

# **Towards a More Agile Government**

## **The Case for Rebooting Federal IT Procurement**

Benjamin J. Balter<sup>\*</sup>

Ben@Balter.com

<http://Ben.Balter.com>

41 Pub. Cont. L.J. 149

The Public Contract Law Journal, Fall 2011

---

<sup>\*</sup> Benjamin J. Balter is a J.D./M.B.A. candidate at the George Washington University and the Technology Editor of the *Public Contract Law Journal*.

## Table of Contents

|                                                                                                                                                                          |           |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------|
| <b>I. Background – Waterfall v. Agile .....</b>                                                                                                                          | <b>7</b>  |
| A. Waterfall Software Development Sees Modern Software Development as an Extension of its Traditional Engineering Roots. ....                                            | 7         |
| B. Agile Software Development Mitigates the Risk Inherent in Uncertainty by Approaching a Project as Multiple Independent Tasks. ....                                    | 8         |
| C. Modern Trends in Both Governance and Technology Push Federal Procurement Towards Agility.....                                                                         | 9         |
| <b>II. Despite its Advantages and Long-Standing History, the Rigidity of Traditional Software Development Bears Risks Disproportionate to its Returns. ....</b>          | <b>10</b> |
| A. The Rigidity of the Waterfall Process Brings Several Advantages to Federal IT Procurement. ....                                                                       | 11        |
| B. Waterfall Software Development Fails to Adequately Respond to the Ever-Changing Conditions that Make Up a Project’s Problem Space. ....                               | 11        |
| i. Both conceptually and practically, waterfall assumes that contracting agencies can accurately forecast a project’s requirements in advance. ....                      | 11        |
| ii. Waterfall development fails to adequately respond to change. ....                                                                                                    | 13        |
| iii. Waterfall development provides a return disproportionately less than its significant administrative cost.....                                                       | 14        |
| <b>III. Given its More Effective Response to Change, Agile Software Development Can Deliver Better Performing Projects More Quickly and Cheaply than Waterfall. ....</b> | <b>15</b> |
| A. The Ability to Rapidly Respond to Change is the Shibboleth of an Agile Project. ....                                                                                  | 15        |
| i. Agile is iterative, breaking long-term projects into multiple independent, manageable tasks. ....                                                                     | 15        |
| ii. Agile prioritizes communication and collaboration over documentation and structure. ....                                                                             | 16        |
| iii. Agile is adaptive rather than predictive. ....                                                                                                                      | 17        |
| B. By More Effectively Responding to Change, Agile Can Offer Agencies Greater Predictability and More Timely and Less Costly Project-Delivery. ....                      | 17        |

|                                                                                                                                                                           |           |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------|
| i. Agile development adapts to change. ....                                                                                                                               | 17        |
| ii. Agile development is more efficient than waterfall development. ....                                                                                                  | 18        |
| iii. Agile is inherently less costly than waterfall. ....                                                                                                                 | 18        |
| iv. Agile is predictable.....                                                                                                                                             | 19        |
| C. Despite its Advantages, Agile Methodologies Present Contracting Officers With Several Key Challenges.....                                                              | 19        |
| i. Agile methods are not well suited to meet government documentation requirements. ....                                                                                  | 19        |
| ii. Agile engenders a lack of specificity pre-award, rendering cost estimates difficult to calculate. ....                                                                | 20        |
| iii. Agile offers contracting agencies less risk-avoidance upfront. ....                                                                                                  | 20        |
| <b>IV. Current Regulations and Agency Practices Restrict Adoption of More Efficient Agile Methods.....</b>                                                                | <b>21</b> |
| A. The Structure of Traditional Government Contracting is Not Amenable to Agile Development Methods.....                                                                  | 21        |
| B. Government Procurement Embodies a Culture Averse to Agile. ....                                                                                                        | 21        |
| C. Current IT Procurement Methods are Not Amenable to Agile Development.....                                                                                              | 22        |
| i. Price directed (sealed bid).....                                                                                                                                       | 22        |
| ii. Competitive negotiations .....                                                                                                                                        | 23        |
| iii. Technology/Conceptual Negotiated Sourcing .....                                                                                                                      | 23        |
| iv. Engineering Selection under the Brooks Act.....                                                                                                                       | 23        |
| v. Modular Contracting .....                                                                                                                                              | 24        |
| <b>V. Significant Reform is Necessary to Allow Federal IT Procurement to Take Advantage of Industry Advances that Have Far Outpaced Laws Intended to Govern Them.....</b> | <b>25</b> |
| A. COs and COTRs Require Education Before Agile Can be Introduced to the Federal Contracting Environment.....                                                             | 25        |
| i. COs must learn best-practices in agile contracting.....                                                                                                                | 25        |
| ii. Agency technical contacts (COTRs) must learn best practices in agile project management.....                                                                          | 26        |

|                                                                                                                            |           |
|----------------------------------------------------------------------------------------------------------------------------|-----------|
| B. Existing Procurement Requirements Demand Extensive Documentation Up Front, Forcing Agencies into Agile Frameworks ..... | 26        |
| i. Engineering services, FAR Part 36.....                                                                                  | 27        |
| ii. Modular contracting, FAR Part 39.....                                                                                  | 27        |
| C. COs Lack Top-Down Support for IT Procurement Reform.....                                                                | 28        |
| <b>VI. Conclusion.....</b>                                                                                                 | <b>28</b> |

Like many government computer systems, the U.S. federal information technology (IT) procurement model is slow, outdated, and long overdue for a reboot.<sup>1</sup> As the largest single purchaser of code,<sup>2</sup> in fiscal year (FY) 2010 the Federal Government spent more than \$77.1 billion on IT procurement, and that number is projected to grow higher by the close of 2011.<sup>3</sup> This is not a recent trend. Over the past decade, federal IT spending has swelled nearly seventy percent, up from \$45.6 billion in 2001,<sup>4</sup> for a total bill of more than \$500 billion.<sup>5</sup> This growth is partially a result of the unfortunate fact that as few as nine percent of projects are delivered on budget and on time.<sup>6</sup> The Government Accountability Office (GAO) reports that roughly forty-eight percent of all IT projects must be rebaselined,<sup>7</sup> and of those rebaselined projects, fifty-one percent must be rebaselined a second time.<sup>8</sup>

Compounding the problem, end users fail entirely to use nearly forty-five percent of features procured and rarely use another nineteen percent of those features.<sup>9</sup> Thus, purchasing agencies ultimately utilize only about one-third of all features paid for by American tax dollars.<sup>10</sup> In the end, nearly forty-five percent of federally procured software features ultimately fail to meet the user's needs.<sup>11</sup> It is therefore no surprise that the Secretary of the Department of Defense (DoD) Robert Gates called federal IT procurement "baroque."<sup>12</sup> Too often IT procurement

---

<sup>1</sup> This holds true for both civilian and military procurement systems. *See generally* OFFICE OF THE UNDER SEC'Y OF DEF. FOR ACQUISITION, TECH., AND LOGISTICS, REPORT OF THE DEFENSE SCIENCE BOARD TASK FORCE ON DEPARTMENT OF DEFENSE POLICIES AND PROCEDURES FOR THE ACQUISITION OF INFORMATION TECHNOLOGY (2009) [hereinafter DOD ACQUISITION REPORT].

<sup>2</sup> Jay P. Kesan & Rajiv C. Shah, *Shaping Code*, 18 HARV. J.L. & TECH. 319, 373 (2005).

<sup>3</sup> *Trends*, IT DASHBOARD, [http://www.itdashboard.gov/export/trends\\_report](http://www.itdashboard.gov/export/trends_report) (last visited Sept. 6, 2011).

<sup>4</sup> *Id.*

<sup>5</sup> WHITE HOUSE FORUM ON MODERNIZING GOV'T, OVERVIEW AND NEXT STEPS 5 (2010).

<sup>6</sup> VICTOR SZALVAY, DANUBE TECHS., INC., AN INTRODUCTION TO AGILE SOFTWARE DEVELOPMENT 1 (2004), *available at* [http://www.danube.com/docs/Intro\\_to\\_Agile.pdf](http://www.danube.com/docs/Intro_to_Agile.pdf).

