

# 5

## Criteria for Code

[T]he genie may then be out of the bottle never to be put back in . . . criteria addressing the process of technology development – ‘rules of the game’ – should be a key part of our acceptability criteria.<sup>1</sup>

The previous chapter discussed criteria intended to govern the creation of legal rules, to ensure that they meet standards of legality and legitimacy regardless of their political content. This chapter mirrors that analysis, reviewing the existing literature on criteria for the use of code in regulating behaviour.

The literature on the regulation of and by technology is large, but analysis of what criteria can legitimate its design is very limited indeed.<sup>2</sup> Goldoni suggests that this is due to a scepticism developing in the decade following Reidenberg’s and Lessig’s work on *lex informatica* and ‘code as law’, respectively.<sup>3</sup> Ohm and Frankle have recently made a similar argument:

Too many scholars have interpreted Lessig as doing little more than issuing a license to imagine that anything is possible online, falling into a ‘science fiction trap’. Too rarely do they consider the process of *how* code ends up the way it does (let alone how regulators can make use of this process), leaving a significant void in the utility of this body of work.<sup>4</sup>

Ironically, the scepticism Goldoni refers to embodies aspects of the legalistic ideology discussed earlier in the book, where an unwillingness to consider extra-legal sources of normativity leads lawyers to retreat to their intellectual

<sup>1</sup> B-J Koops, ‘Criteria for normative technology: The acceptability of “code as law” in light of democratic and constitutional values’ in R Brownsword and K Yeung (eds), *Regulating Technologies: Legal Futures, Regulatory Frames and Technological Fixes* (Hart 2008) 166.

<sup>2</sup> M Goldoni, ‘The politics of code as law: Toward input reasons’ in J Reichel and AS Lind (eds), *Freedom of Expression, the Internet and Democracy* (Brill 2015) 123.

<sup>3</sup> *Ibid.* 117.

<sup>4</sup> P Ohm and J Frankle, ‘Desirable inefficiency’ (2019) 70 *Florida Law Review* 1, 23 (emphasis supplied).

bunkers, from where they can continue to view law as a separate enterprise, ‘fenced off’ from other concerns. This strengthens the instinctive belief that code is not – and should not be seen as – law, and that legal thinking should therefore not concern itself with it other than as a subject of legal regulation like any other. Noting this tendency, Brownsword has suggested that the domain of jurisprudence should be ‘redrawn’ to sensitise it to a ‘bigger regulatory picture’, including forms of a-legal normativity that are ‘at least as important as legal norms in the daily lives of people’.<sup>5</sup>

As Chapter 1 discussed,<sup>6</sup> the purpose of the analysis in this book is not to validate private enterprises as producers of law *per se*. Instead, its aim is to enquire as to how code which has normative effects can be legitimated, which is to say how it can embody effects, features, or affordances alongside its commercially purposive functionality that ameliorate the negative effects of computational legalism. The issue then is not one of the ‘legal-ness’ of code rules *per se*, that is of viewing them as a source of law, but rather the question of how the ‘non-law’ of code can, in spite of those negative effects, be produced in ways that are legitimate from the perspective of the law and of constitutional democracy.<sup>7</sup> A failure to do so leaves a significant and serious deficit in our understanding of how citizens, as end-users, have their behaviour enabled and constrained by the often unintelligible code created by unelected private enterprises.

The rest of this chapter considers the current literature on normative criteria for the production of code. Following the analysis in the previous chapter, it is possible to discern a broad separation between those arguments which focus on input criteria and those which focus on output criteria. First, I discuss briefly what these classifications mean in the context of code, and why input criteria deserve much greater focus, before summarising the most relevant contributions in the literature. We can then take stock before moving on to Part III of the book, where I build on the criteria discussed in this and the previous chapter, setting out a framework of affordances whose presence in code can serve to provide a baseline of legitimacy.

## 5.1 Input and Output Legitimacy in Code

We saw above how a norm’s legitimacy can be considered by focusing on its production and/or the effects it has in operation. Chapter 2 set out why, in

<sup>5</sup> R Brownsword, ‘In the year 2061: From law to technological management’ (2015) 7 *Law, Innovation and Technology* 1, 10–14, 30.

<sup>6</sup> See Section 1.4.

<sup>7</sup> Cf. J Waldron, ‘The concept and the rule of law’ (2008) 43 *Georgia Law Review* 1, 12, discussing how by characterising something as ‘law’ we ‘dignify it with a certain character’.

the computational context, the deontology of input legitimacy is necessary: the *ex ante* characteristics of computational legalism demonstrate that an *ex post* consequentialist perspective is not, on its own, sufficient to ameliorate those negative characteristics. When we move from the traditional legislative sphere into the computational context, though, things pivot somewhat. The focus on process is not just one of participation – indeed, the participatory aspect will in a great many cases be minimal, owing to the private spheres within which code artefacts are developed. Rather, the ‘input’ aspect shifts to have more of a temporal focus, where more granular design decisions about particular aspects of the code’s functionality are the focus of legitimation according to binding criteria. Their private production means they might not be the product of a participatory democratic process *per se*, but they are ‘input’ in the sense that they are crucial constituents of the products (outputs) of the design process that ultimately are responsible for the code’s effects in the world.

The distinction is a subtle but crucial one in the context of computational legalism: if we only assess a system according to its operation in the real world (we apply only criteria that assess output) then the production ship has already sailed, and the opportunity to amend the design to remedy any defects we discover may be limited or gone altogether. Furthermore, limiting our focus to operation assumes that assessments of output are capable of detecting all salient negative effects, which is of course far from guaranteed, especially owing to the opacity of code. The shift towards input criteria puts the focus on the design process, to ensure *ab initio* that certain design characteristics are in place that allow for better output assessments but simultaneously reduce the need for them, because the initial configuration of the system is more legitimate from the outset.

