Credible Election Software - eVACS®

Clive Boughton PhD
Senior Lecturer, Australian National University
Director, Software Improvements Pty Ltd

and

Carol Boughton MSc, FAICD
Managing Director
Software Improvements Pty Ltd

The Beginning

In late 2000 the Australian Capital Territory Electoral Commission (ACTEC) issued a Request for Proposal (RFP) for a trial version of an electronic voting, counting and data entry system to be used for the October 2001 ACT Legislative Assembly election. The project to develop the system, to be the first electronic voting and counting system to be used in Australia, began in March 2001. To obtain further information and history of the ACT election system, visit http://www.legassembly.act.gov.au/education/factsheets/fact06/fsb06.htm.

The winning submission (a joint bid from Linuxcare and Software Improvements) was chosen because it **offered project and quality management together with an option for transparency** through open source under the GNU GPL. Pricing had been set by the ACTEC before bidding commenced. A further constraint specified in the RFP was that the delivered system should not be internet-based.

As Software Improvements role was to provide project and quality management, when at the point of contract negotiation Linuxcare (now Turbolinux) withdrew from the ACT, the ACTEC agreed to Software Improvements continuing to run the project. Former Linuxcare people were contracted by Software Improvements to continue to develop a prototype of the voting part of the system.

Requirements – integrity and working features

Ensuring integrity
• *Starting with an empty ballot box*
• *No ballot box stuffing – no virtual voters*
• *What you see is what you get*
• *No Trojan horses*
• *No peeking before time*
• *No less secure than paper-based system*

Working features
• *Eliciting detailed requirements*
• *Scenarios defined the scope of work*
• *Open source for transparency*
• *Not open source development*
• *Independent auditing of code*
• *ACTEC could cancel project at anytime*

Originally the ACTEC had thought that an electronic system would allow for a higher integrity election system than the incumbent paper ballot system that had been in operation since the ACT gained the right to self-government in 1989. The Software Improvements team pointed out that despite the impression of greater integrity, an electronic voting and counting system could easily be manipulated unless proper care and check procedures were put into place.

Some of the questions put to the ACTEC were:

- How do you ensure that the electronic equivalent of the ballot box is empty when voting begins?
- How do you ensure that the electronic equivalent of the ballot box hasn't been stuffed with bogus votes?
- How can you be sure that what an elector sees displayed on his/her electronic equivalent of a ballot is actually what goes into the electronic equivalent of the ballot box?
- How can you be sure, that whilst ballots are going into the electronic equivalent of a ballot box, that there's not a Trojan horse changing the contents of the electronic ballot box?

- How can you be sure that the contents of the electronic equivalent of a ballot box are not being read and published to nefarious parties before counting of the votes commences (this question is more pertinent to an internet-based election system)?

There were also many questions surrounding "normal" and "unusual" voter behaviour, such as forgetting to place their vote in the ballot box – thus leaving an opportunity for a subsequent voter to vote twice.

Ultimately the ACTEC agreed that **the base requirement should be that the electronic election system should be no less secure than the incumbent paper-based one**.

Whilst the RFP submission contained the essential, feasible ideas for an electronic election system, there were nonetheless many questions that needed to be asked of the ACTEC – especially surrounding how they expected the system to be used by voters and election officials. To elicit this information (requirements) the ACTEC Commissioner and Deputy Commissioner agreed to undertake (with guidance) a process of scenario development. For a mere two days effort from three people, the resultant clarification proved key to ensuring the system delivered what was needed. Software Improvements not only obtained detailed scenarios of operation but also important data that needed to be stored, protected or manipulated (in accordance with the Hare-Clark counting rules).

Software Improvements also obtained very clear agreement as to the actual voting process and the various display screens that represented electors' progress with their vote. The client's understanding of voter behaviour was invaluable in identifying the necessary messages for guiding electors and providing a clear status on their voting progress to election officials (without exposing voting information) when they were asked to be involved. Undertaking the scenario exercise accurately defined the scope of work for both the client and Software Improvements.

During the scenario process the contract requirement of open source was discussed. The ACTEC wished to provide as much transparency as possible in order to avoid potential political ramifications that might be had if the source for the election system software was kept secret. Software Improvements agreed with the sentiment but were somewhat concerned about protecting intellectual property. Also, the software product was not intended to be free in the sense of no payment. Additionally, Software Improvements did not wish to conduct an open source development of eVACS® because it was considered infeasible in implementing a full lifecycle approach to the development. In the end the prototype was proffered as an open source development, but that the mainstream development was "closed" with the final source code, excluding analysis and design models, test procedures and cases published under the GNU GPL.