<sup>7</sup> DOD ACQUISITION REPORT, *supra* note 1, at 44. "Rebaselining" occurs when modifications are made to a project's baseline, i.e. its cost, schedule, and performance goals, to reflect changed development circumstances. U.S. GOV'T ACCOUNTABILITY OFFICE, GAO-08-925, INFORMATION TECHNOLOGY: AGENCIES NEED TO ESTABLISH COMPREHENSIVE POLICIES TO ADDRESS CHANGES TO PROJECTS' COSTS, SCHEDULE, AND PERFORMANCE GOALS 2, 13 (2008). Changes in requirements and objectives (scope creep) was the most commonly cited reason for rebaselining. *Id.* at 8.

<sup>8</sup> *Id.*

<sup>9</sup> Szalvay, *supra* note 6, at 8.

<sup>10</sup> *Id.*

<sup>11</sup> Gwanhoo Lee & Weidong Xia, *Toward Agile: An Integrated Analysis of Quantitative and Qualitative Field Data on Software Development Agility*, 34 MIS Q. 87, 88 (2010).

<sup>12</sup> Robert Gates, *A Balanced Strategy: Reprogramming the Pentagon for a New Age*, FOREIGN AFF., Jan./Feb. 2009, at 28, 34.

requirements are crafted with the input of neither end-users nor product developers.<sup>13</sup> As Office of Management Budget (OMB) Director Peter R. Orszag noted, federal IT projects cost more than they should, take longer than they should, and often fail to meet agency needs.<sup>14</sup> Today's federal regulations shackle government agencies to outdated project management practices and prevent them from harnessing the true power of IT innovations, which have far outpaced the laws that govern them.<sup>15</sup>

To better embrace innovation and respond to changing organizational needs, the Government must embrace a two-pronged approach involving both regulatory reform and top-down support for best-practices education to empower IT-procuring agencies to pursue more agile software development methods. By requiring that detailed specifications be outlined at the onset of the process, government procurement regulations encourage the less flexible, waterfall development techniques, rather than the more modern, agile development approaches used by the private sector today.<sup>16</sup> While most prior attempts to reform federal IT procurement focused solely on statutory changes,<sup>17</sup> this Note proposes more modern project management practices and argues for top-down reform on both a regulatory and a human level.

---

<sup>13</sup> See VIVEK KUNDRA, U.S. CHIEF INFO. OFFICER, WHITE HOUSE, 25 POINT IMPLEMENTATION PLAN TO REFORM FEDERAL INFORMATION TECHNOLOGY MANAGEMENT 17 (2010) (calling for "increased communication with industry" and "high functioning, 'cross-trained' program teams").

<sup>14</sup> See Memorandum from Peter R. Orszag, Dir. of Office of Mgmt. and Budget, Exec. Office of the President, to Heads of Exec. Dep'ts and Agencies 1 (June 28, 2010) [hereinafter Orszag Memorandum], available at [http://www.whitehouse.gov/sites/default/files/omb/assets/memoranda\\_2010/m-10-26.pdf](http://www.whitehouse.gov/sites/default/files/omb/assets/memoranda_2010/m-10-26.pdf).

<sup>15</sup> These inefficiencies are troubling, not only because they represent a significant financial cost to the taxpayer, but also because they undoubtedly represent a significant cost to the realization of agency goals. See STANLEY N. SHERMAN, GOVERNMENT PROCUREMENT MANAGEMENT 30 (1991). As President Barack Obama recently noted at the White House Forum on Modernizing Government, "[w]hen we waste billions of dollars, in part because our technology is out of date, that's billions of dollars we're not investing in better schools for our children, in tax relief for our small businesses, in creating jobs and funding research to spur the scientific breakthroughs and economic growth of this new century." *Attachment B: President's Remarks*, in WHITE HOUSE FORUM ON MODERNIZING GOVERNMENT: OVERVIEW AND NEXT STEPS 17, 18 (2010), available at [http://www.whitehouse.gov/sites/default/files/omb/assets/modernizing\\_government/ModernizingGovernmentOverview.pdf](http://www.whitehouse.gov/sites/default/files/omb/assets/modernizing_government/ModernizingGovernmentOverview.pdf).

<sup>16</sup> DOD ACQUISITION REPORT, *supra* note 1, at 16-17 (noting many large corporations have gained a significant advantage from using agile). See WHITE HOUSE FORUM ON MODERNIZING GOV'T, OVERVIEW AND NEXT STEPS 9 ("Federal IT projects are too often marked by milestones spaced too far apart."). See generally *infra* Parts III-IV.

<sup>17</sup> See Ralph C. Nash, Jr., *Solutions-Based Contracting: A Better Way To Buy Information Technology?*, 11 NASH & CIBINIC REP. ¶ 17, at 60 (Apr. 1997).

## I. Background – Waterfall v. Agile

Two primary schools of thought exist regarding software development practices: waterfall and agile software development. While today's contracting regulations may propel agencies to use waterfall development, changes in industry practice with regard to large IT projects demands a government push toward more agile methodologies.

### A. Waterfall Software Development Sees Modern Software Development as an Extension of its Traditional Engineering Roots.

The traditional approach, “waterfall” software development,<sup>18</sup> seeks all requirements and specifications up front, distinguishing design and analysis from construction and development.<sup>19</sup> Waterfall development techniques trace their roots to long-standing engineering fields such as architecture and electrical engineering.<sup>20</sup> In such fields, late changes are costly and rigid requirements are a practical necessity.<sup>21</sup> With the rise of software development in the latter half of the twentieth century, engineers applied the same methodology to developing computer software.

Despite their similar roots, traditional engineering is not a perfect metaphor for software development.<sup>22</sup> First, unlike traditional engineering fields, software development benefits from nearly negligible switching costs.<sup>23</sup> Second, the bulk of

---

<sup>18</sup> A more in-depth examination of the characteristics of waterfall development is discussed *infra* part II.

<sup>19</sup> Adrian Royce, *Agile in Government: Successful On-Time Delivery of Software* 1, in CONFERENCE PAPERS, 20TH AUSTRALIAN SOFTWARE ENGINEERING CONFERENCE (2009).

<sup>20</sup> Martin Fowler, *The New Methodology*, MARTIN FOWLER (Dec. 13, 2005), <http://www.martinfowler.com/articles/newMethodology.html>.

<sup>21</sup> For example, an architect draws up blueprints for a house and hands them off to a contractor for construction. Should the future homeowner decide after the fact that he wishes to deviate from his original plan, e.g., he prefers the kitchen be five feet wider, such a change would incur substantial time, labor, and material costs as floors are re-laid, pipes re-plumbed, and electrical outlets rewired accordingly.

<sup>22</sup> Initially the application of the traditional methodology to the emerging field was much more straightforward. Room-sized mainframe computers required users to painstakingly code instructions on physical cards, submit them to be processed in batches, and await the result. JAMES O. COPLIEN & GERTRUD BJØRNVIG, *LEAN ARCHITECTURE: FOR AGILE SOFTWARE DEVELOPMENT*, 218 (2010). Today, however, advances in technology and the personal computer have allowed the software development environment to embody an interactive process whereby software engineers can receive immediate feedback as to their software's success or failure. *Id.* at 218-19 (noting that originally high computing costs drove up the cost of mistakes such that “rigid disciplines ruled.” Today, mistakes are cheap.).

<sup>23</sup> Fowler, *supra* note 20. If a federal agency decides that the navigation menu on its new website should be five pixels larger, to make such a change would simply require modifying one or two values in the website's source code, an insubstantial labor cost in a large government contract.

most software development projects is design and system architecture, a process requiring creative and analytical, rather than technical talent.<sup>24</sup> In a world of widely adopted industry conventions, the creative necessities of a large-scale procurement project's information architecture often overshadow technical necessities.<sup>25</sup> Third, because software development regularly prioritizes designing ideal workflows and simplifying human interactions rather than embracing a formulaic science, such creative processes cannot easily be planned or predicted.<sup>26</sup> In light of these industry trends, modern software development has recently embraced a more integrated practice known as agile software development.<sup>27</sup>

### **B. Agile Software Development Mitigates the Risk Inherent in Uncertainty by Approaching a Project as Multiple Independent Tasks.**

Agile development adopts an incremental approach to software development, but it does not seek to build software systems in a sequential, step-wise manner. Instead it recognizes the reality that a future system owner may be unable to gather all requirements up front.<sup>28</sup> Even if this were possible, those requirements likely would change over time to reflect shifting technological and organizational priorities.<sup>29</sup> Thus the project is broken down into several smaller components.<sup>30</sup> Plans are continuously inspected and adapted to the empirical reality of the project through rapid prototyping.<sup>31</sup> As a practice widely used in the private sector, empirically agile development delivers better performing projects on time and under budget.<sup>32</sup>

---

<sup>24</sup> *Id.*

<sup>25</sup> Information architecture necessities can include the creation of user profiles (personas of hypothetical system users), use cases (an analysis of the tasks those users might perform), and wireframes (the visual layout of the system's interfaces). *Id.*

<sup>26</sup> See Fowler, *supra* note 20.

