Submission to the Senate Economics Committee

Inquiry into the 2016 Census

Dr Robert Merkel

A. Summary

The 2016 Census website ("e-Census") was a functionally simple website that nevertheless was a high-risk IT project due to its exacting reliability, security, and performance requirements. This was further complicated by the one-off nature of its use. Based on the public record and media reports, I identify three areas of concern about the website's development and deployment:

- lack of countermeasures for highly predictable "denial-of-service" attacks,
- a performance requirement that seems well below plausible actual demand; and
- reports of faults that should have been caught by appropriate whole-system testing, including failure recovery testing.

These problems raise questions about the nature of the relationship between the primary contractor, IBM, and the Australian Bureau of Statistics:

- Was the ABS a sufficiently informed customer that it was able to clearly articulate its requirements to its contractor?
- Was the primary contractor dedicated to delivering precisely what the customer asked for, or what the ABS actually needed – and if it is the former, what should it mean for future Australian IT contracts?

B. Introduction and Scope of Submission

This submission addresses questions relating to the development and operation of the Census website. As such, it addresses aspects of the following four terms of reference to the Inquiry:

a. the preparation, administration and management on the part of the Australian Bureau of Statistics (ABS) and the Government in the lead up to the 2016 Census;
b. the scope, collection, retention, security and use of data obtained in the 2016 Census;
c. arrangements, including contractual arrangements, in respect of the information technology aspects of the Census;

d. the shutting down of the Census website on the evening of 9 August 2016, the factors leading to that shutdown and the reasons given, and the support provided by government agencies, including the Australian Signals Directorate;

It is based on my background as a researcher and educator in the field of software engineering, media reports and public statements relating to the Census.

I have made no attempt to address the questions relating to the long-term storage of individual Census responses and its linkage to other data. I believe that these are very important topics for the committee to consider but have not had time to make myself fully aware of enough detail to make informed comment.

Please also see “About the Author” at the end of the submission for a disclosure of connections to affected parties (none are, I believe, significant).

C. e-Census: background and rationale

1. Online census forms have been used in Australia since the 2006 Census[1], and in many other jurisdictions.
2. For the 2016 Census, filling out the form online using a dedicated website (referred to as the “e-Census website” in this submission) was the default option; a paper form had to be specifically requested.
3. Online form submission greatly reduces the costs to the ABS of carrying out the Census, and is more convenient for many respondents.
4. As such, the choice to make online submission the default while offering the choice of a paper form appears appropriate.

D. The task

5. The e-Census website was, technically, a straightforward web application. The web server (software and hardware) for the website had two main tasks:
   i. On request, send web pages showing the appropriate census questions to a web browser (client).
   ii. Receive the responses from the web browser which contain the answers to the census questions, and store these answers for later retrieval and analysis.
6. To support these functions, several generic pieces of internet infrastructure would have been required, including:
   iii. a *name server* (also known as a DNS server). The “native address” of a computer on the internet is its *IP address*, a multi-part numeric identifier. For instance, the IP address of

[1] http://www.abs.gov.au/AUSSTATS/abs@.nsf/mediareleasesbyReleaseDate/4C85D1995909281FCA2571CD000F74D9?OpenDocument

the server computer for the Australian Parliament's public website is 202.14.81.88. A *name server* is a computer program (and/or the computer that it runs on) that maps the more familiar domain names (like *aph.gov.au*) to IP addresses.

iv. *Router*s, which are specialized computers that are used to direct internet traffic (as identified by its IP address) to the correct destination.