Within the sphere of privacy by design, Hartzog argues in favour of focusing on processual standards because ‘even certain risky designs can be tolerated so long as companies take the right steps to mitigate potential harm and ensure that debatable design decisions were justified’.<sup>8</sup> Thus, mandating certain processes can potentially mitigate risk through the requirement to consider, during the process of design, the extent to which the proposed code embodies the standards we wish to see in a legitimate normative order. This approach also has practical appeal in terms of reducing the expense and delay of having to reconfigure a design once *ex post* assessment uncovers that it does not meet one or more of the requirements. Because of the integrated

<sup>8</sup> W Hartzog, *Privacy’s Blueprint: The Battle to Control the Design of New Technologies* (Harvard University Press 2018) 179.

nature of software development processes, such ex post ‘patches’ are often less effective than approaches that take matters into consideration from the outset. As Luger and Golembewski note in the sphere of privacy by design,

[a]ddressing these concerns at the end of a design cycle leaves the creators of the system with little time or agency to manoeuvre, and leads to a situation where potential privacy problems are addressed – if at all – as afterthoughts, with inelegant solutions and imperfect implementations bolted on to a mostly-complete design.<sup>9</sup>

Importing the distinction between input and output reasons into the computational sphere, Koops observes that

[i]nput legitimacy implies legitimacy through rules-of-the-game and the procedure followed, output legitimacy means that the result establishes legitimacy . . . [I]n the context of normative technology input legitimacy is a primary concern. Because technology is often irreversible – once it is developed and applied in society, it is hard to fundamentally remove it from society in those applications – the process of developing technology is a key focus when normativity is at stake. After all, it may well be too late when technology simply appears in society to ask whether it is acceptable to use this technology; quite often, the genie may then be out of the bottle never to be put back in . . . [C]riteria addressing the process of technology development – ‘rules of the game’ – should be a key part of our acceptability criteria.<sup>10</sup>

Input criteria are important because the characteristics of computational legalism militate against the effectiveness of ex post assessments of effects in the world (outputs). Goldoni argues along similar lines:

Given the nature and logic of architectural regulation, the emphasis on output legitimacy is misplaced for several reasons . . . The difficulty of reversing embedded code is often evident and makes it fundamental to focus on the procedure and the actors involved in the development of the technology. Second, given the opacity of architectural regulation, to be aware of how technology is directly or indirectly impacting upon agents’ behaviours may prove to be too difficult in many cases. Last but not least . . . the importance

<sup>9</sup> E Luger and M Golembewski, ‘Towards fostering compliance by design; drawing designers into the regulatory frame’ in M Taddeo and L Floridi (eds), *The Responsibilities of Online Service Providers* (Springer 2017) 296. See also L Diver and B Schafer, ‘Opening the black box: Petri nets and privacy by design’ (2017) 31 *International Review of Law, Computers & Technology* 68, 74–5.

<sup>10</sup> Koops (n 1) 166.

of default technology cannot be underestimated. What appears to be default in code is often taken as a natural and immutable fact.<sup>11</sup>

He concludes therefore that ‘input-based legitimacy should become the primary concern in choosing normative criteria’.<sup>12</sup> Moving from a focus on output (ex post) to input (ex ante) legitimacy is necessary if the public dimension involved in traditional rule-making is to be imported into the computational sphere, particularly when so much of the latter is privatised. Crucially, however, the latter does not replace the former – ex post measures remain necessary, to maintain a connection with institutional legal processes. Goldoni thus advocates for a shift from a ‘descriptive to a normative approach’ to code as law (recall that this is a reversal of the effects of computational legalism, where the normative becomes the descriptive<sup>13</sup>).

Goldoni categorises the literature between analyses of input and output criteria, noting a tendency towards the latter. This is perhaps to be expected, because observations of the real-world effects can be more easily subjected to an orthodox critique from a compliance perspective. The problem with this view is that it does not address directly those who produce the very code that is in question – it sustains the ‘fencing off’ of jurisprudential analysis from the object of that analysis. Lawyers continue to be viewed as ex post assessors of code, without acknowledging designers as the ex ante producers of it.

Ultimately, Goldoni suggests that two principles should govern code production: transparency and ‘publicness’.<sup>14</sup> The first suggests that rules embodied in code must be knowable in order that they can be observed and their creators held accountable, while the second implies that there must be opportunity for those subject to the rules to have a say in their creation.

In the remainder of this section I consider the literature on the question of normative criteria for code, following Goldoni in separating the works broadly into those who focus on substantive output criteria and those who focus on input criteria. Again, my argument is that the latter is the more appropriate focus for criteria that can assist in guiding the design of digital artefacts. While ex post assessments are also important, we have seen how

<sup>11</sup> Goldoni (n 2) 128.

<sup>12</sup> Ibid.

<sup>13</sup> See Chapter 3. Bańkowski uses this exact formulation: ‘What we see then, is how the normative has become the descriptive. This gives us an example of rule following which has the machine-like quality of heteronomy.’ See Z Bańkowski, ‘Don’t think about it: Legalism and legality’ in MM Karlsson, Ó Páll Jónsson and EM Brynjarsdóttir (eds), *Rechtstheorie: Zeitschrift für Logik, Methodenlehre, Kybernetik und Soziologie des Rechts* (Duncker & Humblot 1993) 56.

<sup>14</sup> Goldoni (n 2) 128–9.

computational legalism tends towards obfuscation, and thus the ability to carry out the assessments is itself contingent on ex ante design decisions that ensure they are possible. This complementarity of input and output criteria is therefore a part of the framework I develop in the next chapter.

