Richard Self
Research Fellow, Big Data Lab and Senior Lecturer in Analytics and Governance
University of Derby

Big Data Applications: Making them Deliver Value
Classic software testing often only addresses verification of software to the detailed spec.

User Acceptance Testing attempts to address Validation but is far too late in the development cycle.
Essentially no improvements in the success rates of IT related projects over the last 22 years!

Causes are holistic at the project level.

Old definitions pre- 2013

**project success:** The project is completed on-time and on-budget, with all features and functions as initially specified.

**project challenged:** The project is completed and operational but over-budget, over the time estimate, and offers fewer features and functions than originally specified.

**project failed:** The project is cancelled at some point during the development cycle.

Post 2013, **Successful projects** were redefined as being those which were on time, on budget with a satisfactory result, which is in some ways a lower bar. The change was because so many projects had previously been classified as not delivering all the initially specified functionality. So, presumably, it was assumed
that this new definition should deliver more “successful” projects. It is extremely ironical that it has, on the contrary, delivered even more challenged projects, see the comparisons for 2011 and 2012.
One of the most worrying aspects of the use of IT is the level of value destruction that it causes. At the upper boundary, this suggests that IT causes more value destruction than even the Financial Services sector caused with the 2007/2008 Credit Crunch because it continues to occur with no apparent improvement in sight for better implementation of IT related projects.

On this slide, the lower boundary for costs of IT failure is based on a very simple direct cost of the failed projects and some of the costs of security failures.

The upper boundary, estimated by Roger Sessions at http://simplearchitectures.blogspot.co.uk/2009/09/cost-of-it-failure.html, is based on a variant of Total Cost of Ownership.

Clearly neither of these two numbers is “accurate” however, they do suggest that we need to consider project risks much more carefully.
LASCAD  failure of communications systems in London leading to major problems with control of London Ambulances

Denver International  Baggage handling, failure of algorithms to be able to manage new luggage handling automated systems. Previous indications that the Operations Research techniques could not cope with the problem

Heathrow T5, failures to learn from Denver International debacle, lack of testing and training

Knight Capital, 1 Aug 2012, loss of $467M in 45 minutes

Repurposed system flag, incomplete roll-out of code to 7 out of 8 servers to replace old functionality and replace with new. 8th server retained old functionality. No implementation oversight. No effective monitors of error trades. Destroyed company.

Egg Bank  failure to have rapidly scalable infrastructure to cope with unplanned demand on first day of launch. No means of load shedding leads to complete collapse of the system

Derby Kingsway Roundabout, new large roundabout with traffic lights. Timing sequence problems leading to gridlock in the retail centre and surrounding area for two weeks. Once lights switched off, traffic flow far better than before. New high capacity roundabout turns out to be the solution, not traffic lights.
Common Themes

- Requirements capture and specification
  - Mis-match between needs and specification
  - Incorrect algorithms
  - Missing features and needs (security prime case)
- Unit test failures
- Integration test failures
- User acceptance failures
- Volume test failures
- Inappropriate test data / contexts or modelling of algorithms
- Machine Learning / AI systems
- Inappropriate HCI factors
- Modern “continuous beta standard” software
How can the Testing function move to a much earlier stage in the development cycle?

How can the disciplines be applied at stage 2 where it will have the most impact on success and value for money?
Most testing is verification that software meets the specification.

UAT attempts to validate the systems against the real needs of the users and customers but this is far too late in the process.

How can the initial requirements be validated before the expensive stages are started? This is, however, difficult due to the problems of actually achieving a completely accurate set of requirements and then an accurate and complete specification.

The challenge is, therefore, to begin to apply the techniques of testing to the Requirements capture process and the Specification development.

Whilst this is conceptually simple, it is actually complicated, even for the waterfall process. It is much more difficult for Agile and DevOps approaches. How can we apply Testing methodologies to Requirements capture and the other design processes.
Big Data brings new challenges in terms of volume and complexity of data and processing.

What are the new issues that Big Data brings to the Testing profession?
The classic 3 Vs of Big Data, each of which have significance for overall projects from identification of need through requirements capture and specification to development, testing and implementation and post implementation support.

High volume impacts storage and bandwidth

Velocity impacts storage, computational capability and processing timing, now or later?

Variety brings issues with data integration and meaning
How can testing ensure safety at all times? What is the nature of the testing.

How can we be certain that all cars have identical clones of the “tested” software, especially if each car continues to “learn” from its own experience? (The homologation issue)
History and current experience tells us that our current approaches to testing do not lead to satisfactory systems and apps.

A fundamentally new approach to the application of the concept of testing to system design, development and implementation is needed.

How effective are current requirements capture processes? How are they verified and validated? How do you ensure that they are comprehensive and complete? Because if they are not, the resultant system will not deliver the expected benefits and will result in a wide range of vulnerabilities.