7. Displaying web pages and storing responses are required for virtually all websites. Software developers do not re-invent the wheel for every such website they create; they make use of a variety of reusable "software components". For instance, a *database* is a software component that can be used to efficiently store and retrieve large quantities of structured information, as was required for the census.
8. Compared to many other web applications, the e-Census had two characteristics that should have made the implementation simpler:
   - The census form displayed to the site's users is very similar for all respondents.
   - The data collected is simply stored, rather than used to update the website itself (for instance, a social media site like Facebook makes user-submitted status updates available for others to view immediately).
9. The overwhelming majority of the software used in the creation of the e-Census website would have been pre-written software components supplied by IBM and other vendors. The remaining "glue" code to join those components together should have been tiny by the standards of modern software development. I would be surprised if the entire system contained more than a few tens of thousands of lines of custom-written code.
10. There is an ever-expanding collection of such software components available to website (and other software) developers. The merits of particular components are often a subject of vigorous debate within the programming community. However, there are literally thousands of different combinations of software components that could have been used to satisfactorily build something as simple as the e-Census website.
11. Similarly, standard computing hardware available from many different suppliers would offer sufficient performance to support the e-Census website. Media reports indicate that the actual hardware used was "cloud infrastructure" provided by a subsidiary, Softlayer, of the primary contractor, IBM, and was located at Softlayer's data center in Sydney. In general, the use of rented cloud infrastructure is a very efficient and cost-effective way to provide computing applications, and its use is ubiquitous across the public and private sectors.

E. The Challenges

12. While the *functional requirements* – the description of what the e-census website had to do – were very simple, the *non-functional requirements* were not.
13. Firstly, the users of e-Census website expected it to be available and responsive when they chose to submit their information. Therefore, it was required to be reliable, even if individual hardware components failed, or other reasonably foreseeable negative events occurred.
14. Related to this, the site had stringent performance requirements – it needed to remain responsive even when a large number of people were submitting their census form in a short period of time. The question of just how many submissions the e-Census site should have been

able to deal is discussed in Section I. That said, while the e-Census website required higher capacity than most Australian government websites, it is tiny compared to the world's most used websites. While the exact figure is not disclosed, in excess of 200 million Google searches are conducted every hour – and each web search requires much more computation to process than the e-Census form.

15. Finally, the site had a number of security requirements, key amongst them:
    - Be sufficiently resistant to foreseeable "denial of service" attacks to remain available to respondents.
    - To keep respondents' Census information private.
16. The one-off, time-limited nature of the Census was an additional complicating factor. Even with good software engineering practices, problems are usually found when a new software system is deployed into "production use" for the first time. Therefore, if possible, software engineers prefer to initially deploy new software to a small group of users (often in a "beta test" or "soft launch"), and gradually expand usage, as bugs are identified and fixed, and the system proves itself capable of coping. When this is not possible, there is an elevated risk of problems. The Obamacare website[2], the Seven Network's Rio Olympics App[3], and the popular mobile game Pokemon Go[4] are examples of high-profile software systems that had severe technical problems when they were launched with a "big-bang".
17. The e-Census is inherently a "big-bang" deployment, given the ABS encouraged people to complete their Census on the night of August the 9th.
18. There are a number of steps that can be taken to mitigate the risks of a big-bang deployment, including:
    - more rigorous pre-deployment testing
    - deliberately over-engineering the design to cope with stresses in excess of what it might be expected to face.
    - Contingency planning to deal with any failures that crop up as expeditiously as possible.
19. Overall, despite the simplicity of the functionality required, the exacting reliability, performance, and security requirements, the one-off deployment, and the fact that a large fraction of the Australian population were effectively compelled to use the website, made this a high-risk IT project.

F. Agile development methodology

20. In 2013, it was reported that the ABS planned to extend the use of "Agile" software development processes throughout the organization, including for the 2016 Census[5].
21. In software engineering terminology, a software development process is something like a general-purpose recipe for turning inputs – user requirements – into the output of a working system that meets those requirements. It describes (at varying levels of detail) the tasks to be performed, who should do them, and when.