An agreement was also made between the ACTEC and Software Improvements for the final delivered code to be audited by a qualified and independent auditor and for the code to remain unchanged once the audit process was complete.

Contractually the ACTEC could have cancelled the project at any time if it was felt that the negative political or media influences were becoming too great or embarrassing.

High Points

- *Few changes made to scenario document*
- *Contract programmers productive early*
- *Counting system demonstrably more accurate*
- *ACTEC convinced of accuracy and reliability*

There were very few changes made to the scenario documents – the requirements were therefore stabilised very early in the project. This was important considering that the development time had been shortened by 6 weeks in a 33 week schedule due to protracted negotiations arising from the demise of Linuxcare.

The analysis and design products enabled very quick productivity of contracted programmers (week 13).

Except for the first time during testing, the counting system proved to be more accurate than hand counting of test data votes provided by the ACTEC. The failure on the first test was the result of an ambiguity in the specified counting process described by the ACTEC for the multi-member, preferential system used in the ACT. The ACT Electoral Commissioner's belief that a properly developed and maintained electronic election system must be far more accurate and reliable than a human-based system, was vindicated.

Software Improvements subsequently undertook enhancements to the system for the 2004 election.

Information and Statistics

- *Full software development life cycle*
- *Supported 12 languages*
- *Audio for vision-impaired*
- *Effort of 3.45 person years*
- *22K source lines*
- *Budget of AUD200K*
- *16,000 electors for trial*
- *180,000 paper votes entered*
- *One candidate elected with 55 vote margin*

The development process consisted of:

- System development planning.
- Requirements elicitation and confirmation (with review)
- Requirements analysis using a real-time structured approach (with review)
- Structured design including module descriptions with pre- and post-conditions (with review).
- Coding (mostly 'C') with unit and integration testing (with inspections).
- System/acceptance testing with test cases based on agreed requirements (with review).
- Configuration management with CVS (some problems here)
- Deployment on a strongly configured version of Debian linux (separate, bootable CDs for client and server).
- Independent audit of final code.
- Post mortem with customer and temporary polling place officials.
- Visit http://www.softimp.com.au for further information.

eVACS® supports multiple languages/scripts, 12 languages/scripts are specified by the ACT Government, and provided audio allowing vision impaired electors to vote secretly for the first time in Australia.

The total effort to produce eVACS® was 3.45 person years for a code size of 22,000 NCNB SLOC. The budget of AUD200,000 represented (at the time in 2001) approximately USD100,000.

16,000 electors used the electronic voting part of the system, with 180,000 votes counted after paper ballots were entered, electing 17 candidates for the 3 electorates. The last elected member for one electorate was in by a close margin of 55 votes.

Trust or Apathy

- *Do democratic election systems need to be of high integrity to gain elector trust?*
- *Electors generally know little of their election systems - trust is implicit*
- *Does open source ensure trust?*
- *Is open source necessary when trust seems implicit*
- *Trust in an election system when there's evidence of low integrity is apathy*

Software Improvements has demonstrably done a professional job in developing eVACS®. The ACTEC published the delivered source code for the purposes of transparency. Both Software Improvements and the ACTEC believe that democratic, secret government elections (electronic or otherwise) involve many stakeholders and require high integrity in order to capture and maintain elector confidence/trust. Given the background of issues surrounding electronic voting in other countries at the time, one might conclude Software Improvements/ACTEC gained the trust of electors with eVACS®, as the one complaint about the system originated from a candidate who lost his seat. The ACTEC did not support his case to undertake a manual count of votes because of:

i) the checks built into the electronic and data entry systems, and
ii) the Electoral Commissioner knew from the test results and manual counts for previous elections that the result would be different and no more valid than that produced by eVACS®.

The Electoral Commissioner also used the argument that every candidate had an independent person scrutinise the data entry process before counting began and so there was plenty of opportunity for issues to be raised during that period. The data entry process was based on an independent double entry method which identified both input and ballot paper errors.