<sup>27</sup> Mark Lycett et al., *Migrating Agile Methods to Standardized Development Practice*, COMPUTER, June 2003, at 79, 79.

<sup>28</sup> Szalvay, *supra* note 6, at 2; GOVERNMENT CONTRACTING FOR SOFTWARE DEVELOPMENT COURSE MANUAL 33 (1994) [hereinafter COURSE MANUAL]; Lan Cao & Balasubramaniam Ramesh, *Agile Requirements Engineering Practices: An Empirical Study*, IEEE SOFTWARE, Jan./Feb. 2008, at 60, 64; Fowler, *supra* note 20 (noting engineers' "nature is to resist change. The agile methods, however, welcome change").

<sup>29</sup> Fowler, *supra* note 20.

<sup>30</sup> Szalvay, *supra* note 6, at 1-2; Cao & Ramesh, *supra* note 28, at 63.

<sup>31</sup> Cao & Ramesh, *supra* note 28, at 63.

<sup>32</sup> *Id.* Similarities can be drawn between modern software development and the Japanese auto industry forty years ago. Beginning in the 1970s Japanese auto manufacturers began adopting lean manufacturing practices to develop new vehicles. Michael A. Cusumano, *Japanese Technology Management: Innovations, Transferability, and the Limitations of "Lean" Production* 1 (Sloan Sch. of



### C. Modern Trends in Both Governance and Technology Push Federal Procurement Towards Agility.

Today's government IT landscape reflects a purposeful march toward privatization.<sup>33</sup> Growing budget deficits in recent decades have rendered the Government anxious to deliver smarter, more cost-effective services to its citizens.<sup>34</sup> In addition to advances in hardware and technology, the Government's approach to building complex, national systems has also evolved from the stand-alone computer systems of the 1960s, to the networked stovepipes of the 1970s and 1980s, to the glued-together middleware systems of the 1990s, and finally to the present day disaggregated, service-oriented architecture model.<sup>35</sup> In a setting like today's, mission-critical applications are distributed across multiple, loosely-coupled systems that communicate freely with one another,<sup>36</sup> allowing for rapid development and repurposing of investments.<sup>37</sup> Despite advancing industry approaches to large IT

---

Mgmt., Mass. Inst. of Tech., Working Paper No. 3477-92, 1992). Due to this refined approach's reduced overhead, Japanese manufacturers were able to generate thirty percent more profit per vehicle than Chrysler, one-hundred percent more per vehicle than Ford, and two-hundred percent more per vehicle than General Motors. MARY POPPENDIECK, CUTTER CONSORTIUM EXECUTIVE REPORT, LEAN DEVELOPMENT AND THE PREDICTABILITY PARADOX 1 (2003). When compared to Detroit's traditional design and manufacturing practices, Japanese manufacturers produced higher quality vehicles in one-third less time. *Id.* While in the United States nearly half of all projects missed their target date, in Japan, all but one sixth hit the market on time. *Id.* Japan's lean-manufacturing methods produced cheaper, more successful cars faster. *Id.*

<sup>33</sup> Darrell A. Fruth, *Economic and Institutional Constraints on the Privatization of Government Information Technology Services*, 13 HARV. J.L. & TECH. 521, 524-25 (2000).

<sup>34</sup> *Id.*

<sup>35</sup> DOD ACQUISITION REPORT, *supra* note 1, at 7.

<sup>36</sup> A detailed explanation of the evolution to a service-oriented architecture is beyond the scope of this Note, but in simple terms, if one thinks of information as a product, the evolution mimics the move from individual manufacturers producing a good, to a factory producing a good, to that factory adopting an assembly line approach. A government agency might build a system once, (e.g. an application to calculate the appropriate census block based on a given address,) and then will have subsequent systems query that already-built application, rather than duplicate the functionality. Any given request for information may go through five, ten, perhaps even a hundred systems, each of which can be procured or upgraded individually. Interview with Greg Elin, Chief Data Officer, FCC & Michael Byrne, Chief Geographic Info. Officer, FCC, in Wash., D.C., (Mar. 11, 2011) [hereinafter FCC Interview].

<sup>37</sup> *Id.*; Orszag Memorandum, *supra* note 14, at 1 (noting that "[b]y setting the scope of projects to achieve broad-based business transformations rather than focusing on essential business needs, Federal agencies are experiencing substantial cost overruns and lengthy delays in planned deployments. Compounding the problem, projects persistently fall short of planned functionality and efficiencies once deployed.").

projects, by forcing all requirements to be outlined upfront, federal procurement regulations continue to lock agencies into large, single-pass procurements.<sup>38</sup>

## **II. Despite its Advantages and Long-Standing History, the Rigidity of Traditional Software Development Bears Risks Disproportionate to its Returns.**

Traditional software development methodologies, commonly known as waterfall software development, are defined primarily by a sequential process<sup>39</sup> consisting primarily of two distinct stages: analysis and coding.<sup>40</sup> The analysis stage, which in some situations can span months or even years,<sup>41</sup> asks prospective system owners to outline all conceivable requirements in a rigidly structured process before formally giving the requirements documentation to a software development team.<sup>42</sup> The development team then completes a prolonged development cycle, fulfilling the specifications the customer described in the requirements documentation.<sup>43</sup> Following this cycle, stakeholders and developers come together (for the first time) to verify that the software meets all initial specifications.<sup>44</sup> These five steps are traditionally understood as the typical waterfall framework completed over the course of the procurement system.<sup>45</sup> While waterfall development provides many advantages in the federal contracting environment, it also introduces many shortcomings both conceptually and practically.

---

<sup>38</sup> See *Examining the President's Plan for Eliminating Wasteful Spending in Information Technology: Hearing Before the S. Subcomm. on Fed. Fin. Mgmt., Gov't Info., Fed. Servs., and Int'l Sec.*, 112th Cong. 3-4 (2011) (statement of Vivek Kundra, Fed. Chief Info. Officer, Office of Mgmt. and Budget) (noting that the current procurement model does not align well with the rapid pace of today's technology cycle).

<sup>39</sup> Szalvay, *supra* note 6, at 1-2.

<sup>40</sup> Winston W. Royce, *Managing the Development of Large Software Systems: Concepts and Techniques*, in *PROCEEDINGS OF IEEE WESCON* (1970), reprinted in *PROCEEDINGS OF THE NINTH INTERNATIONAL CONFERENCE ON SOFTWARE ENGINEERING* 328, 328 (1987).

<sup>41</sup> GEN. SERVS. ADMIN. STRATEGIC INFO. TECH. ANALYSIS DIV., WHITE PAPER: MODULAR CONTRACTING 16, 8 (1997) [hereinafter MODULAR CONTRACTING WHITE PAPER]. See KUNDRA, *supra* note 13, at 17 (noting “[t]he acquisition process can require program managers to specify the government’s requirements up front, which can be years in advance of program initiation”); Orszag, *supra* note 14, at 2 (observing that “[h]istorically government IT projects have involved expansive, long-term projects that attempt to change almost every aspect of a business system at once . . . [and have] taken years, sometimes a decade, and have failed at alarming rates”).

<sup>42</sup> COURSE MANUAL, *supra* note 28, at 33.

<sup>43</sup> See W. ROYCE, *supra* note 40, at 328.

<sup>44</sup> *Id.* at 329.

<sup>45</sup> More formally, the standard waterfall process follows five distinct steps: (1) requirements are defined and documented, (2) the system is designed and architected, (3) development teams code the software, (4) the software is tested and validated against the requirements from step one, and finally (5) the system is deployed. See *id.* at 328-29.