## 5.2 Output Legitimacy

### (a) Brownsword's 'Technological Management'

Brownsword's primary criterion for assessing techno-regulation is that of justification, which he characterises as a judgement on 'whether we are over-regulating or under-regulating'.<sup>15</sup> This has a bearing on the central theme of Brownsword's work more generally: the acceptability of techno-regulation assessed from the perspective of human rights and human dignity. He views the latter as a question of 'empowerment', which consists of three elements: 'that one's capacity for making one's own choices should be recognised; that the choices one freely makes should be respected; and that the need for a supportive context for autonomous decision-making (and action) should be appreciated and acted upon'.<sup>16</sup> This conception of dignity leads ultimately to the suggestion, in the computational context, that individuals always retain the choice not to follow the rule as inscribed in the artefact.

To encourage the development of 'moral community',<sup>17</sup> the individual should where appropriate be empowered to take moral rather than merely 'prudential' choices (that is, choices that are in her own interest).<sup>18</sup> Technological management is problematic not because it naturally favours a particular form of (a)moral reasoning, but because it has the capability of bypassing practical reason altogether,<sup>19</sup> effacing opportunities for either moral or prudential choice.<sup>20</sup> Without the opportunity to exercise such choice, the possibility of moral community falters through individuals being 'de-moralised', that is having their capacity for moral judgement corroded through the removal of

<sup>15</sup> R Brownsword, 'What the world needs now: Techno-regulation, human rights and human dignity' in R Brownsword (ed.), *Global Governance and the Quest for Justice*, vol. 4 (Hart 2004) 205.

<sup>16</sup> Ibid. 211.

<sup>17</sup> A concept characterised as a community that is built on publicly proclaimed principles that are open to review according to processes that are inclusive of its members. See R Brownsword, 'Lost in translation: Legality, regulatory margins, and technological management' (2011) 26 *Berkeley Technology Law Journal* 1321, 1335 *et seq.*

<sup>18</sup> Ibid. See also Brownsword, 'In the year 2061' (n 5) 32–3.

<sup>19</sup> R Brownsword, 'Code, control, and choice: Why east is east and west is west' (2005) 25 *Legal Studies* 1, 13.

<sup>20</sup> Brownsword, 'In the year 2061' (n 5) 34–5.

opportunities to exercise it.<sup>21</sup> The result is a blunting of sensitivity to social norms, and a breakdown in moral community.<sup>22</sup> Indeed, the very concept of morality might disappear altogether if the possibility of infringing rights (doing harm) is removed by techno-regulation.<sup>23</sup>

In his earlier work on techno-regulation, Brownsword maintains a focus on state regulators as its source, suggesting that any movements from what he terms normative regulation (that is, measures which invite compliance) towards non-normative regulation (measures that do not permit scope for choice) should be ventilated by means of a ‘regulatory margin’ that can ‘facilitate deliberation about, and review of, changes to the complexion of the regulatory environment’.<sup>24</sup> In later work he suggests that this must take place *ex ante* in order to ratify the use of technological management before it is rolled out. Failure to do so will result in the potentially unlegitimated use of code which, because of the efficiency with which it enforces rules (its immediacy), closes the gaps in enforcement that previously permitted civil disobedience and the resulting friction and conflict that can be a driver for positive social change.<sup>25</sup>

This overarching goal is welcome at a policy level but does not engage with the practices of producing the code that implements the techno-regulation. The essential concern is that we ought to be wary of decisions that might lead to unfettered use of code for regulation. Brownsword’s overarching goal of respect for human dignity, embodied in the preservation of the ability to reason practically and to exercise choice, is important. In later work he expands beyond the focus on dignity and moral community to consider more explicitly legal-theoretical ideas, for example Fuller’s principles of legality (his analysis of which is considered below). For him, Fuller’s characterisation of legality as involving a reciprocal relationship between the end-user and the

<sup>21</sup> Brownsword, ‘Code, control, and choice’ (n 19) 4. See also A Le Sueur, ‘Robot government: Automated decision-making and its implications for Parliament’ in A Horne and A Le Sueur (eds), *Parliament: Legislation and Accountability* (Hart 2016) 192–3, discussing the passage of the Social Security Act 1998 and the potential effects of the increased use of automation in public administrative decision-making.

<sup>22</sup> Brownsword, ‘Code, control, and choice’ (n 19) 19, quoting criminologist DJ Smith, ‘Changing situations and changing people’ in A von Hirsch, D Garland and A Wakefield (eds), *Ethical and Social Perspectives on Situational Crime Prevention* (Bloomsbury Publishing 2004). See also Brownsword, ‘Lost in translation’ (n 17), especially 1355–6.

<sup>23</sup> Brownsword, ‘What the world needs now’ (n 15) 231.

<sup>24</sup> Brownsword, ‘Lost in translation’ (n 17) 1351. At the level of design, this margin might be manifest in the digisprudential affordance of delay, discussed in the next chapter.

<sup>25</sup> Brownsword, ‘In the year 2061’ (n 5) 36–7. I discuss the role of this ‘agonism’ in democratic societies in the next chapter.

state is key to the latter's use of code, and therein lies his prescription for the 'regulatory margin' that can facilitate the participatory mechanisms that will legitimise such regulation.<sup>26</sup> Brownsword's earlier focus on the public regulation of citizens means that his analyses do not venture far beyond relationship (b) in Figure 1.1 in Chapter 1 (depicting the normative relationship between the state and the citizen/end-user<sup>27</sup>).

I return to Brownsword's more recent work on input legitimacy below, but for now three criteria can be identified for the proper application of 'techno-regulation'. These are (1) respect for individual dignity through the preservation of choice (and more choice is better), (2) reciprocity between the regulator and the regulatee in the designing of norms, and (3) the need for a delaying 'regulatory margin' that can facilitate this reciprocity.