[2] http://www.businessinsider.com.au/why-the-healthcaregov-website-failed-at-launch-in-one-slide-2013-11

[3] http://www.news.com.au/sport/olympics/sports-life/sevens-olympic-app-plagued-by-technical-problems-blamed-on-unprecedented-demand/news-story/5dc9aa183bb8fd3a0688affc7309c40c

[4] http://kotaku.com/pokemon-gos-launch-has-been-terrible-1783336449

[5] http://www.itnews.com.au/news/bureau-of-statistics-uses-agile-for-2016-census-366588

22. While some software development projects have no explicit process, this is a recipe for near-certain disaster on projects of any substantial size or budget.
23. Early attempts at a software development process became known as the “waterfall” approach, after a diagram from a 1970 academic paper by Winston W. Royce[6] shown below:

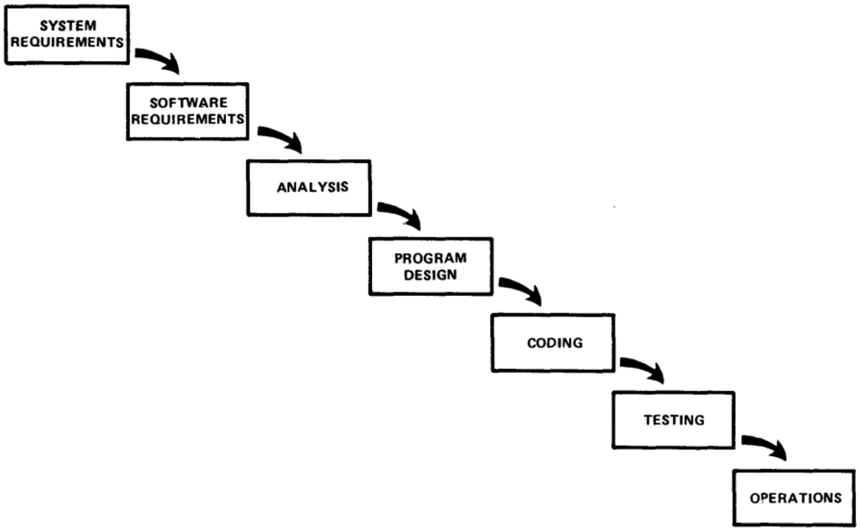

Figure 1 - the software development methodology that became known as the "waterfall model", as depicted by Henry Royce

24. In essence, to build a system using the waterfall approach, the following occurs once, in sequence:
    - A detailed description of exactly what the system is to do is produced.
    - A “plan” for how the software is to be constructed is written.
    - The software is constructed according to the plan.
    - The software is tested to give sufficient confidence that it meets the requirements as described earlier.
    - The software is then put into operation.
25. In practice, waterfall-like processes struggled to deliver software that met the needs of its users. Among other problems, initial attempts at requirements gathering almost never perfectly reflect actual requirements. Using the waterfall model, problems with the requirements are often not revealed until software is put into operation at the end of a long development process. This usually results in large amounts of rework and delays while the defects are remedied.
26. The waterfall model also reflects an era where distributing and deploying modifications to software was a long and costly process. This is often no longer the case, particularly for online systems. Updates to the software used for a website can be done multiple times on a daily basis, without users noticing any interruption.
27. In response to these issues, a number of new software development processes were devised in the 1990s and 2000s[7]. Originators and practitioners of these new methods, recognizing their

[6] Winston W. Royce (1970). "Managing the Development of Large Software Systems" In: Technical Papers of Western Electronic Show and Convention (WesCon) August 25–28, 1970, Los Angeles, USA. http://www.cs.umd.edu/class/spring2003/cmsc838p/Process/waterfall.pdf

[7] This of course greatly simplifies a rich and complex history which, while fascinating for the author, is unlikely to be of relevance to the committee.

common features, proposed the Agile Manifesto, which summarized the key distinguishing features of their approaches. The Manifesto reads:

We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:

Individuals and interactions over processes and tools
Working software over comprehensive documentation
Customer collaboration over contract negotiation
Responding to change over following a plan

That is, while there is value in the items on the right, we value the items on the left more.