### **A. The Rigidity of the Waterfall Process Brings Several Advantages to Federal IT Procurement.**

In both form and function, waterfall development mirrors current procurement models. The first step requires that future system owners document their needs extensively, and such requirements are easily incorporated into a solicitation. This up-front documentation allows agencies to safely budget funds and allows contracting officers (COs) to marry funds to specific, pre-defined contractual events. "Phase gates" within the contract provide agencies objective standards to evaluate a project's propensity for success and allow agencies to abandon projects in some instances when the prospect of successful delivery appears grim. Within an agency, waterfall development requires COs to justify projects to stakeholders early in the development cycle, and this overall process has gained agencies' tacit endorsement over the years.<sup>46</sup> By nature of its formal requirements, waterfall software development leaves behind it a wake of documentation, the lifeblood of bureaucracy.

### **B. Waterfall Software Development Fails to Adequately Respond to the Ever-Changing Conditions that Make Up a Project's Problem Space.**

Despite its long-standing history, waterfall software development comes with shortcomings. The outdated methodology assumes requirements can be predicted upfront. As a result it fails to adequately respond to changing conditions and forces agencies to incur costs disproportionate to project returns.

#### **i. Both conceptually and practically, waterfall assumes that contracting agencies can accurately forecast a project's requirements in advance.**

Waterfall relies on two key assumptions: (A) That requirements are fully understood by both contractor and contracting agency, and (B) That requirements are unlikely to change.<sup>47</sup> Yet due to the often-unique nature of government systems,

---

<sup>46</sup> Richard K. Cheng, *On Being Agile*, NEXTGOV, Sept. 23, 2010, [http://www.nextgov.com/nextgov/ng\\_20100923\\_7965.php](http://www.nextgov.com/nextgov/ng_20100923_7965.php).

<sup>47</sup> Szalvay, *supra* note 6, at 1-3; COURSE MANUAL, *supra* note 28, at 33. Such an approach makes sense when it is traced back to its traditional roots. Engineers building bridges rely on well-established laws of physics supported by the fundamentals of mathematics, two sciences that are rigidly defined and easily relied upon. Software development, on the other hand, is not bound by the rules of the physical world. Gravity, an assumed constant in engineering, has relatively little relevance to programming.

software contractors typically face challenges to which they have never been exposed, rendering predicting of future needs an inherently futile task.<sup>48</sup>

The traditional procurement model, which leans heavily on waterfall's central tenants, demands specificity in planning, funding, and acquisition, and requires the contractors and COs to attain a certain level of "predictive precision" simply not found in the ever-changing world of technology.<sup>49</sup> As such, by requiring that all specifications be outlined upfront, the current procurement model encourages agencies to attempt an inevitably unsuccessful endeavor and thereby increases project risk.<sup>50</sup>

The unrealistic premises of waterfall procurement create a snowball effect. Because it is impossible for COs to resolve all unknowns at the onset of a procurement, they often err on the side of caution and compound errors by over-documenting requirements and making erroneous assumptions.<sup>51</sup> This risk is further exacerbated as a result of waterfall's insistence on delaying testing until late in the development cycle,<sup>52</sup> meaning mistakes often do not become apparent until corrections are cost-prohibitive.<sup>53</sup> Especially given the size of typical government procurements, "[t]rying to implement everything at once increases the risk of failure on a grand scale if the technology fails to work as expected or if the original idea was misconceived."<sup>54</sup> To function optimally, waterfall methodologies require both

---

<sup>48</sup> See Mark Leicester, *Government Projects the Agile Way: Can It Be Done?*, ST. SERVICES COMMISSION (Apr. 16, 2009), <http://www.ssc.govt.nz/blog/government-projects-agile-way-can-it-be-done>.

<sup>49</sup> MODULAR CONTRACTING WHITE PAPER, *supra* note 41, at 7; Cao & Ramesh, *supra* note 28, at 63. See *Further Details on Modular Contracting Left for Public Meeting--Proposed Rule Adds Little to Clinger-Cohen Act*, 39 GOV'T CONTRACTOR ¶ 161, Apr. 2, 1997 [hereinafter *Further Details*].

<sup>50</sup> MODULAR CONTRACTING WHITE PAPER, *supra* note 41, at 7 (noting that "forcing this rigidity into the systems development acquisition process – intending to minimize risks and mistakes – can actually exacerbate the same."). As FCC Geographic Information Officer Michael Byrne commented, "procurement is like a board game. You are trying to move your pieces to the other side. This would be easy if you had no opponent. When the other team begins moving forward there's no way to . . . know how to optimally get to the other side, and waterfall software development expects just that. Data changes, technology changes, policy changes, governance changes - this is the opposing player - and there's no way to predict the interaction with my pieces." FCC Interview, *supra* note 36.

<sup>51</sup> SHERMAN, *supra* note 15, at 210-11; POPPENDIECK, *supra* note 32, at 1, 5 (arguing "the more companies attempted to define the process of product development, the less the organization was able to carry out that process").

<sup>52</sup> *Further Details*, *supra* note 49.

<sup>53</sup> *Id.*

<sup>54</sup> *Id.* Consider the process of documenting the requirements to develop a BlackBerry or iPhone. Both smartphones are roughly 3x5, square with rounded edges, black, have a keypad and on/off switch, make calls, send and receive e-mail and text messages, browse the internet, and have downloadable applications. On paper they are identical. It is difficult to objectively describe, for example, "intuitive interface." Prototyping is the vocabulary agile uses to solve this problem. FCC Interview, *supra* note 36.

contracting parties to possess perfect information. This a practical impossibility when federal agencies acquire major IT systems.

**ii. Waterfall development fails to adequately respond to change.**

A single-pass approach assumes that all variables remain constant throughout the contracting process.<sup>55</sup> Yet experts and lay consumers alike know that technology slowly nudges even the most recent purchases toward obsolescence. Despite this, government bureaucracy is, by design, a time consuming and cumbersome endeavor.<sup>56</sup> Under waterfall, because design is distinct from development, development necessarily occurs in isolation from management efforts.<sup>57</sup> Requirements are not permitted to change throughout the process, regardless of the practical or technical realities of the agency's needs.<sup>58</sup> Ultimately, technology and the needs of the contracting agency often outpace the procurement process itself.<sup>59</sup>

If partway through the procurement process a superior new technology emerges or an agency reevaluates its requirements and decides its needs have changed—a not-too-distant reality given the span of multi-year procurements—the contractor is both disincentived and contractually prohibited from adapting accordingly.<sup>60</sup> Traditional approaches to procurement treat any deviation from the contract's initial requirements as a failure on the contractor's part and risk the contracting agency withholding payment.<sup>61</sup> In a sense, waterfall equates software development to Ford's assembly line, but in this case the conveyor belt moves so slowly<sup>62</sup> that the product is obsolete by the time it is ready for delivery.<sup>63</sup> In the

---

<sup>55</sup> Szalvay, *supra* note 6, at 1.

<sup>56</sup> Audio recording: Removing Barriers to Organizational Agility, George Washington University Tech Alumni Group Federal Executive Roundtable (Nov. 4, 2010) [hereinafter Federal Executive Roundtable] (on file with author).

<sup>57</sup> *Id.*

<sup>58</sup> *Id.*

<sup>59</sup> *Id.*; KUNDRA, *supra* note 13, at 17 (observing “the lag between when the [G]overnment defines its requirements and when the contractor begins to deliver is enough time for the technology to fundamentally change”).

<sup>60</sup> Federal Executive Roundtable, *supra* note 56.

<sup>61</sup> MODULAR CONTRACTING WHITE PAPER, *supra* note 41, at 13.