Is security a late “bolt-on”? It should be at the core of the requirements capture process.

When was the last time that the HCI concept was actually tested? See J. Nielson work.

A key lesson from the Standish Group Chaos reports is that projects with “challenging” targets, timescales and resource budgets will either be challenged or will fail. To paraphrase one of their reports, “allocate at least 40 contingency to your most thorough project estimates and then project manage everything like...
crazy and you might just be lucky to be successful!”
Project Governance

More than just Verification Testing

Richard J Self - University of Derby
Governance is ..

- doing the right thing in the right way at the right time with the right resources to the right quality in the right place for the right reasons ....
If we are to really begin to address the problem of challenged and failed projects, using the Standish Group definitions, we need to totally re-think what we mean by the term “Testing”.

Software Testing experts have much to contribute to this re-think across all these aspects of system development and delivery.
This is a list of questions that can help you to gain a full understanding of many of the factors that can affect the success of your project.

See http://computing.derby.ac.uk/c/big-data-analytics-analytics-12-vs/ for more on the subject.
Value

- Is the project really business led? What are the questions that can be answered by the project and will they really add value to the organisation and who will get the benefit and what is the benefit? Is it monetary? Is it usability? Is it tangible or intangible?

- What is the value that can be found in the data? Is the data of good enough quality?

Vulnerability

Is Security designed into the system, or added as an afterthought? Major consequences leading to significant reputation damage

Incorrect processing leads to reputation damage
Big Data brings new problems which must be included in the testing process for Big Data based projects
Detecting Anomalies

- Single reading errors very difficult to detect, unless there are other data sources that can give a context
- In time series data, there may be algorithms which can be used to identify “rogue” data points, what are they and how reliable are they? How can they be tested?

Sensor calibration drift

- The calibration of all sensors drifts over a period of time. How can this be detected?
- What is the performance data that characterises this? How can this be included to improve overall accuracy?

Tech Stack failures

- Assisted GPS stack has significant problems at start-up, errors up to 1800km for first reading.
- Significant problems in a range of conditions, even in circumstances with excellent satellite visibility!!

Multi-Source data integration

- Financial services – using social media data – connection to corporate master data. How definite is the connection for “John Smith”? Massive adverse consequences for incorrect connections
- Location data, e.g. IP address used for location identification. E.g. in the
USA IP addresses without certain ownership allocated to a single Lat / Long in the centre of the USA (600M IP Addresses at the location).
Many very significant issues with machine learning

Modern **Machine Learning** technologies are extremely opaque to humans. It is almost impossible to determine exactly what the machine has learned. **Neural networks** are particularly problematic. “Faith in Technology” is significant.

1990s example of training a system to distinguish between images of friendly and enemy tanks on the battlefield. System actually learned to distinguish between sunny desert scenes and forest scenes due to the nature of the training photos.

Development of Predictive Analytics now becoming done by Model Factories (e.g. SAS) with fully mechanised processes to choose the best predictive model and algorithm. Often the retraining is repeated every 3 or 6 months with the new data collected over that period. Who is involved in proving that the new algorithms should be used (testing). What are the organisational implications of having fluid algorithms? What are the vulnerabilities to the organisation if a customer or supplier can prove that the new (or old) model would have given a more favourable decision to the one that was operational at the time of the decision.

Current assumptions seem to be that the Data Scientists will develop and train the machine learning and run the Model Factories. Is there any involvement of
the Testing experts in this process? Should they be involved?
Consequences of Agile and DevOps (1)

- Subverted Agile methodology
  - Original
    - Rapid light weight function development
    - Little documentation
    - Integrated User involvement and prototyping
  - Subverted Agile (modern)
    - Rapid(ish) heavy weight system development
    - Reams of documentation
    - Where are the Users if not co-located on the team?
Parnas’ Principle applied to humans as well as software objects
- Missing functionality and data
- Little communication between software developers, written function specification (for each “object”) is assumed to be complete and perfect, therefore no communication required or needed.
- Compare to 1970s system design and development teams, shared knowledge and understanding, continual communication

Complex Systems
- Consequences of system completeness and usability of small components

Permanent Beta Release
- Users find bugs and problems
- When will problems be fixed?
- Will new bugs be created?
- Reputation?
Consequences of Agile and DevOps (3)

- Role of Testing?
  - Verification of code to spec?
    - What is the spec? Indeed, is there a spec?
  - Validation of requirements and system and software specs?
Thank you.
Questions and Discussion

Richard J Self
Research Fellow, Big Data Laboratory
University of Derby
http://tinyurl.com/puyg6u8
http://computing.derby.ac.uk
email: r.j.self@derby.ac.uk

Richard J Self – Big Data Laboratory,
University of Derby