*(b) Leenes's 'Techno-regulation'*

Leenes expands the concept of techno-regulation to include as producers of code the private sector as well as the state.<sup>28</sup> He maintains a focus on techno-regulation as regulation borne of identifiable legal sources, namely state legislation or private contracts. This focus sets his analysis apart from my own, although he does obliquely reference the kinds of extra-legal normativity I am concerned with:

In the case of techno-norms implementing contractual terms or deriving legal status from the law . . . the legal status of the norms embedded in the artifact and the legal effects of breaching the norms are clear. In other cases the norms may be legally null and void and hence not legally bind individuals, yet as long as the norms remain embedded in the technology they in fact do regulate behaviour: legitimacy and effectiveness may be disjoint in practice.<sup>29</sup>

The latter class of norms that Leenes refers to is of course the focus of digisprudence, although he does not say much more about it (later work, discussed below, does consider this aspect). Ultimately, for Leenes the key factor is transparency of the 'techno-norms' and the process by which they are arrived at. For him, in an ideal situation regulatees consider the norms promulgated by privately produced code to be legitimate, the latter being achieved by 'engaging this community in deliberate discourse' which 'requires a free flow

<sup>26</sup> Brownsword, 'Lost in translation' (n 17) 1363–4.

<sup>27</sup> See 'Normative Relationships in Code and Law' in Section 1.4.

<sup>28</sup> R Leenes, 'Framing techno-regulation: An exploration of state and non-state regulation by technology' (2011) 5 *Legisprudence* 143.

<sup>29</sup> *Ibid.* 168.



of unhindered vital information'.<sup>30</sup> This overlaps with Brownsword's regulatory margin and Goldoni's transparency and 'publicness' requirements. I will consider below the limitations of this kind of participation in the privatised design process.

(c) *Koops's 'Criteria for Normative Technology'*

Koops provides an overview of criteria to be considered when assessing what he calls 'normative technology'.<sup>31</sup> He notes many of the concerns we have already seen in the discussion of computational legalism, around the ability of code to establish new norms, the effect of translating textual norms into code, and the applicability of democratic and constitutional values even in the context of private sector code production.<sup>32</sup>

Koops's survey of the literature provides a useful, complex, overview of the criteria that are discussed. His analysis conflates or bundles concepts that I believe should be kept apart, however. For example, he classifies due process, legality, and 'checks and balances' all under the umbrella of the 'rule of law', and refers to them as substantive (as opposed to procedural or formal) criteria.<sup>33</sup> Similarly, 'transparency of rule-making', 'transparency of rules', and accountability are listed as 'secondary criteria', while 'rule of law' and 'democracy' are listed as primary.<sup>34</sup> It is not clear that these concepts are quite so easily distinguished or prioritised.<sup>35</sup> They are all contested of course, but without clearer theoretical delineation Koops's criteria are somewhat limited (a point he acknowledges, and something he suggests be left to further research).

This is reflected in Koops's 'pragmatic, bottom-up' approach, where his aim is to identify the criteria suggested by other scholars, as opposed to taking an alternative 'top-down . . . theory-based interpretation of law'<sup>36</sup> (another point of departure). Although he acknowledges the fundamental importance of procedural (input) legitimacy, Koops's analysis is explicitly concerned primarily with what he calls 'outcome justice', and the ex post assessment of specific technologies. The set of criteria he ultimately develops is intended as a heuristic for structuring the process, rather than a means of performing it. His fourth level of abstraction begins to push towards concrete practices,

<sup>30</sup> Ibid. 167.

<sup>31</sup> Koops (n 1).

<sup>32</sup> Ibid. 159–61.

<sup>33</sup> Ibid. 168.

<sup>34</sup> Ibid. 169.

<sup>35</sup> For a useful overview separating out these various concepts, see J Tasioulas, 'The rule of law' in J Tasioulas (ed.), *The Cambridge Companion to the Philosophy of Law* (Cambridge University Press 2019).

<sup>36</sup> Koops (n 1) 162.

particularly in his class of ‘secondary criteria’, where he includes for example review, audit, the possibility of choice, optimal default settings, and context-adaptability.<sup>37</sup> He notes that the ‘proof of the pudding is in the eating’,<sup>38</sup> suggesting that what matters is testing of the criteria against concrete technologies. This will never, he says, ‘be a straightforward or uncontested exercise. For one thing, several of the criteria are culture-dependent, in their interpretation (for example moral values and democracy) or in their importance (for example human rights and choice).’<sup>39</sup>

Like other authors, this is the *ex post* ‘output’ legitimacy that represents a kind of ‘thick’ version of legitimacy. As a result, the substantive aspects of, for example, human rights become part of the assessment, contributing to both the difficulties that Koops refers to and to the complexity of his criteria. I doubt whether such assessments of substantive legal requirements can ever be expected to be carried out by designers all across the private sector, particularly in light of the complexity and nuance of the law and the limited resources of companies (especially SMEs) who do not have dedicated legal departments with the expertise required to find the relevant law and interpret it on behalf of designers.

Koops’s perspective seems, as mentioned above, to privilege the position of the lawyer as code assessor, thus maintaining an inbuilt bias towards legalistic *ex post* assessment.<sup>40</sup> The gaps between lawyers and designers, and between a product’s design and runtime phases, are thus maintained rather than bridged. Koops also suggests that the list of criteria itself will require periodic reassessment,<sup>41</sup> but again this is made necessary because the focus is on substantive rather than formal or procedural legitimacy. The latter should be able to stand the test of time, as in Fuller’s internal morality of law, because a procedure that follows legitimising formal principles ought to underpin the making of all code-based norms regardless of their substantive content.<sup>42</sup> Indeed, in a constitutional democracy we might say that it is a prerequisite

<sup>37</sup> Ibid. 168.

<sup>38</sup> Ibid. 171.

<sup>39</sup> Ibid. 170.

<sup>40</sup> Goldoni observes that ‘[i]n a rather typical legalistic and formalist fashion, Koops would also have lawyers testing the set of normative criteria.’ See Goldoni (n 2) 127–8.

<sup>41</sup> Koops (n 1) 171.