<sup>62</sup> A prime example of such a phenomenon is the U.S. Patent and Trademark Office (PTO). In 1983 the PTO initiated a billion-dollar automation effort using waterfall. At the time the PTO's patent files contained some thirty-three million documents, supporting patents issued since 1790, and grew at the rate of approximately one million documents per year. The goal was to have a fully deployed system by 1990, however by 1987, the PTO began to realize that the program was fatally behind schedule and switched to an agile approach. COURSE MANUAL, *supra* note 28, at 66-67. The PTO was able to successfully deploy its patent search system (CSIR) later that year, and by 1992 had a significant portion of its automation system integrated into its workflow. U.S. GOV'T

context of long-term projects, waterfall's quest to eliminate uncertainty at all costs creates the risk that when delivery finally occurs, what is delivered may no longer meet the agency's needs.<sup>64</sup>

**iii. Waterfall development provides a return disproportionately less than its significant administrative cost.**

The documentation requirements of waterfall impose an unnecessary burden on agencies. Agencies often spend as much as two years making project assumptions,<sup>65</sup> defining requirements, and developing cost-and-time estimates.<sup>66</sup> Today, detailed requirement-gathering has become a discipline in and of itself, which is wholly separate from system-development<sup>67</sup> and is often contracted out by the Government.<sup>68</sup> Traditional software development requires a one-direction, time-delayed information flow<sup>69</sup> that is inherently inefficient.<sup>70</sup> This waterfall software development inefficiency requires contracting agencies to invest significant amounts in outlining specifications, often forcing those agencies to hire more contracting specialists, technical experts, consultants, and managers than actual software developers.<sup>71</sup>

---

ACCOUNTABILITY OFFICE, GAO/AIMD-93-15, PATENT AND TRADEMARK OFFICE: KEY PROCESSES FOR MANAGING AUTOMATED PATENT DEVELOPMENT ARE WEAK 4 (1993).

<sup>63</sup> See generally Szalvay, *supra* note 6, at 1. The ability to adapt to change has implications beyond project efficiency as well. Hackers, cyber terrorists, and other malcontents are not bound by the rigidity of waterfall and are often able to adapt more rapidly to government countermeasures than the Government can develop defenses. See Federal Executive Roundtable, *supra* note 56. Similarly, after spending nearly two years and \$600 million dollars developing handheld computers for use in conjunction with the 2010 census, the U.S. Census Bureau was forced to drop its plans and revert to paper-based collection when it realized the waterfall approach it had adopted would not deliver a working system in time for the census data collection. WHITE HOUSE FORUM, *supra* note 5, at 5-6.

<sup>64</sup> The Federal Communications Commission recently procured the National Broadband Map using an agile framework. Within six months, the agencies had a working prototype, and six months following that, a final product. Less than ten percent of the project's total funding was used to make a prototype, the source code of which being distributed alongside the project's RFQ. The contract's first deliverable was requirements documentation outlining a project management plan, but daily scrum meetings provided both agency and contractor the opportunity to revise requirements. FCC Interview, *supra* note 36.

<sup>65</sup> Project assumptions might typically include the format of input data, deadlines, technology, policy, or governance. FCC Interview, *supra* note 36.

<sup>66</sup> MODULAR CONTRACTING WHITE PAPER, *supra* note 41, at 7.

<sup>67</sup> POPPENDIECK, *supra* note 32, at 6.

<sup>68</sup> SHERMAN, *supra* note 15, at 222-24.

<sup>69</sup> While documentation may memorialize requirements, the act of documentation delays the information's communication and prevents meaningful dialog regarding the project specifications.

<sup>70</sup> POPPENDIECK, *supra* note 32 at 6.

<sup>71</sup> Federal Executive Roundtable, *supra* note 56.

### **III. Given its More Effective Response to Change, Agile Software Development Can Deliver Better Performing Projects More Quickly and Cheaply than Waterfall.**

Agile development mitigates many of the risks imposed on agencies by waterfall development. Agile represents an iterative approach to software development that prioritizes communication over documentation and that adapts to change rather than futilely attempting to predict it. As a result, agile projects tend to be more efficient, less costly, and more predictable than their waterfall counterparts.

#### **A. The Ability to Rapidly Respond to Change is the Shibboleth of an Agile Project.**

At its most fundamental level, agility is the ability to quickly and effectively respond to change.<sup>72</sup> Agile software development came about in the mid-1990s as an alternative to heavyweight approaches' tendency for budget overruns.<sup>73</sup> Whereas waterfall requires an "over the fence" approach, by delaying decision making, agile software development embraces an "I'll know it when I see it" mentality.<sup>74</sup> As a result, agile software development presents agencies with three key advantages over the traditional waterfall approach to IT procurement, in that agile software development prefers: (i) many short development cycles over a single pass; (ii) person-to-person communications over up-front documentation; and (iii) adapting to changing needs i

##### **i. Agile is iterative, breaking long-term projects into multiple independent, manageable tasks.**

Agile development can be thought of as the amalgamation of many short waterfall development cycles.<sup>75</sup> An agile project consists of multiple coding sprints, commonly known as iterations, in which tasks are chosen from an ongoing feature backlog<sup>76</sup> and become realized early on through rapid prototyping and testing.<sup>77</sup> "An

---

<sup>72</sup> Lee & Xia, *supra* note 11, at 88.

<sup>73</sup> *Id.* at 89.

<sup>74</sup> Szalvay, *supra* note 6, at 3; POPPENDIECK, *supra* note 32, at 11-12.

<sup>75</sup> Cheng, *supra* note 46.

<sup>76</sup> From the agency's perspective, an agile backlog is simply a wish list of features in priority of business value, wherein the agency lists the functionality it desires from a system and when that can be expected. From the contractor's perspective, the backlog is a to-do list, in order of priority, listing the project deliverables and their deadlines.

<sup>77</sup> For a more detailed explanation, *see* Fowler, *supra* note 20.

iteration produces a small, tested, integrated increment of business value that is validated by customers and used as feedback for the next iteration. Iterations occur at short, regular intervals and they involve everyone: from architects to testers to the help desk staff.”<sup>78</sup> Agile methodologies see the process as dynamic, evolving, and organic, rather than static, predefined, and mechanistic.<sup>79</sup> Agile breaks a system into smaller, more fungible chunks of immediate value to the user, rather than attempting a software development project as a single, undivided task.<sup>80</sup>

**ii. Agile prioritizes communication and collaboration over documentation and structure.**

To facilitate agile’s hallmark rapid delivery in place of time-consuming formal documentation and rigid structure, agile relies on extensive person-to-person communication and organic processes. In simplest terms, agility is defined by a software team’s ability “to efficiently and effectively respond to and incorporate requirement changes during the project life cycle.”<sup>81</sup> To achieve this goal agile focuses on collaboration rather than documentation. Agile promotes frequent, continuous delivery of working software and embraces changing requirements through close collaboration, self-organizing and self-motivated teams, face-to-face communications, and in turn, continuous adaptation.<sup>82</sup>

In practical terms, agile values individuals and interactions over processes and tools, working software over comprehensive documentation, customer collaboration over contract negotiation, and responding to change over following a plan.<sup>83</sup> Because agile prioritizes one-on-one communication over formal documentation, unnecessary documentation is wholly alleviated and overhead costs are reduced.<sup>84</sup> Moreover, requirements tend to be more concrete and clearer and to produce a better final product.<sup>85</sup> Given that project teams undertake smaller projects under an agile framework, systems can be delivered without the sluggishness inherent to waterfall projects.

---

<sup>78</sup> POPPENDIECK, *supra* note 32, at 9.

<sup>79</sup> See Lee & Xia, *supra* note 11, at 89.

<sup>80</sup> Federal Executive Roundtable, *supra* note 56.

<sup>81</sup> Lee & Xia, *supra* note 11, at 90.

<sup>82</sup> *Id.* at 89, 92.

<sup>83</sup> *Id.* at 89.

<sup>84</sup> Cao & Ramesh, *supra* note 28, at 63.

<sup>85</sup> *Id.* at 64.



### **iii. Agile is adaptive rather than predictive.**

Traditional engineering methodologies emphasize rigid planning and forecasting.<sup>86</sup> By contrast, agile embraces the philosophy that requirements are discovered through use and refinement.<sup>87</sup> As requirements emerge organically during the software development process,<sup>88</sup> they evolve as stakeholders gain a deeper understanding of the project and their needs.<sup>89</sup> Put differently, by delaying decisions to the last possible moment, the agile model allows agencies to make decisions “based on facts rather than forecasts.”<sup>90</sup> This provides several distinct advantages over traditional waterfall methodologies.

#### **B. By More Effectively Responding to Change, Agile Methodologies Can Offer Agencies Greater Predictability and More Timely and Less Costly Project-Delivery.**

Agile projects more readily adapt to change and are delivered on time and for less cost, ensuring a high degree of predictability.