(Signatories to the Manifesto omitted for brevity)

28. One of the best-known Agile methodologies, and the one adopted by the ABS, is known as Scrum. Scrum, or variations thereof, are very widely used. The key features of Scrum, as described in the Scrum Primer[8], are shown in Figure 2, and are briefly summarized below.

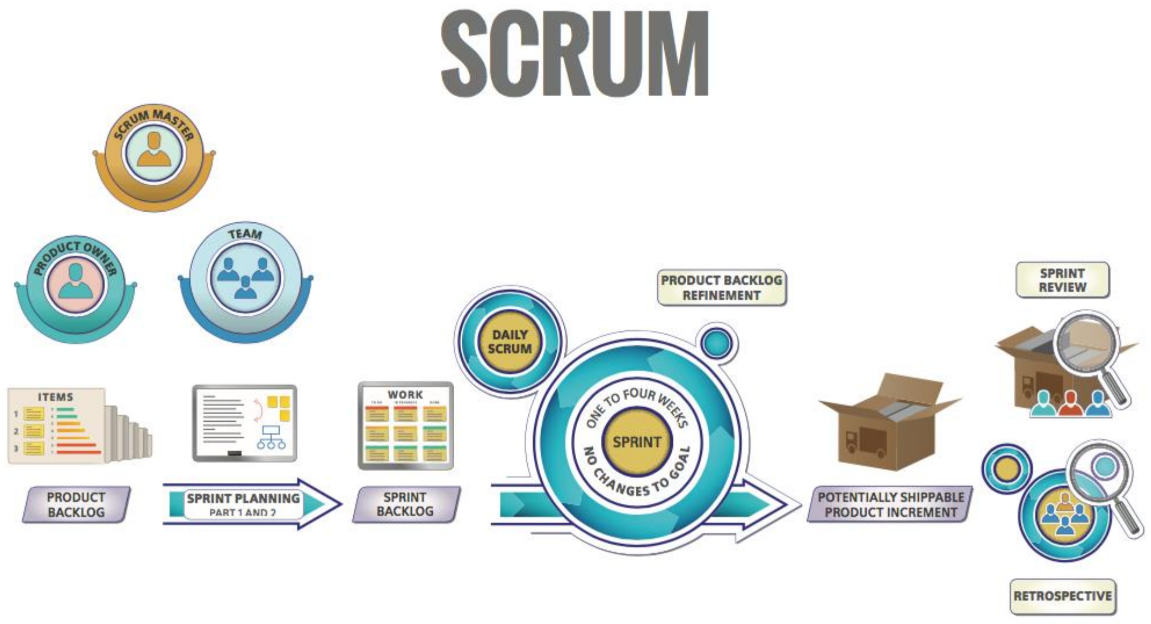

Figure 2 - Overview of Scrum. Figure from Deemer et al

[8] http://scrumprimer.org/scrumprimer20.pdf

- In Scrum, software development occurs in teams (the Team in Figure 2) of around 7 people (larger projects require collaboration across multiple such teams).
- The Product Owner is responsible for maintaining the Product Backlog – a prioritized list of improvements to be made to the current state of the product. Where and how the Product Owner gets information from stakeholders to inform that list is not defined by the Scrum methodology. The Product Backlog is a living document subject to regular change. In Scrum, the Product Owner must be *one person*.
- The Team is responsible for building the software product that the Product Owner indicates.
- The Scrum Master helps both the team and Product Owner apply the principles of Scrum. They are not a team manager in a traditional sense; the Scrum Primer describes their role as "removing impediments, protects the Team from outside interference, and helps the team to adopt modern development practices."
- Over many iterations ("sprints") of between one and four week's duration, the Team tackles as many of the highest-priority items on the product backlog as they believe that they can complete. At the end of a sprint, the team delivers a "potentially shippable product increment" – that is, a new version of the product with (hopefully) all the features agreed to at the start of the sprint. This product should be really "done": the Scrum Primer states that in the case of software it should be "integrated, tested, end-user documented and potentially shippable" – a more accurate term for shippable in the modern context is "deployable".
- After the sprint Is complete, the Team, Product Owner, Scrum Master, review the product. The information gleaned can be used to further inform the development of the Product Backlog.
- A Retrospective is conducted internally with the Team and Scrum Master (and may or may not include the Product Owner). This Retrospective identifies issues with the *process* of development, and identifies opportunities for improvement for future sprints.