<sup>42</sup> Although the legisprudential principle of temporality also requires periodic reassessment, this is of the substance of the rule and not the affordance of reassessment itself. The latter (that is, providing the ability to reassess) is timeless, while the justification for a rule may change over time or indeed disappear – it is the principle of temporality (and its concordant affordance) that allows for this to be determined.

of those rules being legitimate.<sup>43</sup> Furthermore, focusing on procedure also potentially simplifies the criteria that need to be applied, at least at this level, since (as we shall see) there are fewer of them, and they are more or less constant.

Koops finishes with an enjoiner to consider the question of ‘ambient law’, or the incorporation of legal norms and values of legality into technological infrastructure itself. This notion, developed alongside Hildebrandt,<sup>44</sup> is the precursor to the latter’s concept of ‘Legal Protection by Design’, discussed next.

### 5.3 Input Legitimacy

In contrast to Koops’s explicitly ‘bottom-up’ approach, Hildebrandt, Brownsword, and Asscher each take a legal-theoretical approach to assessing legitimacy, and in so doing they push the focus away from output legitimacy towards input, or production, legitimacy.

#### (a) *Hildebrandt’s ‘Legal Protection by Design’*

The concept of ‘Legal Protection by Design’, or ‘LPbD’, is very closely aligned with the spirit of disprudence. In earlier work Hildebrandt used the term ‘ambient law’,<sup>45</sup> arguing that we must ‘find ways to articulate the legal framework of democracy and the rule of law into the technological architecture it aims to regulate, creating what has been called “Ambient Law”’.<sup>46</sup>

Chapter 1 discussed the use of the term ‘by design’ to denote the separate concerns of substantive compliance with particular fields of law (most commonly data protection, as in privacy/data protection by design) and the more general and indeed fundamental<sup>47</sup> goal of achieving legal protection. It is therefore more of a philosophical project about the nature of law and its operation in and through computational architectures, rather than the application of substantive doctrine within the computational context.<sup>48</sup> Hildebrandt

<sup>43</sup> Cf. M Hildebrandt and B-J Koops, ‘The challenges of ambient law and legal protection in the profiling era’ (2010) 73 *The Modern Law Review* 428, 454.

<sup>44</sup> *Ibid.*

<sup>45</sup> M Hildebrandt, ‘A vision of ambient law’ in R Brownsword and K Yeung (eds), *Regulating Technologies: Legal Futures, Regulatory Frames and Technological Fixes* (Hart 2008).

<sup>46</sup> *Ibid.* 176.

<sup>47</sup> As Hildebrandt and Koops suggest, ‘the challenge of Ambient Law is altogether far more fundamental than transposing “legal” norms into “technical” architectures’. See Hildebrandt and Koops (n 43) 460.

<sup>48</sup> M Hildebrandt, ‘Legal protection by design: Objections and refutations’ (2011) 5 *Legisprudence* 223, 238–9. Recall the discussion of this crucial distinction in “Why Not “Compliance by Design”?” in Section 1.4.

suggests that LPbD as an umbrella concept is concerned with both aspects – on the one hand, technological normativity should comply with substantive law, and on the other it should be both resist-able and contestable in a traditional court of law.

As I have previously discussed, the first requirement (substantive compliance) is not the focus of the present analysis. The second and third requirements point to the design of an artefact, and what it enables the end-user to do: can she exercise choice, and can she contest the design in a court? For Hildebrandt, '[t]he "resistability" requirement rules out deterministic environments, and the contestability requirement rules out invisible regulation.'<sup>49</sup> The goal is that 'the exercise of . . . rights should not be obstructed by the intended or unintended effects of new technologies'.<sup>50</sup> In essence, then, there are for her two criteria for the non-doctrinal (input) aspects of LPbD, namely choice and transparency.<sup>51</sup> I have already set out in detail in Part I of the book how computational legalism creates the conditions she is arguing against. The challenge now is to move beyond them to suggest ways forward. Hildebrandt does not provide concrete suggestions on how these can be achieved, instead setting out the overarching goals of LPbD. She sounds a warning, too: '[d]eveloping a methodology for LPbD entails a vertiginous challenge to traditional doctrinal research methods within legal scholarship and to the scientific methods of computer science, requirements engineering and electronics.'<sup>52</sup> This challenge is precisely what Part III will begin to grapple with, building on the design theory set out in Chapter 2 to suggest ways that the second aspect of LPbD might be achieved. This answers Hildebrandt's second enjoiner that we

should always include attention to the 'resistability' and contestability of the ensuing normativity, and should always involve testing how the configuration or design of the affordances can best serve the goals of justice, legal certainty and purposiveness.<sup>53</sup>

As we saw earlier, these latter three elements of justice, legal certainty, and purposiveness are drawn from Radbruch's antinomian theory of law.<sup>54</sup> The

<sup>49</sup> M Hildebrandt, *Smart Technologies and the End(s) of Law: Novel Entanglements of Law and Technology* (Edward Elgar Publishing 2015) 218.

<sup>50</sup> Hildebrandt, 'Legal protection by design' (n 48) 240.

<sup>51</sup> See also Hildebrandt and Koops (n 43) 456.

<sup>52</sup> Hildebrandt, *Smart Technologies* (n 49) 218.

<sup>53</sup> *Ibid.*

<sup>54</sup> G Radbruch, 'Legal philosophy' in K Wilk (ed.), *The Legal Philosophies of Lask, Radbruch, and Dabin* (Harvard University Press 1950) 111–12.

implication here is that the focus is on the design stage, where the affordances of the product are developed and where it can be considered whether or not they meet both the product's commercial requirements and those of legitimacy so conceived.