#### **i. Agile development adapts to change.**

By delaying irreversible decisions until later on in the procurement cycle, agile’s incremental and iterative framework allows teams to rapidly respond to the empirical realities of a project through continuous evaluation of project plans.<sup>91</sup> Following each short coding sprint, agency stakeholders are presented with the ability to reevaluate each desired function’s relative priority, and the subsequent coding sprints adapt to embrace those shifting requirements.<sup>92</sup> Agile’s multiple, rapidly executed increments allow stakeholders to specify more easily defined objectives, rather than abstract technical requirements.<sup>93</sup> Agile’s evolutionary approach embraces early, successive prototyping, and thus produces timely working releases that meet stakeholders’ needs.<sup>94</sup> This approach is particularly useful in the Federal

---

<sup>86</sup> See Fowler, *supra* note 20.

<sup>87</sup> Federal Executive Roundtable, *supra* note 56.

<sup>88</sup> Cao & Ramesh, *supra* note 28, at 60.

<sup>89</sup> See *id.* at 63.

<sup>90</sup> POPPENDIECK, *supra* note 32, at 3.

<sup>91</sup> See Szalvay, *supra* note 6, at 6.

<sup>92</sup> See ROYCE, *supra* note 19, at 4.

<sup>93</sup> DOD ACQUISITION REPORT, *supra* note 1, at 47-48.

<sup>94</sup> *Id.*

Government's modular, service-oriented architectures.<sup>95</sup> Agile's iterative approach in the end allows projects to more accurately and more immediately meet agency needs.

## **ii. Agile development is more efficient than waterfall development.**

In the context of large-scale government projects, agile can deliver systems sooner than its similarly sized waterfall counterparts.<sup>96</sup> Agile software development is more efficient than traditional development methodologies for five key reasons: First, agile allows development teams to begin work immediately, even before requirements are fully known, an impossibility under a waterfall contract;<sup>97</sup> second, whereas waterfall development uses multiple factors to prioritize tasks, agile focuses solely on maximizing business value,<sup>98</sup> and thus, high-priority goals are realized and delivered to the customer early on;<sup>99</sup> third, because agile communicates concrete specifications via face-to-face discussions, rather than lofty goals through formal documentation, and because agile embraces a streamlined decision-making process, rather than a rigid hierarchical structure, development teams gain a better understanding of and produce products that better meet agency needs and requirements;<sup>100</sup> fourth, agile promotes the end user's involvement throughout the process and thus ensures early and constant validation of features, minimizing the need for costly and time-consuming pre-deployment changes;<sup>101</sup> finally, because agile's continual validation model requires constant use of the procured system, users receive training and preparation to use the new system on day one.<sup>102</sup> For all of these reasons, agile is more likely to deliver projects on time.

## **iii. Agile is inherently less costly than waterfall.**

Agile projects tend to be less costly overall for three primary reasons. First, because COs do not have to spend extensive time pre-award defining project requirements, agile development inherently imposes a lesser administrative burden.<sup>103</sup> Face-to-face communication may require less investment than extensive documentation, in so far as in-person communications facilitate prompt clarification

---

<sup>95</sup> *Id.*

<sup>96</sup> ROYCE, *supra* note 19, at 4.

<sup>97</sup> Szalvay, *supra* note 6, at 8.

<sup>98</sup> Cao & Ramesh, *supra* note 28, at 64.

<sup>99</sup> *Id.*

<sup>100</sup> *Id.*; DOD ACQUISITION REPORT, *supra* note 1, at 53.

<sup>101</sup> Cao & Ramesh, *supra* note 28, at 65.

<sup>102</sup> DOD ACQUISITION REPORT, *supra* note 1, at 47-48.

<sup>103</sup> Cao & Ramesh, *supra* note 28, at 65.

of any discrepancies or misunderstandings. Second, administrative costs are further reduced as the need to modify the original contract if requirements change is wholly eliminated.<sup>104</sup> Finally, once the agency has awarded the contract, because agile breaks the project into smaller, more approachable tasks, agile reduces cost and cycle time by better utilizing the contractor's resources.<sup>105</sup> As a result, agile more consistently delivers projects on or under budget more often than waterfall.

#### **iv. Agile is predictable.**

While at first glance agile may seem less predictable, decreasing the amount of speculation involved in decision making inherently increases the predictability of the final outcome.<sup>106</sup> Unlike waterfall's hands-off approach, small increments of constant feedback provide a safety net of predictability upon which the contracting agency can base its decisions.<sup>107</sup>

### **C. Despite its Advantages, Agile Methodologies Present Contracting Officers With Several Key Challenges.**

Agile's define-as-you-go approach does not often lend itself to satisfying government documentation requirements. Its lack of pre-award specificity makes costs difficult to estimate, and, when compared to waterfall, agile exposes the contracting agency to greater risk.<sup>108</sup>

#### **i. Agile methods are not well suited to meet government documentation requirements.**

Extensive government contracting documentation and reporting requirements impose the greatest barrier to the adoption of agile procurement in the U.S. Traditional government contracting requires that project "artifacts" be generated at each step of the process to memorialize project activity.<sup>109</sup> Agile projects, on the other hand, forgo formal paperwork.<sup>110</sup> Yet agile is not necessarily

---

<sup>104</sup> *Id.*

<sup>105</sup> DOD ACQUISITION REPORT, *supra* note 1, at 53. See POPPENDIECK, *supra* note 32, at 15.

<sup>106</sup> See POPPENDIECK, *supra* note 32, at 3.

<sup>107</sup> *Id.*; DOD ACQUISITION REPORT, *supra* note 1, at 47-48, 53.

<sup>108</sup> Chief among agile's criticisms in the private sector may be that its emphasis on functional, rather than technical requirements often underserves the project's non-functional goals. Cao & Ramesh, *supra* note 28, at 64-65.

<sup>109</sup> Glen B. Alleman et al., *Making Agile Development Work in A Government Contracting Environment*, in CONFERENCE PAPERS, AGILE DEV. CONFERENCE 114, 114 (2003).

<sup>110</sup> Cao & Ramesh, *supra* note 28, at 64.

incompatible with these procurement requirements. Although there is no perfect alternative to reporting progress to plan, in most cases, agencies can use earned value project management to reconcile the two schools of thought.<sup>111</sup> More difficult, agile methodologies can be reconciled with formal documentation requirements at the solicitation stage.

**ii. Agile engenders a lack of specificity pre-award, rendering cost estimates difficult to calculate.**

Agile does not require that potential systems outline technical specifications owners before contract award, potentially causing COs difficulty as they attempt to estimate project costs.<sup>112</sup> Often, project timelines are at best only roughly defined, and project teams may change drastically as requirements change.<sup>113</sup>

However, while it may allow for less specificity pre-award, agile constantly adapts to the actual needs of the project (rather than remaining true to initial projections). As a result, in the long run, agile provides more accurate cost metrics and greater budget fidelity overall.

**iii. Agile offers contracting agencies less risk-avoidance upfront.**

As a risk-avoidance measure, government contracting requires extensive reporting and detailed paper trails to accompany each step of the procurement process.<sup>114</sup> By integrating the contractor and contracting agency, agile development forces both to share the burden of risk.<sup>115</sup> Yet, while agile offers less risk avoidance up front, by integrating both the contractor and the contracting agency early on, it reduces the overall level of risk.<sup>116</sup>

---

<sup>111</sup> Earned value project management relies on progress payments, which are payments based on the value the contractor earned during that period rather than on the level of completion as it relates to an established project plan. Alleman et al., *supra* note 109, at 114.

<sup>112</sup> Cao & Ramesh, *supra* note 28, at 64-65.

<sup>113</sup> *Id.*

<sup>114</sup> Cheng, *supra* note 46.

<sup>115</sup> *Id.* Multiple, small investments of time and effort acknowledge that humans make mistakes and mitigates project risk through independent, componentized steps. FCC Interview, *supra* note 36.

<sup>116</sup> Early engagement between technical contacts, COs, and legal teams is crucial to a successful project. The process itself should be agile. The requirements writing process should not be waterfall if the project is not. FCC Interview, *supra* note 36.

#### **IV. Current Regulations and Agency Practices Restrict Adoption of More Efficient Agile Methods.**

Current procurement practices restrict the Government's ability to adopt the agile model. The structure of the contracting process often conflicts with agile's central tenants, and even if this basic incongruence is overcome, government contracting embodies a culture averse to agile.