29. All of the activities of the "waterfall" approach to software development – requirements, design, implementing, testing, and delivery – occur in Scrum projects. However, they occur at different times, and are organized and documented in very different ways.
30. In practice, using Scrum (or any other Agile methodology) does not remove the need for some key decisions to be made early in a project. Some changes in requirements can mean large-scale changes to the existing system – for instance, if the "platform" on which the software is required to run changes. Therefore, it is also important to identify key high-level goals for the system early, so hard-to-change decisions about the solution can be made on a rational and informed basis.
31. For Scrum to work effectively a number of ingredients must be present. Teams must be composed of motivated, competent individuals with the right mix of skills for the project. Product Owners must have an excellent grasp of the needs which the product is trying to fulfil and the ability to communicate them clearly to the team. There needs to be a high level of two-way communication, based on mutual trust and respect, between the Product Owner and Team, so that the Product Owner and Team are fully aware of the consequences of decisions made.
32. Scrum also works most effectively when the product under development is made available for inspection, and preferably use, by stakeholders as soon as is possible in the project development cycle. As noted earlier, the e-Census website was only to be used to full capacity on one evening.

33. There is a considerable tension between Scrum principles, and the traditions of contract-based service provision (that are not relevant for software developed internally within an organization). Implicit in a tendered contract process is the assumption that the customer has specified all material requirements in advance, and that the supplier is free to deliver *any* product that meets those stated requirements.
34. In summary, Agile methods, and Scrum in particular, are standard methods for managing software development. However, there are a number of prerequisites for Agile methods to work well, and a number of specific challenges to applying them in government contracting work and to this project.

G. Denial-of-service attacks

35. As noted earlier, one of the key security requirements for the e-Census websites was to remain available despite reasonably foreseeable denial-of-service attacks.
36. A denial-of-service attack (DoS) is simply an attempt by some malicious entity (the attacker) to prevent legitimate users accessing a software system.
37. In the context of a network service such as a website, the simplest form of denial-of-service attack is to flood the service with traffic, thus preventing legitimate traffic from getting through.
38. A distributed denial-of-service attack (DDoS) involves many computers, all controlled by the attacker, creating the flood of traffic.
39. A denial of service attack is a criminal offence in Australia (under the Cybercrime Act[9]) and in most other jurisdictions.
40. DDoS attacks are often carried out using "botnets" – groups of computers (often poorly secured home computers) owned by third parties that the attacker has previously taken control of. Making unauthorized use of a computer, as botnet operators do, is a crime in Australia and most other jurisdictions. Despite this, botnets, some consisting of millions of computers, flourish.
41. Little technical knowledge is required to conduct a DDoS attack. "IP Stresser"[10] and "Booter" websites offer point-and-click DDoS attacks for modest payment (which can be made through a variety of electronic currencies). Again, such services are almost certainly illegal but they continue to operate.
42. Some types of DDoS attack can be very difficult to distinguish from a spike in legitimate traffic to a website. A DDoS attack on a website can be carried out by commanding a botnet to do web page requests that look very similar to those a web browser would make.
43. DDoS attacks are carried out for a variety of reasons, including simple vandalism.
44. The prominence of the Census, and the repeated assurances by the ABS that the site would function reliably, would almost certainly have been perceived as a challenge by some attackers. As noted by the Prime Minister himself[11], DDoS attacks on the census website were entirely foreseeable and predictable.

[9] https://www.legislation.gov.au/Details/C2004C01213

[10] "stresser" is a euphemism implying that the purpose of the service is to "stress test" one's own server infrastructure.