More generally, LPbD requires that the design of a product's 'commercial' affordances (what makes it attractive or useful to the end-user) must take account of its legal affordances; in (dis)affording particular behaviours for the end-user, the code must at all times permit the operation of the ideals of legality, which means the possibility of the end-user (1) observing the rules to which the system is subjecting her, (2) exercising choice as to which rules apply, and (3) contesting those rules in court.<sup>55</sup> Hildebrandt's analysis thus concerns input criteria, even though the focus is about the end-user having the ability to exercise her rights *ex post*. Her discussion of affordance and the 'designing in' of mechanisms to facilitate LPbD is inherently concerned with input criteria and the requirement that the design process reflect those *ex ante* requirements; if that is achieved then the *ex post* operation of the system will by definition embody the procedural, if not necessarily the substantive, aspects of output legitimacy (*ceteris paribus*).

### *(b) Applying Fuller to Code*

Both Brownsword and Asscher have considered how to adapt or apply Fuller's principles of legality in the context of code.

#### *Brownsword*

In recent work Brownsword has focused more on conventional legal theories with respect to code, demonstrating a pluralist perspective that is sensitive to the private production of code and its capability to 'compete with or complement, or simply supersede Hartian legal norms'.<sup>56</sup> For him, the principles of legality are an example of 'cosmopolitan values' that normatively bind all regulators, regardless of the substantive content of the regulations they promulgate.<sup>57</sup>

Brownsword appears throughout to maintain an ontological separation between the 'rule' or decision which animates the use of a particular code measure, and the substantive effect of the measure itself (this is made explicit in his treatment of the second principle, as I will explain below).

<sup>55</sup> M Hildebrandt, 'Law as an affordance: The devil is in the vanishing point(s)' (2017) 4 *Critical Analysis of Law* 116, 122.

<sup>56</sup> Brownsword, 'In the year 2061' (n 5) 10–14, 19.

<sup>57</sup> R Brownsword, 'Technological management and the rule of law' (2016) 8 *Law, Innovation and Technology* 100, 113.

This creates a distance between his analysis and the materiality of code, and therefore the design questions (what an artefact (dis)affords, and how it mediates reality for the end-user) that, as I explore in the next chapter, some of the Fullerian principles can usefully point towards. Nevertheless, alongside Asscher, Brownsword's is one of only two analyses of code from the perspective of Fuller's principles that I am aware of. The discussion below follows Brownsword's ordering of the principles.<sup>58</sup>

**Promulgation of rules (Fuller's second principle).** Brownsword claims that code environments are not governed by rules *per se* (and therefore there can be no operation of the first Fullerian principle, which states that there must be rules). Thus, for him, the second principle is converted into a requirement of transparency vis-à-vis the proposed use of technological management (that is, code).<sup>59</sup> The result is that regulatees should be given 'a fair opportunity to participate in the processes that will determine whether such a use is authorised',<sup>60</sup> the idea being that 'the purpose of promulgation is to invite public debate about the use of [code] measures'.<sup>61</sup> As mentioned above, whether this is workable in the commercial contexts where digital artefacts are designed is questionable, not least given the lack of incentives designers have to consult end-users or the communities who will be affected by their code. Brownsword aims at transparency of intent rather than actual technical transparency, but as I will discuss later, the distinction – and gap – between the two is problematic. At any rate, while the role of participation in the design process may be a desirable one, it is at most complementary to digisprudence, because it speaks to either (1) organisational processes (Brownsword's focus on the decisions to use code, rather than the code itself), or (2) the substantive functionality that makes the code attractive to a given class of end-user, at which point the question has moved beyond the 'constitutional' standards of legitimacy that should be present in all digital artefacts regardless of their commercial purpose. (This is a theme I return to below.)

**Rules should be prospective, not retrospective (third principle).** Although it is possible for retrospective acts in technologically managed environments (Brownsword gives the examples of database records being deleted, or contractual provisions being altered), he suggests that in general changes to the environment are prospective and therefore technological management

<sup>58</sup> Set out in LL Fuller, *The Morality of Law* (Yale University Press 1977) chapter 2. I discussed Fuller's principles within the legal context in the previous chapter.

<sup>59</sup> Brownsword, 'Technological management and the rule of law' (n 57) 117.

<sup>60</sup> *Ibid.*

<sup>61</sup> *Ibid.* 118.

does not introduce any new risk of ‘unfair retrospective penalisation of conduct’.<sup>62</sup>

**Rules should not require the impossible (sixth principle).** Brownsword’s discussion here focuses on the notional mental state of the regulatee, and how various legal systems deal with criminal attempts that are frustrated because of their impossibility.<sup>63</sup> Here, though, his focus shifts to the subjective position of the regulatee, rather than the legitimacy of the technological management measure, and so his analysis of this principle is not strictly relevant here.

**Rules should be clear (fourth principle).** The channelling of regulatees’ behaviour ‘should be done with less friction and confusion where the regulatory signal is clearly and decisively transmitted’.<sup>64</sup> Regulatees ought to have it communicated clearly that their conduct will be limited in some way by a technological measure.

**Rules should be relatively constant (seventh principle).** Brownsword suggests that frequent changes in what an application of technological management permits and denies, either because of malfunction or because of a deliberate change to the ‘regulatory coding’, might invite the uncertainty in regulatees that the principle aims to guard against. He warns against causing confusion to regulatees, caused by frequent code changes, resulting in their contravention of the ‘terms’ of the system and thus the levying of what are therefore unfair penalties because of a lack of constancy.<sup>65</sup>

**Rules should not be contradictory (fifth principle).** In the technological management context this principle should be ‘consistent in allowing or disallowing a certain “act”’.<sup>66</sup> This would appear more or less to match his proposal for the seventh principle, discussed above. He suggests further that where the system permits or renders possible a particular act, the regulatee should be given the benefit of an assumption against levying a penalty where it was the ‘fault’ of the system that what should have been a prohibited act (presumably owing to some other legal requirement) was in fact made possible.<sup>67</sup>

**The practical administration of rules must match their content (eighth principle).** Again, Brownsword maintains the ontological separation between code and the ‘offline’ rules that sit ‘behind’ the technological measure and animate its use. This is perhaps necessary for this particular principle.

<sup>62</sup> Ibid. 120.

