORTZ, Christoffer, 2018, Auslegung und Gestaltung agiler Projektverträge: Vertragsrechtliche Analyse unter Berücksichtigung der verschiedenen Rollen in Scrum-Projekten, *Multimedia und Recht*, pp. 287-292.
- BRUSH, Kate; HAUGHN, Matthew, 2020, *Definition of IT Operations*, in <https://searchitoperations.techtarget.com/definition/IT-operations> (18.04.2020).
- COELHO, Francisco Pereira, 2017, “Cessaçã dos contratos duradouros: regime específico e contrato de agência”, in *Actas do Colóquio Distribuição Comercial nos 30 anos da Lei do Contrato de Agência: 23 setembro 2016*, António Pinto Monteiro (Ed.), Coimbra, pp. 225-244.
- FUCHS, Anke, *et alii*, 2012, Agile Programmierung – Neue Herausforderungen für das Softwarevertragsrecht?, *Multimedia und Recht*, pp. 427-433.
- HEISE, Dietmar; FRIEDL, André, 2019, Flexible (“agile”) Zusammenarbeit zwischen Unternehmen versus illegale Arbeitnehmerüberlassung – Ende von Scrum?, *Neue Zeitschrift für Arbeitsrecht*, pp. 129-136.
- HOEREN, Thomas; PINELLI, Stefan, 2018, Agile Programmierung: Einführung und aktuelle rechtliche Herausforderungen, *Multimedia und Recht*, pp. 199-204.
- KARGER, Michael, 2004, “Kooperation” bei komplexer Softwareentwicklung, *IT-Rechtsberater*, pp. 208-210.
- KARSTEN, Timo, *et alii*, 2016, The Future of IT Sourcing in Germany. Changes con-

Os contratos de desenvolvimento e manutenção de *software*: o carácter duradouro e as influências do scrum sobre a tipologia, a cessação e a responsabilidade contratuais \ **Vinicius Aquini Gonçalves**

- cerning labour law and IT-contracts, *Computer Law Review International*, vol. 17, n. 6, pp. 172-179.
- KOCH, Frank A., 2010, Agile Softwareentwicklung – Dokumentation, Qualitätssicherung und Kundenmitwirkung, *IT-Rechtsberater*, pp. 114-119.
- KOCH, Moritz Philipp; KUNZMANN, Luise; MÜLLER, Norman, 2019, Agile Softwareentwicklung und EVB-IT: Einordnung und Gestaltung agiler Softwareentwicklungsverträge bei öffentlichen Aufträgen, *Multimedia und Recht*, pp. 707-711.
- KREMER, Sascha, 2010, Gestaltung von Verträgen für die agile Softwareerstellung, *IT-Rechtsberater*, pp. 283-289.
- LITSCHEN, Kai; YACOUBI, Ilies, 2017, Arbeitnehmerüberlassung und agile Prozess- und Organisationsmethoden, *Neue Zeitschrift für Arbeitsrecht*, pp. 484-489.
- LUTZ, Holger; BACH, Simone, 2017, Agile Softwareentwicklung – Werkvertrag oder doch Dienstvertrag?, *Betriebsberater*, pp. 3016-3020.
- MARTINEZ, Pedro Romano, 2015, *Da cessação do contrato*, 3.<sup>a</sup> ed., Almedina, Coimbra.
- MENEZES LEITÃO, Luís Manuel Teles de, 2019, *Direito das Obrigações – Vol. III*, 13.<sup>a</sup> ed., Almedina, Coimbra.
- MICROSOFT AZURE (2020), *What is DevOps?*, in <https://azure.microsoft.com/en-us/overview/what-is-devops/> (16.04.2020).
- MONTEIRO, António Pinto, 2009, *Contratos de distribuição comercial*, 1.<sup>a</sup> ed., Almedina, Coimbra.
- PEREIRA, Alexandre Dias, 1999, “Programas de computador, sistemas informáticos e comunicações electrónicas: alguns aspectos jurídico-contratuais”, *Revista da Ordem dos Advogados*, vol. III, Ano 59, pp. 915-1000.
- PEREIRA, Alexandre Dias, 2011, *Das Licenças de Software e de Bases de Dados*, in [http://repositorio.uportu.pt/xmlui/bitstream/handle/11328/1265/01\\_ALEXANDRE-PEREIRA.pdf?sequence=1&isAllowed=y](http://repositorio.uportu.pt/xmlui/bitstream/handle/11328/1265/01_ALEXANDRE-PEREIRA.pdf?sequence=1&isAllowed=y) (29.12.2020).
- RED HAT (2020), *Integração e entrega contínuas: pipeline CI/CD*, in <https://www.redhat.com/pt-br/topics/devops/what-is-ci-cd> (16.04.2020).
- RÖTTGEN, Klaus, 2017, Schadensersatzhaftung für Softwaremängel, *IT-Rechtsberater*, pp. 191-194.
- SCHNEIDER, Jochen, 2010, “Neue” IT-Projektmethoden und “altes” Vertragsrecht: Herausforderungen für die Gestaltung von Erstellungs- und Anpassungsverträgen, *IT-Rechtsberater*, pp. 18-23.
- SCHUSTER, Fabian, 2019, Abnahme, Gewährleistung & Schadensersatz bei Software-Werkverträgen, *Computer und Recht*, pp. 345-352.
- SCHWABER, Ken; SUTHERLAND, Jeff, 2017, *The Scrum Guide*, in <https://www.scrumguides.org> (08.03.2020).
- SÖBBING, Thomas, 2010, Die rechtliche Betrachtung von IT-Projekten: Rechtliche