##### **A. The Structure of Traditional Government Contracting is Not Amenable to Agile Development Methods.**

For three reasons, government contracts are not written for agile.<sup>117</sup> First, as a matter of public policy, government contracting typically requires agencies to prefer competition at the cost of project deliverables.<sup>118</sup> Second, government programs generally involve significant lead times - forcing funding to be mapped out well in advance - rendering a "develop as you go" method counterintuitive.<sup>119</sup> Finally, most contract vehicles call for stable requirements, a method inherently incompatible with an adaptive model.<sup>120</sup>

##### **B. Government Procurement Embodies a Culture Averse to Agile Methodologies.**

Government procurement embraces an entrenched culture of rigid parameters, documentation, and meticulously defined processes.<sup>121</sup> This manifests itself in several forms. For one, government budgeting focuses on projects that fall within a single budget year, are controlled by a single budget program, and promise an easily predicted, low-risk return.<sup>122</sup> Additionally, at the most basic level, it is not agencies who procure IT but rather individuals on behalf of those agencies. As in any sector, these individuals' incentives may not align with their employer's goals. A

---

<sup>117</sup> Cheng, *supra* note 46.

<sup>118</sup> SHERMAN, *supra* note 15, at 3.

<sup>119</sup> Cheng, *supra* note 46.

<sup>120</sup> Fowler, *supra* note 20.

<sup>121</sup> Cheng, *supra* note 46 (commenting "[a]n entrenched project management culture has yet to give up rigid parameters, documentation and processes"). See Hilary Berger, *Agile Development in a Bureaucratic Arena*, 27 INT'L J. OF INFO. MGMT. 386, 392 (2007) (noting "where the organizational nature is one of bureaucracy and hierarchical status," a culture "counter-productive to [a]gile development" emerges because "management driven responsibility inhibits the fast, authoritative decision-making activities" that are "crucial" to agile).

<sup>122</sup> See JERRY MECHLING & VICTORIA SWEENEY, OVERCOMING BUDGET BARRIERS: FUNDING IT PROJECTS IN THE PUBLIC SECTOR 14 (1997).

CO who chooses an innovative but ultimately flawed contracting vehicle may face termination or reprimand, and without specific financial incentives, may gain nothing if the innovation proves beneficial.<sup>123</sup> Thus even if federal acquisition regulations on paper allow for agile development, in practice, COs have an incentive to avoid the near-term risk presented by innovation.<sup>124</sup> Third, new ideas must compete against proven ones and “urgent needs [...] take precedence over [future] possibilities.”<sup>125</sup> High-value but high-risk ideas like agile require support from upper management to take hold, and this support by-and-large does not exist.<sup>126</sup> Finally, across the board both COs and in-house IT departments are either unfamiliar with or unaccustomed to agile methods.<sup>127</sup> Before agile's adoption, significant cultural reform and education must take place on both an organizational and an individual level.

### **C. Current IT Procurement Methods are Not Amenable to Agile Development.**

Many IT contracting vehicles may exist in the context of any given project. While price-directed bids, competitive negotiations, conceptual negotiated sourcing, engineering selection, and modular contracting may provide some promise, no existing contracting vehicle is fully compatible with agile methodologies.

#### **i. Price directed (sealed bid)**

Where bids are solicited and chosen solely based on cost, traditional acquisition takes the form of price-directed or sealed bid procurement.<sup>128</sup> This model does not accord with agile in that it requires precisely defined requirements in advance of solicitation.<sup>129</sup> Additionally, such technical requirements crowd out many private sector firms and spawn a government-specific industry that limits competition and available resources.<sup>130</sup>

---

<sup>123</sup> *See id.* at 8.

<sup>124</sup> *See id.*

<sup>125</sup> *Id.* at 11.

<sup>126</sup> *See id.*

<sup>127</sup> Cheng, *supra* note 46.

<sup>128</sup> SHERMAN, *supra* note 15, at 230.

<sup>129</sup> *See id.*; EMMETT E. HEARN, FEDERAL ACQUISITION AND CONTRACT MANAGEMENT 55 (5th ed. 2002).

<sup>130</sup> SHERMAN, *supra* note 15, at 251.

## **ii. Competitive negotiations**

A negotiated procurement model allows contracting agencies to solicit proposals, similar to the price-directed model.<sup>131</sup> Unlike the price-directed model, however, bids are not sealed and agencies are permitted to discuss deficiencies with firms, which may then resubmit their bids.<sup>132</sup> Nevertheless, the negotiated procurement model still requires the agency to outline technical specifications up front, making it incompatible with agile methods.

## **iii. Technology/Conceptual Negotiated Sourcing**

Technology or conceptual negotiated sourcing is a two-phase model. First the contracting agency awards the initial phase of design and conception to multiple competing bidders through multiple small-dollar, short-term contracts.<sup>133</sup> The agency then selects its preferred plan from among the submissions and continues with the procurement as if it were a sealed bid.<sup>134</sup> This approach is not ideal from an agile perspective, as it lengthens the time for which major resources are committed and delays prototyping until multiple firms can submit project plans.<sup>135</sup> This approach is also less efficient, as it forces the Government to fund parallel development and planning efforts.<sup>136</sup> For the federal IT procurement to better embrace innovation, change is needed.

## **iv. Engineering Selection under the Brooks Act**

Under the Brooks Act,<sup>137</sup> initially intended for traditional engineering fields, agencies may use a two-phase negotiated procurement model to procure systems that may justifiably be performed by engineers.<sup>138</sup> In the first phase, following a solicitation, competing firms submit proposals outlining methodology and techniques.<sup>139</sup> Purchasers evaluate and rank proposals based not on price but rather on each firm's unique qualifications, performance history, and likelihood of

---

<sup>131</sup> See FAR Part 15.

<sup>132</sup> *Id.*; SHERMAN, *supra* note 15, at 265.

<sup>133</sup> SHERMAN, *supra* note 15, at 299-301.

<sup>134</sup> *Id.* at 301.

<sup>135</sup> *Id.* at 302.

<sup>136</sup> *Id.*

<sup>137</sup> H.R. REP. NO. 92-1188, at 1, 2 (1972).

<sup>138</sup> FAR 36.6.

<sup>139</sup> *Id.*

delivering a final product that meets agency needs.<sup>140</sup> From these rankings, the purchasing agency generates a “short list” and requests more detailed proposals from the top-ranked firms before making a final selection.<sup>141</sup> Despite the engineering provision’s compatibility with agile, however, contracting agencies are statutorily prohibited from using the seemingly ideal contracting vehicle when procuring software.<sup>142</sup>

## **v. Modular Contracting**

Section 5202 of the Information Technology Management Reform Act of 1996 (ITMRA)<sup>143</sup> encourages agency heads to use modular contracting when acquiring major information systems.<sup>144</sup> Designed to transform federal IT acquisitions into a results-oriented procurement system,<sup>145</sup> under modular contracting, complex systems are disaggregated into several discrete, interoperable services and procured accordingly.<sup>146</sup> In order to “reduce program risk and incentivize contractor performance while meeting the government’s need for timely access to rapidly changing technology,”<sup>147</sup> ITMRA Section 5202 states that agencies *should* “to the maximum extent practicable” use modular contracting to acquire major systems, and *may* use modular contracting for all other systems.<sup>148</sup> The Act notes that modular contracts are easier to manage, have an increased likelihood of success when implementing complex systems, mitigate risk through compartmentalization, and allow subsequent increments to take advantage of

---

<sup>140</sup> FAR 36.602-1(a).

<sup>141</sup> 40 U.S.C. § 1103(d) (2006).

<sup>142</sup> FAR 36.601 specifically notes that “services that do not require performance by a registered or licensed architect or engineer . . . should be acquired pursuant to parts 13, 14, and 15.”

<sup>143</sup> National Defense Authorization Act of 1996, Pub. L. No. 104-106, 110 Stat. 659, 689 (1996) (codified in scattered sections of 40 U.S.C. and 41 U.S.C.). For a comprehensive overview of the ITMRA (later known as the Clinger-Cohen Act) reforms, see Stephen W. Feldman, *Clinger-Cohen Act of 1996*; GAO *Procurement Protest Authority*, 1 GOVERNMENT CONTRACT AWARDS: NEGOTIATION AND SEALED BIDDING § 1:12 (2010).

<sup>144</sup> FAR 39.103.