<sup>63</sup> Ibid. 120–2.

<sup>64</sup> Ibid. 122.

<sup>65</sup> Ibid. 123.

<sup>66</sup> Ibid. 124.

<sup>67</sup> Ibid.

Here, his focus is on the translation of rules into code, and it is there where his concern over congruence arises: whether the rule as stated (or written) is properly reflected in the technological management measure.<sup>68</sup> This is the well-known problem of compliance by design discussed in Chapter 1, and also of the translation of “law in the books” to “law in other technologies”.<sup>69</sup>

**Rules should be general (first principle).** Here Brownsword shifts focus onto the question of algorithmic profiling, whereby the technological management system can feasibly have as many bespoke rules as there are regulatees subject to it.<sup>70</sup> Whether or not the code in question is ‘data-driven’, the concept of generality is relevant here. For example, end-users can believe themselves to be having the same experience as one another, when in fact this is not the case (an evocative example is the Facebook ‘emotional contagion’ experiment, an example of so-called ‘A/B testing’ discussed in Chapter 2<sup>71</sup>). Another example is when designers release ‘alpha’ and ‘beta’ versions of software, where end-users elect to access new features in a system before they are fully completed and ready for widespread distribution. In some cases, updates are released that fragment the uniformity of the code across the userbase.

Brownsword summarises his understanding of Fuller’s principles in the context of technological management as requiring ‘openness, or transparency, in authorising the use of measures of technological management for particular regulatory purposes, supported by ideals of fairness and due process’.<sup>72</sup> As I previously mentioned, this focus on authorisation maintains an ontological separation between the policy animating the use of technological management and the code that actually implements the normativity. The focus too on *ex ante* deliberation, the ‘regulatory margin’, is sensible from the perspective of large public regulatory bodies but is, I think, less plausible in the context of small commercial enterprises creating low-margin digital artefacts. The suggestion that such firms submit to ‘special procedures possibly akin to applications for planning permission’<sup>73</sup> seems unlikely to receive much purchase given the fecundity of the Internet as a generative platform and the ease with which almost anyone can get started creating code that has normative effect. Where Brownsword does discuss the private sector, he sets up

<sup>68</sup> Ibid. 124–5.

<sup>69</sup> Hildebrandt and Koops (n 43) 452, *et seq.*

<sup>70</sup> Brownsword, ‘Technological management and the rule of law’ (n 57) 125–6.

<sup>71</sup> R Booth, ‘Facebook reveals news feed experiment to control emotions’ *The Guardian* (29 June 2014) <<https://www.theguardian.com/technology/2014/jun/29/facebook-users-emotions-news-feeds>> last accessed 4 March 2021.

<sup>72</sup> Brownsword, ‘Technological management and the rule of law’ (n 57) 127.

<sup>73</sup> Ibid.



a dichotomy between ex ante ‘approval and authorisation’ on the one hand, and ex post ‘challenge and review’ on the other. Where the ex ante measures are not present, the ex post measures are thus necessary.<sup>74</sup> This is a good starting point for thinking about the responsibilities of software designers and their employers, but on its own does not go far enough. As I have argued throughout, and as we saw from the discussion of computational legalism in Chapter 3, an either/or approach is insufficient; if we rely only on ex ante measures, we cannot account for emergent and/or unexpected regulatory effects, while if we rely only on ex post measures, there may be significant harm done that will not be detected in order for those processes to be invoked.

Brownsword’s application of Fuller boils down to the need for openness, transparency, and due process in the authorisation of the use of technological management, together with the requirement – a longstanding theme in his work – that the conditions for moral community be maintained. As I discussed earlier in the chapter, for him the latter conditions are provided where there is preservation of individual choice and the ability to make a moral decision.<sup>75</sup> Brownsword does not engage with the concrete materiality of design, beyond a passing reference to transparency ‘about how the particular technologies work’.<sup>76</sup> For present purposes, this observation is particularly apposite: ‘while it is certainly a *necessary* condition for the acceptability of a particular use of technological management that the underlying rule or policy is compatible with the Rule of Law, it might not be *sufficient*’.<sup>77</sup> Indeed, digisprudence views it as not sufficient, because it does not engage with the materiality of the design that actually implements the normativity. This will be discussed in the next chapter, but for now I turn to Asscher’s application of Fuller’s principles to code.

### *Asscher*

Asscher’s analysis is more closely focused on the idea of code *per se*, as opposed to Brownsword’s focus on the legitimacy of the rules operating behind the code. His approach in adapting Fuller’s principles is to pose numerous questions for the assessment of code.<sup>78</sup> First, is it transparent: can citizens discern the rules they are subject to, or, in computational terms, can we be sure of

<sup>74</sup> Ibid. 117.

<sup>75</sup> Ibid. 129–31.

<sup>76</sup> Ibid. 139.

<sup>77</sup> Ibid. (emphasis supplied).

<sup>78</sup> L Asscher, “Code” as law: Using Fuller to assess code rules’ in E Dommering and L Asscher (eds), *Coding Regulation: Essays on the Normative Role of Information Technology* (TMC Asser Press 2006).

what the code is doing, and is this what we expect to happen?<sup>79</sup> Second, is the code consistent, both in the temporal sense (that is, it is not updated arbitrarily), and in the sense of congruence both with other code rules and with legal rules? This speaks to the trust that end-users can have in the system. Third, is its provenance clear, that is can end-users identify who is responsible for its production? ('[C]an a sovereign be distinguished that can also be held accountable for the influence of the software?'<sup>80</sup>) Fourth, is autonomy respected through the preservation of the choice of whether or not to obey?<sup>81</sup> He distils these adaptations of Fuller's principles into the following 'checklist':