[11] http://www.smh.com.au/federal-politics/political-news/angry-bitterly-disappointed-turnbull-lashes-abs-for-census-failures-20160810-gqpsq5.html

45. According to media reports[12], the contingency plan for responding to DDoS attacks was to block all internet traffic coming from IP addresses outside Australia. While this is, in theory, a useful countermeasure, it ignored the possibility of a DDoS being conducted using machines located in Australia. According to the ABS, this is what actually occurred, though there is an alternative explanation (see section I).
46. A number of companies offer DDoS defence services which are regarded as effective against most DDoS attacks[13]. I do not know what the exact cost of employing one of these services for the e-Census would have been, but it would have been trivial in the context of the total budgeted Census cost of $330 million – or, indeed, the expected savings of $100 million compared to the previous Census.
47. Mitigating DDoS attacks, particularly those that closely mimic regular web traffic ("application layer attacks"), is also assisted by having spare capacity. While DDoS filtering services can often filter out most such traffic, substantial spare capacity is an effective additional mitigation and, again, would have been a trivial cost in this context.
48. Ensuring computer systems are sufficiently secure against attackers, including DDoS attacks, usually requires auditing and testing by independent expert third parties. It is unclear to me whether such independent experts were engaged, and whether the extent of their activities were sufficient. Revolution IT, the company engaged by the ABS to conduct load testing on the e-Census website (see Section H), has apparently performed security testing on other government IT projects, but has stated that it was not engaged to do so for the e-Census.

H. Load Testing

49. As previously noted, while the actual functionality of the Census website was straightforward, to ensure that all the functionality, performance, reliability, and security needs were actually met would have been considerably more challenging.
50. Firstly, before designing, building, and testing a system, the requirements of that system:
    - must be stated sufficiently clearly so that the system can be evaluated against them – in short, they must be *testable*;
    - and must reflect actual needs.
51. In the case of the e-Census, the key performance requirement was expressed in terms of the number of completed forms that could be submitted in a given time period. The official ABS Census Twitter account stated in response to a query that "The online Census form can handle 1,000,000 form submissions every hour. That's twice the capacity we expect to need."[14]
52. While IBM was the primary contractor for the e-Census website, the load testing was conducted by another company, Revolution IT. Revolution IT has provided load testing services for a

[12] http://www.abc.net.au/news/2016-08-10/what-is-a-ddos-attack/7712372

[13] For example, Cloudflare (https://cloudflare.com), Incapsula (https://incapsula.com), and Akamai (https://www.akamai.com/us/en/cloud-security.jsp)

[14] https://twitter.com/abscensus/status/755588601656725505

number of government projects. In media reports[15], they stated that they had tested the site to comfortably exceed that level of performance:

"Director Hamish Leighton told the Herald Sun the ABS and IBM had expected a peak of 250 Census forms to be submitted per second.

His company tested to make sure the site could handle close to double that rate."

53. 250 forms per second equates to 900,000 per hour, and 500 forms per second equates to 1.8 million forms per hour.
54. This performance requirement is, indeed, is clearly stated and testable.
55. To test this performance requirement, Revolution IT would have used a network of computers to simulate form submissions from web browsers at the required rate. Software tools to simplify this process are widely available.
56. It is entirely plausible that this load testing was performed as described by Mr. Leighton, and the results were as he described to the Herald Sun.
57. However, it is open to question whether the performance requirements of the system reflected the actual needs.