The original concerns of the ACTEC regarding possible negativity toward the introduction of electronic voting, were not reflected by the electorate. In the light of this outcome, together with widely publicised information on the happenings in Europe, UK and the USA, it would seem that most electors from democratic regions have largely assumed that their respective electoral bodies and maintained election systems (electronic or otherwise) are essentially free from interference and that the likelihood of election fraud is very low. In some circumstances, this is obviously a naive assumption – as has been demonstrated in recent elections in the USA. Few electors have scrutinised either the election systems that they themselves use or the election systems of others – except maybe to comment on differences in voting and counting methods. For example, in Australia all government election systems at the federal and state levels use some form of preferential system enabling voters to provide a preferential order to their voting choices, with counting (according to well known rules) involving the elimination of candidates with the least number of votes and their votes being redistributed according to the voters next preference, to obtain a definitive result. First-past-the-post election systems seem quite foreign and perhaps simplistic to many Australians. However, most Australians wouldn't be able to tell you the rules that are used to count votes in any of their own systems, even though the information is readily available from electoral commission websites.

These observations raise the following questions:

- Do electors from democratic regions trust the incumbent election systems, or are they merely apathetic?
- Do election system producers need to make their source code available by default in order to ensure trust?
- What's the point in exposing source code widely if electors already express trust or that they are apathetic about the election systems that they are required to use?

Perhaps the overriding principle should be for politicians and those in charge of operating and maintaining election systems for democracies, to be very careful not to give reason for electors to lack trust. Perhaps an additive principle is for electors to be vigilant concerning the possibility of election fraud.

Making the source code for an election system public probably adds little value to ensuring trust. Most electors neither wish to, nor are capable of adequately scrutinising strangely expressed (computer) language even more foreign than legalese. To them the openness is immaterial. Of course those that might add value will be capable of closely scrutinising the source code and are probably more than likely wanting to prove that a system should not be trusted, rather than the opposite. This is fine, and appropriate, but is better done before a system is deployed. Or at least that would seem the most logical approach. Surely it is a concern for electors when experts demonstrate the inadequacies of election systems before deployment but the systems are nonetheless deployed.

In elections in Maryland in 2003 (http://avirubin.com/vote/analysis/) it would seem that electors, politicians and electoral officials alike just didn't want to know about the potential inadequacies of the election system under consideration for deployment. This (to us) is apathy!

IP and Fraud

- *Releasing source code raises two concerns:*
  - *loss of IP*
  - *potential for election fraud*
- *GPL probably not adequate to meet high integrity aspects and transparency too*
- *Current open source development methods lack the auditing measures and protection required for a high integrity system*
- *Software Improvements is now developing a high integrity election system to take to the world*

Despite assumed trust and potential apathy from/of electors, it remains the responsibility of election system developers, politicians and electoral officials to maintain the integrity of democratic election systems. If part of the integrity equation is to release source code then at least two concerns need to be addressed:

1. loss of intellectual property, and
2. potential for election fraud.

To some, the first concern is perhaps a moot point as it may not be too difficult to discover "stolen" code in the relatively small election systems market. The GPL licensing is intended to protect IP and is probably adequate for doing so under a normal open source development scheme. Adding to such a moot point is the fact that there is little IP in the code when it is generated effectively from analysis/design models – especially through "translative" techniques. However, to maintain the high level of integrity required of an election system some very close controls on the system content need to be implemented in order to protect not only the IP, but also reduce the possibility of fraud – the second concern. Despite all arguments to the contrary, Software Improvements is not confident that the GPL is adequate for meeting the "high integrity" aspects of election system source code whilst maintaining transparency.

To develop a 'high integrity" system of any kind requires some proof that the development organization can be trusted. This usually means that there exists within the organization some significant capability both in terms of people, methods and auditable processes able to be employed to ensure the production of a *reliable* system upon which users can *depend* to operate, according to requirements, without failure and to consistently produce correct, accurate outputs (Neil Storey *Safety Critical Computer Systems*, Pearson 1996). Part of the process is to ensure protection of all development artefacts from interference. These statements do not imply that an open source development cannot produce such reliable systems, but rather suggest that whilst a degree of auditability and protection is usually in place in an open source development it is usually not sufficient for the development of a truly high integrity system.

Software Improvements is undertaking a new "high integrity" development of eVACS® to incorporate:

- many different types of election system (from first-past-the-post to proportional representation),
- code that is provably correct,
- even greater security against election fraud than what is in the current eVACS® system,
- provision for any number of languages/scripts and audio tracks, and
- maintenance of transparency.