- Fragestellungen in den unterschiedlichen Phasen eines IT-Projekts, *Multimedia und Recht*, pp. 222-227.
- SÖBBING, Thomas, 2014, Agile Projekte in der IT-rechtlichen Praxis, *IT-Rechtsberater*, pp. 214-219.
- VON SCHENCK, Sophie, 2019, Gestaltung agiler Softwareverträge: Kommentierter Vertragsentwurf, *Multimedia und Recht*, pp. 139-142.
- WADE, John, 2018, *Greenfield vs. Brownfield Software Development. What's the Difference?*, Synoptek, in <https://synoptek.com/insights/it-blogs/greenfield-vs-brownfield-software-development/> (07.04.2020).
- WITTE, Andreas, 2010, Agiles Programmieren und §651 BGB, *IT-Rechtsberater*, pp. 44-47.

### Jurisprudenciais

- ACÓRDÃO DO STJ DE 03.11.1983 (SANTOS SILVEIRA), *Boletim do Ministério da Justiça*, n.º 331, pp. 489-503.
- ACÓRDÃO DO STJ DE 05.07.1994 (CARLOS CALDAS), JSTJ00024721, in <http://www.dgsi.pt/jstj.nsf/954f0ce6ad9dd8b980256b5f003fa814/cc8527c216e329c-6802568fc003a884a?OpenDocument&Highlight=0,085411>.
- ACÓRDÃO DO STJ DE 21.11.2006 (SEBASTIÃO PÓVOAS), JSTJ000, in <http://www.dgsi.pt/jstj.nsf/954f0ce6ad9dd8b980256b5f003fa814/0a7610adc034c2b-f80257241004fceff?OpenDocument>.
- ACÓRDÃO DO TR DE LISBOA DE 18.09.2008 (FARINHA ALVES), TRL\_4444/2008-2, in [http://bdjur.almedina.net/juris.php?field=doc\\_id&value=89175](http://bdjur.almedina.net/juris.php?field=doc_id&value=89175).
- ACÓRDÃO TR DE LISBOA DE 08.09.2015 (MARIA DO ROSÁRIO MORGADO), 89359/10.7YI-PRT.L1-7, in <https://tribunal-relacao.vlex.pt/vid/728473229>.
- ACÓRDÃO TR DE LISBOA DE 04.07.2019 (ANTÓNIO SANTOS), 9584/16.0T8LRS.L1-6, in <http://www.dgsi.pt/jtrl.nsf/33182fc732316039802565fa00497eec/89dde-2ca3ff0d75a802584330037b788?OpenDocument>.