<sup>145</sup> Cohen *Plans IT Procurement Reform Legislation*, 37 GOV’T CONTRACTOR ¶ 83, Jan. 15, 1995, at 8-9 (observing that the act’s sponsor argued that “by the time a contract is awarded, the technology that is being bought is obsolete”).

<sup>146</sup> MODULAR CONTRACTING WHITE PAPER, *supra* note 41, at 24 (identifying a module as an “economically and programmatically separable segment” that “should have a substantial programmatic use even if no additional segments are acquired [and as] . . . a functional system or solution that is not dependent on any subsequent module in order to perform its significant functions”).

<sup>147</sup> FAR 39.103(a).

<sup>148</sup> *Id.*; FAR 2.101. A “major system” is a system estimated to exceed \$2 million or the threshold defined by OMB Circular A-109 or that is designated a “major system” by the head of the agency responsible for that system. FAR 2.101.



advances in technology.<sup>149</sup> In its traditional form, however, modular contracting still requires that specifications be known and documented prior to each module, and thus is not fully compatible with agile methods.<sup>150</sup>

By forcing requirements to be outlined at the onset of the process, the procurement vehicles outlined above force contracting agencies to utilize outdated development methods and help to perpetuate the stereotype that government systems are overpriced, inefficient, and obsolete at the time of delivery.

## **V. Significant Reform is Necessary to Allow Federal IT Procurement to Take Advantage of Industry Advances that Have Far Outpaced Laws Intended to Govern Them.**

For the Federal Government to become more agile, three distinct but parallel initiatives are necessary. First, COs and technical representatives must be educated as to the best practices for implementing agile in government contracting. Second, statutory requirements encouraging agencies into waterfall frameworks by demanding extensive requirements documentation pre-award should be avoided through alternative contracting vehicles. Finally, even if permitted by law, proponents must garner significant top-down support from within their agencies for agile practices to be accepted.

### **A. COs and COTRs Require Education Before Agile Can be Introduced to the Federal Contracting Environment.**

Before federal IT procurement can embrace agile methodologies, best-practices education is required on two levels. First, COs must learn best practices for agile contracting. Second, technical officers within the agency must learn best practices for agile project management in the government sector.

#### **i. COs must learn best-practices in agile contracting**

Many COs may not know what agile development is, let alone that it is a potential contracting vehicle and how best to implement it. The White House Office of Science and Technology Policy (OSTP) should oversee the joint efforts of

---

<sup>149</sup> FAR 39.103(b). See Danielle Conway-Jones, *Research and Development Deliverables under Government Contracts, Grants, Cooperative Agreements and Cradas: University Roles, Government Responsibilities and Contractor Rights*, 9 COMPUTER L. REV. & TECH. J. 181, 192 n.75 (2004).

<sup>150</sup> FAR 39.103.

the General Services Administration (GSA), the White House OMB,<sup>151</sup> and the Office of Federal Procurement Policy (OFPP) to educate COs as to agile contracting best practices.<sup>152</sup>

Solicitations should outline functional rather than technical requirements<sup>153</sup> and should be crafted without the significant lead time often characteristic of government procurement. The statement of work should specify high-level goals or systems, rather than specific features, and these goals should eventually become the framework that defines each subsequent sprint. The contract should also include specific, functional milestones and accompanying deadlines, which may also be married to an award schedule.

**ii. Agency technical contacts (COTRs) must learn best practices in agile project management.**

Alongside existing professional development efforts, contracting organization technical representatives must be trained in best practices for executing and managing development projects through an agile framework. This may include formal workshops for employees; the recruitment of experienced, private sector project managers; or simply a push to increase awareness of agile methodology and educational resources already at agencies' disposal.

**B. Existing Procurement Requirements Demand Extensive Documentation Up Front, Forcing Agencies into Agile Frameworks**

Even if an agile-friendly contracts are prepared, executed, and overseen successfully, two statutory stumbling blocks may hinder agile's adoption: (i) the

---

<sup>151</sup> Strengthening IT program management by exposing best practices and aligning IT procurement with innovation is a central tenet of the White House's plan to reform federal IT. KUNDRA, *supra* note 13, at 17 (noting "[t]he acquisition process can require program managers to specify the [G]overnment's requirements up front, which can be years in advance of program initiation . . . We need to make real change happen . . . by educating the entire team managing IT projects about the tools available to streamline the acquisition process.").

<sup>152</sup> Ideally instructors would create a scalable model outlining the various issues contracting agencies must broach before pursuing an agile project, such as how to ensure robust competition, how broad or specific the statements of work should be, whether to award to a single vendor or multiple vendors, and when to use fixed-price contracts or rely instead on other contracting vehicles. *See generally* KUNDRA, *supra* note 13.

<sup>153</sup> An example of a functional approach would involve requiring that a system use industry best security practices rather than describing a particular encryption algorithm.

solicitation and subsequent award of the contract; and (ii) the governance of the contract while it is in progress.<sup>154</sup>

### **i. Engineering services, FAR Part 36**

The single largest statutory roadblock to agile-friendly contracts is the requirement for technical specifications prior to solicitation. This requirement locks contracting agencies into a waterfall method post-award and is antithetical to agile methodologies. To pursue an agile framework, agencies should establish functional rather than technical benchmarks, developing a framework that can be used to define the focus of each subsequent sprint.

Part 36 of the FAR provides a modified procedure for procuring architectural and engineering services.<sup>155</sup> Such a procurement would follow a two-phase procedure. In phase one competing firms would submit proposals including their technical approaches and technical qualifications, but these would not include detailed design information or cost or price information. Agencies would then rank the top firms, which would be asked to submit detailed proposals before the winning firm was selected for a contract to be executed through the design-build framework.<sup>156</sup>

Under FAR 36.601(4)(a)(3), to be eligible for the architecture-engineering route, the acquisition must “logically or justifiably require performance by registered architects or engineers or their employees.”<sup>157</sup> Congress should expand the definition of this section to include computer and software engineers to allow IT procurement to follow this route.

### **ii. Modular contracting, FAR Part 39**

FAR 39.103 allows agencies to break large acquisitions into several smaller acquisition increments.<sup>158</sup> To aid in accounting, budgeting, and documentation, an agile procurement can be contracted as one overarching contract defining the scope of the work and outlining a schedule by which each sprint’s focus will be defined.

---

<sup>154</sup> This holds true for both the FAR and DoD regulations. DOD ACQUISITION REPORT, *supra* note 1, at 4.

<sup>155</sup> FAR 36.601.

<sup>156</sup> FAR 36.602-1.

<sup>157</sup> FAR 36.601(4)(a)(3). The U.S. Bureau of Labor and Statistics classifies the software engineering role as a subset of computer specialist, *see* [http://www.bls.gov/soc/soc\\_structure\\_2010.pdf](http://www.bls.gov/soc/soc_structure_2010.pdf), while all other engineering disciplines are a subset of engineer, [http://www.bls.gov/soc/soc\\_structure\\_2010.pdf](http://www.bls.gov/soc/soc_structure_2010.pdf). Additionally, software engineers do not receive professional engineering licenses from the state. If a state were to grant such a license, this would serve as an interesting test case.

<sup>158</sup> FAR 39.103 (implementing the Clinger-Cohen Act of 1996, Pub. L. No. 104-106).

This large contract may incorporate smaller contracts for each subsequent sprint, if necessary, as part of a process wholly in line with the Government's push toward system-oriented architectures.

### **C. COs Lack Top-Down Support for IT Procurement Reform**

The above goals cannot be realized without significant top-down support from the White House Office of Science and Technology Policy, OMB, and Chief Technology Officers at each agency. COs must be empowered to take risks with such an unconventional approach. While top administration officials have made statements supporting an agile approach to IT procurement reform, such rhetoric remains noncommittal. Tangible, specific guidance from senior administration officials to agency CIOs and CFOs must occur in order for innovation to succeed.

## **VI. Conclusion**

The federal IT procurement system is outdated. Projects are consistently delivered late, over-budget, and obsolete. Much of this trend can be traced back to flawed legal frameworks that lock agencies and contractors into an outdated development model. Through education, reform, and organization-wide support, federal agility can become a reality. Any computer user knows that as systems age they begin to slow. Today, the federal IT procurement system is running slowly, to the detriment of both agencies and the public, and it is long overdue for a system-wide upgrade.

**Figure 1:**  
**Comparison of Waterfall to Agile Software Development Timelines**