It is the last of these features that will be concentrated on here.

Considerations

- *Election system software must be provably correct - is this possible?*
- *Can trustworthiness of developers be guaranteed?*
- *Meeting customer deadlines*
- *There are probably no guarantees - there'll always be someone who will try and undermine a system's integrity.*

Part of the integrity of an electronic election system is that any software deployed onto any machine being used during the election has to be provably correct at least in the sense that no interference has occurred.

Is this possible? Yes, but a lot of controls may need to be put in place and all points of risk of interference identified together with appropriate checks to identify that no interference has occurred.

The individuals developing an election system must be trustworthy in terms of their capabilities, ethics and responsibilities. This will mean that a social conscience is far more important than company loyalty. In particular, such individuals must be able to, without fear or favour, say when they are being pressured in any way by anyone within or outside the company to act contrary to their professionalism, social ethics and responsibilities.

Who'd employ such individuals? Do such individuals exist? Software Improvements would and we know such individuals exist.

What about also meeting customer deadlines? In the case of election systems having to be developed initially from scratch, this is an important factor, especially when electoral agencies have to win over governments of the day to provide the financial backing. All too often (it seems) the deadlines become much shorter than is reasonable. An open source development of an election system where short deadlines are required is potentially very risky unless the appropriate number of professional, capable and ethical individuals commit early enough.

At this stage, Software Improvements is not intending to use any sort of open source development model for the new development of eVACS®. However, Software Improvements does intend to make available the source code for the purposes of transparency but on a no code change basis.

Those election agencies that purchase the new eVACS® system for use will be licensed and permitted to deploy the code for their own needs. They will be given the source code for external/independent inspection but will not be allowed to alter the code. Only Software Improvements or its qualified representatives will be allowed to modify the code and only under a strict change control procedures combined with a mandatory external/independent inspection to certify that any changes made have not voided the original independent certification. Such a scheme is intended to reduce the prospect of fraud.

Conclusions

- *Open source development of election systems is attractive from community perspective*
- *Need significant controls to protect electors' interests*
- *Election systems need to be of highest integrity to maintain elector trust and to deflect ignorant, negative claims of impropriety*
- *Trusted systems construction needs highly skilled people who can uncover maximum number of points of failure*
- *eVACS demonstrates trust*

The idea of the open source development of an election system is attractive from the viewpoint of community involvement, but in order to protect electors' interests such a development will need significant controls to protect against loss of IP and fraud.

Software products and the corresponding development processes for democratic election systems must be of the highest integrity, as too must be the personnel within the companies and electoral offices that produce the systems and run the elections, respectively. Otherwise electors have no reason to place their trust in election results that are produced through this combination of equipment and human resources.

Part of establishing trust is transparency of code. This is not so much to increase the confidence of the elector directly but to allow those with the ability to understand the code to identify potential flaws or (preferably) give their stamp of approval.

It is not enough merely to describe the election products, and the processes used to create, maintain and operate them, to gain the confidence and trust of electors. The people, methods, and processes used to develop an election product need to be of the highest integrity, as do those who have the opportunity to inspect the source. Doing so will help to deflect ignorant claims such as:

- "They used Java – everyone knows Java is an awful language"
- "The team developing the software were obviously a bunch of hackers"
- "No software product is without defects – ergo…!"

Most of these types of criticisms are rarely backed up with evidence, but they (unfortunately) do hold political weight. Obviously the cost of checking out ill-founded criticisms needs to be avoided. To do that requires a great deal more professionalism than currently exists in all sectors of the software industry (see Standish CHAOS reports http://www.standishgroup.com/sample_research/chaos_1994_1.php and the BCS/RAE report http://www.bcs.org/BCS/News/PositionsAndResponses/Positions/complexity.htm).

Trusted computer systems have been constructed. However only by people possessing high skills, who know what trust means and who work hard to identify all potential points of failure that might cause a system to operate incorrectly. **Electronic election systems need to be trusted systems. eVACS® demonstrated that this is possible.**

References

http://avirubin.com/vote/analysis/
http://www.bcs.org/BCS/News/PositionsAndResponses/Positions/complexity.htm
http://www.legassembly.act.gov.au/education/factsheets/fact06/fsb06.htm
http://www.standishgroup.com/sample_research/chaos_1994_1.php