I. Load Estimation

58. The basis on which the ABS decided on their capacity estimates has not, to my knowledge, been made public.
59. I, along with many others, have made estimates that call into question whether those capacity requirements were appropriate.
60. Firstly, even the average load on the e-Census website on Census night could plausibly have been higher than the ABS expected. As I put it in my article for *The Conversation*: "…around 18 million Australians live in the eastern states, which equates to about 7 million households. If even 50% of those households attempted to submit their census during the evening hours from 7pm to 9pm, that would equate to 1.75 million form submissions per hour".
61. Secondly, the peak loads on a website over periods of seconds or minutes are likely to be well in excess of that average. Website traffic has been known since the 1990s to be "bursty", or, in statistical terms, "self-similar"[16]. A consequence of this is that website traffic peaks are far higher than average levels, and this "peakiness" can be observed when measured over both relatively short and longer time periods.
62. In many cases, when a website is overloaded and inaccessible, people will seek alternative sources for the information or service they attempted to access, or try again after a substantial period. In the case of the e-Census website, this would not have been the case – most people

[15] http://www.heraldsun.com.au/news/australian-census-2016-melbourne-it-firm-revolution-it-defends-involvement-in-censusfail/news-story/4d89a18337bfdb4d59e3b5cfe9dfaad7?nk=bd6b5d031c80afb88fdad12909a333ed-1472815866

[16] Explaining World Wide Web Traffic Self-Similarity, Mark E. Crovella and Azer Bestavros

http://www.cs.bu.edu/fac/crovella/paper-archive/self-sim/paper.html would have repeatedly tried to access the site, either for initial access, or to complete the submission of their Census form. As such, unmet demand would have built up like water behind a dam.

63. If the load on the servers exceeded their maximum capacity, this would have appeared very similar to, and had identical effects to, some forms of denial-of-service attack.
64. If the servers were already close to, or at, their maximum load, even a small denial-of-service attack could have been the straw that broke the camel's back and caused an outage.

J. Inadequate system testing?

65. A full official account of the events of August 9th is not yet available. However, IT journalist Patrick Gray has produced the most detailed account available in the public domain.[17] While Mr. Gray's sources are anonymous, his account of events has been widely circulated and no information has emerged to contradict it. Therefore, it seems a reasonable basis for the committee to pose on-the-record questions to interested parties. In the unlikely event that the committee has not already done so, I would recommend that they speak to Mr. Gray directly.
66. Mr. Gray's account details a series of technical and human failures which I will not bother to repeat here. The committee will undoubtedly endeavor to investigate his account in detail.
67. However, two points in Mr. Gray's account do raise concerns about a lack of system-wide testing and preparedness to deal with contingencies.
68. The first relates to a contingency plan:

> *They hadn't synced the ruleset when they rebooted the firewall so the secondary was essentially operating as a very expensive paperweight. This resulted in a short outage.*

69. In lay terms, Mr. Gray's account states that the e-Census website relied on a device called a router. Because this represents a single point where failure could render the website inaccessible, they had a backup router. However, the backup router had not been configured to behave as the original router had, making it, as Mr. Gray put it, "an expensive paperweight".
70. It is a dictum of IT operations that until a backup is demonstrated to work, it is worse than useless as it provides the illusion of security.
71. Therefore, it seems reasonable to ask whether their backup router was ever tested before Census night.
72. Secondly, Mr. Gray's account suggests that the final outage was the result a deliberate decision to shut it down. This occurred after routine status messages were misidentified by an automated monitoring system as an attempt to "exfiltrate" – that is, smuggle illicitly accessed data out of the system.
    - *Some time later IBM's monitoring equipment spat out some alerts that were interpreted by the people receiving them as data exfiltration. Already jittery from the DDoS disaster and wonky firewalls, they became convinced they'd been*

[17] http://risky.biz/censusfailupdate

owned and the DDoS attack was a distraction to draw their focus away from the exfil.

- *They pulled the pin and ASD was called in.*
- *The IBM alerts were false positives incorrectly characterizing offshore-bound system information/logs as exfil.*

73. If a full test of the entire system, including the security monitoring systems providing the alerts, had been conducted, prior to Census night, the false alarms caused by the status messages should have occurred under those test conditions. There would have been ample time to reconfigure the monitoring system to identify the status messages as such and not emit alerts.
74. Full system tests can be done by system developers; if they are performed by or on behalf behalf of a client they are known as "acceptance tests". The committee should seek further information on the details of the acceptance testing performed on behalf of the ABS.