1. Can *rules* be distinguished in the code?
2. Can they be *understood*, i.e., is it understandable how code works and what it does? If so, are those rules transparent, are they accessible to the general public?
3. Can the rules be *trusted*, is there any guarantee that rules are not changed during the game? Are code rules *reliable* in the sense that they are predictable?
4. Is there a sovereign? An *authority* who makes the code rules?
5. Is there a choice? Can consumers/citizens choose not to obey the rules? Can consumers/citizens freely *choose* another system of law/code?<sup>82</sup>

If the answer to the first question is negative, the rest can be ignored (I discuss below why things are not quite so simple). Questions two and three are connected, while the fourth is a practical concern. Interestingly, Asscher connects the fifth question – whether the end-user retains choice – explicitly with the issue of competition (contrast this with Brownsword's framing of choice as a foundation for moral reasoning and community).<sup>83</sup> For Asscher, the questions are about restoring balance between code and law. This is connected with the traditional process of legislation and law-application, one element of which is the practice of balancing competing interests. Asscher suggests that his Fullerian analysis of code is apt to assess whether the balance of power has tipped away from institutional law, in favour of the 'code world', and thereby whether some kind of state intervention is required to restore it.<sup>84</sup>

<sup>79</sup> Ibid. 84.

<sup>80</sup> Ibid.

<sup>81</sup> Ibid. 84–5.

<sup>82</sup> Ibid. 85 (emphasis supplied).

<sup>83</sup> Ibid.

<sup>84</sup> Ibid. 85–6.

Asscher's criteria can be summarised as follows. When there are rules enforced by code, (1) the code must be transparent (understandable to those regulated by it), (2) the code must be trustworthy and reliable (it operates as expected, and is not changed arbitrarily), (3) the producers of the code must be identifiable, and (4) end-users must have the choice of whether or not to obey its rules. We will see in the next chapter how these considerations are adopted in the framework of digisprudential affordances.

In Chapter 3 I considered Asscher's suggestion that 'rules should not be confused with the technical commands within a certain computer language but must be understood on the conceptual level'.<sup>85</sup> As discussed there, the rules at the conceptual level that Asscher refers to (that is, rules *qua* technological normativity) are necessarily dependent, at the lower level, on 'technical commands within a certain computer language'. The issue of code-as-rules *per se* thus cannot be quite so easily dismissed. It is appropriate (and indeed necessary) to focus, at least to some extent, on what the code actually says and does. It may be that Asscher is implying that code does not present us with rules in any conventional legal-theoretical sense, and that insofar as there are 'subtle examples of the intertwined effects of legal policy and software effects'<sup>86</sup> our focus should nevertheless remain at the conceptual level. As I have argued throughout, however, the failure to engage with the normativities that code in fact generates is precisely what allows its illegitimacies to go undetected, given cover by the notion, at the 'conceptual level', that the code 'underneath' does what it purports to do, or implements the orthodox rule its designer says it does.

#### 5.4 Conclusion

To conclude, we can summarise the various contributions from the literature surveyed above. Brownsword and Koops are interested mostly in substantive assessments of code's real-world effects. Brownsword's earlier contributions focused on public regulation, arguing that 'techno-regulation' must maintain scope for individual choice in order that the possibility of moral decision-making, as a foundational element of moral community, is retained. Koops, apart from providing a useful interpretation of other scholars' criteria, focuses on ex post assessments of specific technologies as they operate in the world. The criteria he identifies are mixed, although he prioritises the substantive

<sup>85</sup> Ibid. 83. See 'Does Code Contain Rules *per se*?' in Section 3.2.

<sup>86</sup> Ibid. 87.

(output) criteria of human rights and ‘other moral values’ before the rule of law and democracy.

Hildebrandt, Asscher, and Brownsword (in his later work) include a focus on input criteria, the latter two applying Fuller’s principles to code. Hildebrandt’s ‘Legal Protection by Design’ requires transparency in the rules which technological normativity embodies, the ability of the end-user to exercise choice (that is, to resist the default configuration), and the possibility of contesting the rules in a court of law. Brownsword surveys Fuller from a rule of law perspective, but his ontological separation between code itself and the policy rule which animates its use means that his target of assessment is not the design of the code *per se*, but rather the decision of whether or not to use it. He requires transparency and due process in the authorisation of the use of code regulation, as well as the original requirement of the retention of the possibility of (moral) choice on the part of the individual. In contrast to Brownsword, Asscher maintains a closer focus on code *per se*. His distillation of the Fullerian principles requires that code be transparent, that it works as expected and is not changed arbitrarily, that its producers can be identified, and that end-users retain choice as to whether or not to obey its rules. I adopt various of these requirements in the framework of digisprudential affordances developed in the next chapter.

This chapter has surveyed the literature on normative criteria for code, noting certain gaps in existing analysis, particularly with respect to the private production of code (as opposed to state-sanctioned use of code as a regulator), the production of unintended normativity, and the focus on *ex post* assessments of an artefact’s operation instead of the *ex ante* design decisions which lead to those effects.

The analysis in this part of the book has sought to provide both a legal-theoretical view on legitimate normativity, and the state of the art in the literature on criteria for code as a normative order. As we saw in Part I, with code there is an inevitability about the initial configuration – once the decision to build something has been made, that something by definition embodies a set of initial commitments that necessarily privileges one configuration of normativity over all the other possibilities. Furthermore, as the discussion of input and output legitimacy showed, in the code realm we cannot simply await a determination by the courts of a particular (legal) issue; in code the ‘decision’ is by definition *ex ante*. Restating the central concern of the book, the composition of the system, chosen by the designer, has the potential to introduce significant path dependencies in how it regulates end-user behaviour. This is coupled with the plasticity of software, which empowers the designer to impose such regulation in a vast, near-infinite number of ways. We must therefore interrogate critically that initial configuration in

order to ensure its legitimacy from the outset. The next part of the book uses the understanding gleaned here both to develop a framework of design strategies that can help to ameliorate computational legalism, and then to suggest some concrete practices that can serve as a starting point for implementing the framework.