K. Conclusions

75. From my perspective, and even considering only the issues with the Census website, #CensusFail was indeed, as Jen Dudley-Nicholson put it, an "omnishambles"[18]. It appears that a large number of mistakes were made that led to the website outage.
76. The failure to have sufficient DDoS protection was, as widely acknowledged, inexcusable. The ABS, as the customer, should have insisted upon it, as should IBM as the contractors, and given the stakes both parties should have sought independent third-party assessment that the DDoS protection was adequate. Undoubtedly, the committee will seek to find out whether they did so.
77. There are reasonable grounds to question whether the site's capacity to handle legitimate traffic was adequate, and it is possible that this inadequate capacity planning contributed to the difficulties on August 9th. The committee should seek more information as to the basis for the ABS's confidence in their capacity planning, and investigate whether a peak of legitimate traffic, instead of a DDoS may have contributed to the difficulties the website encountered.
78. If Mr. Gray's reports are accurate, it is also reasonable to question whether sufficient system testing by IBM, and acceptance testing on behalf of the ABS, was conducted.
79. The ABS's competence as a purchaser of software is clearly open to question. Adoption of agile practices does not remove the need for a customer representative who understands their requirements deeply, and who can work with contracting parties effectively to realize them.
80. However, the primary contractor's competence and practices are also worthy of examination.
81. IBM's contracting division has been responsible for many Australian public sector IT projects. However, it is also associated with arguably the largest government-sector IT failure in Australian history, the debacle of the Queensland Health payroll system.

[18] http://www.dailytelegraph.com.au/rendezview/census-2016-australia-has-become-a-global-joke/news-story/0491d8ffd72da480da27483c39e5043b

82. The Board of Inquiry report[19]into that failure was 225 pages long. However, to attempt to summarize, the report describes a customer who had little idea what they actually needed, a fixed deadline, and a contractor who delivered (approximately) what they were asked for who did not concern themselves in any way with what was actually needed.
83. This committee may wish to consider whether traditional tender and contracting processes are adequate in the context of:
    - Agile processes (which the ABS is right to attempt to employ given their success in other organizations),
    - inherently hard-to-define project requirements,
    - and vendors who can maximize profits by delivering systems that may meet articulated requirements but nevertheless fail to meet actual ones that they could have reasonably foreseen.

L. About the Author and Disclosure Statement

84. I am a Lecturer in Software Engineering at Monash University, Melbourne, Australia. Software Engineering is "…application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software; that is, the application of engineering to software"[20] My primary research area is software testing - the systematic exercise of a software system to evaluate some aspect of its quality (usually with the goal of removing any weaknesses found). I teach a broad range of software engineering topics at an undergraduate and graduate level.
85. I have written a number of articles for *The Conversation*[21] on a variety of software engineering related topics for a general readership. I was asked by the editors before Census night whether I would be interested in writing an article explaining events if the Census website failed. I thought that this was unlikely, but agreed. The editors turned out to be prescient and my article[22] was the result.
86. For the record, I have recently become a member of the Australian Greens; however, the events surrounding the Census did not play any part in this decision. I am also an acquaintance of Ms. Kelly O'Dwyer, who was the Minister responsible for the Bureau before the 2016 election. However, I have not seen Ms. O'Dwyer in some time and have never discussed the Census with her.
87. I have no connection with the Australian Bureau of Statistics, IBM, or indeed any other contracting party to the best of my knowledge.

[19] http://www.healthpayrollinquiry.qld.gov.au/__data/assets/pdf_file/0014/207203/Queensland-Health-Payroll-System-Commission-of-Inquiry-Report-31-July-2013.pdf

[20] IEEE Computer Society Systems and Software Engineering Vocabulary https://pascal.computer.org/sev_display/index.action

[21] https://theconversation.com

[22] https://theconversation.com/census-website-cracks-after-malicious-attack-by-hackers-63734