Continuous Validation for Data Analytics Systems[*]

Mark Staples Liming Zhu
NICTA
Level 5, 13 Garden St,
Eveleigh NSW 2015, Australia
firstname.lastname@nicta.com.au

John Grundy
Deakin University,
Locked Bag 20000,
Geelong VIC 3220, Australia
j.grundy@deakin.edu.au

ABSTRACT

From a future history of 2025: Continuous development is common for build/test (continuous integration) and operations (devOps). This trend continues through the lifecycle, into what we call 'devUsage': continuous usage validation. In addition to ensuring systems meet user needs, organisations continuously validate their legal and ethical use. The rise of end-user programming and multi-sided platforms exacerbate validation challenges. A separate trend is the specialisation of software engineering for technical domains, including data analytics. This domain has specific validation challenges. We must validate the accuracy of statistical models, but also whether they have illegal or unethical biases. Usage needs addressed by machine learning are sometimes not specifiable in the traditional sense, and statistical models are often 'black boxes'. We describe future research to investigate solutions to these devUsage challenges for data analytics systems. We will adapt risk management and governance frameworks previously used for software product qualities, use social network communities for input from aligned stakeholder groups, and perform cross-validation using autonomic experimentation, cyber-physical data streams, and online discursive feedback.

Categories and Subject Descriptors

D.2.4 [**Software Engineering**]: Software/Program Verification—*validation*; K.6.4 [**Management of Computing and Information Systems**]: System Management—*quality assurance*; K.4.1 [**Computers and Society**]: Public Policy Issues—*ethics*

Keywords

software validation, continuous development, devOps, machine learning, data analytics, ethics, governance

1. INTRODUCTION

The goal of software engineering (SE) is to ensure software-based systems meet their needs for use. Requirements engineering, design analyses, formal methods, and testing help us understand whether systems will meet those needs. However, the ultimate reality check is through usage validation: whether the usage of the system in its real-world context meets user needs. Here 'needs' include functionality, non-functional qualities, and broader ethical/legal constraints.

This paper defines a research agenda to address usage validation challenges in the nexus of two trends: the integration of usage validation into continuous development practices, and the specialisation of software engineering practices and platforms for data analytics systems. The next two sections review these trends and their usage validation challenges. Then we outline our research agenda, before concluding.

2. CONTINUOUS USAGE VALIDATION

Agile practices are now established in conventional SE. The move from phased development to continuous development became widespread for build and unit test in continuous integration. This later expanded through the lifecycle, as 'devOps' integrated IT operations with software development [4]. Agile principles encourage frequent feedback from user representatives, but initially this was only on test systems, and for batches of changes after development sprints. In recent years, usage validation is increasingly integrated with continuous development workflows, in what we call 'devUsage'. Techniques used for devUsage include performance monitoring, online A/B testing (experimentation on live user populations to assess preference between design variants), click path analysis (observations of users' navigation through web sites), online user surveys, and crowd-sourced solutions.

2.1 Usage Validation Goals

Specific usage validation goals depend on the system, the users, and their context. The main goal is that user needs are met for the core system functionality. Similarly, requirements specification databases often describe goals for non-functional properties such as performance, usability, safety, privacy and security. Usability and functionality are the major focus of UX (User eXperience) evaluation, and we mean devUsage to encompass continuous UX. However, we also mean to include broader usage validation goals covering ethical and legal issues. These considerations often inform non-functional properties such as privacy or safety, but are usually more wide-ranging and have a more fundamental

[*] This paper is written as future history from 2025. To avoid temporal paradoxes, we do not cite papers from after 2015.

ICSE '16 Companion, May 14 - 22, 2016, Austin, TX, USA

ISBN 978-1-4503-4205-6/16/05…$15.00
DOI: http://dx.doi.org/10.1145/2889160.2889207

force than the often narrow explications of user needs captured and agreed in requirements specification databases.

Software systems are critical for the allocation and delivery of services and resources, and so have potential for harm (inadvertent or otherwise). Such software systems are ethically charged, and SE professionals must ensure that these systems are consistent with the public interest [9]. Boehm [5] and others [16] have argued that continuous assessment is required to manage these ethical issues. In industries such as nuclear control, avionics, autonomic ground vehicle control, and rail, there may also be legislated regulatory compliance conditions for safety. In other industries such as banking and health, there may be compliance conditions for integrity and privacy. Even in unregulated industries, there are usually corporate governance requirements for legal compliance [2].

2.2 Impediments to Usage Validation

The real-world use of a system and SE models of it are different, and there is always a gap that cannot be bridged by purely logical analysis [19]. Empirical observation and real use testing of the system is necessary, to probe the limitations of the engineering models and support judgements about the adequacy of the system.

There are two common complications for usage validation for software-based systems. First, large systems of systems are usually highly complex with discontinuous and unexpected emergent behaviors. There is no reasonable way to validate all of these behaviours [20]. Second, these systems are typically used for 'wicked problems' [17], where users' needs can conflict with each other and can change as a consequence to using systems that address their needs. So, validation goals may be in tension and be constantly shifting.

Two emergent technologies further complicate validation.

End-user programming is a long-held dream in SE, but in recent years this dream is becoming a more wide-spread reality. We have reached a critical threshold of capability in domain-specific languages, model-driven platforms, and visual programming systems. These enable non-expert-programmers to build and maintain polished expert-like solutions, leveraging their domain expertise to express and compute with complex models. However, this flexibility presents a challenge for usage validation. What of the myriad of supported purposes are end-users working towards, and how do we ensure that their creations suit their purposes? Domain experts may not understand usage validation processes, are less likely to perform adequate validation, and may not anticipate all of the functional, ethical and legal consequences of their creations. Requirements—and users—are naturally emergent and very hard to design for.

Another trend is the rise of multi-sided platforms, which bring together and enable interactions between many parties. Traditionally, a single company would own, develop, and operate systems for their users. The platformization of services and industries means that platform owners may only provide administrative support and high-level governance. Most of the functionality on a platform is provided not by the platform itself, but by symbiotic apps or services legally owned by platform users (not necessarily the end users). Thus most of the system operation and monitoring is performed by platform users, not the platform owners. As with end-user programming, the huge space of functionality supported by the platform and instantiated by platform users creates a challenge for usage validation. However, the blended ownership between the platform and its users complicates governance responsibilities and usage validation.

3. SE FOR DATA ANALYTICS SYSTEMS

SE is increasingly specialised [12]. A clear example of this is in the development of data analytics systems[1] ('SE4ML'). Statistical machine learning (hence 'ML') lead in the integration with development practices for data analytics systems, but is now often combined with techniques from operations research and AI. As ML moved from research to widespread industrial application, there was a realisation that the bespoke algorithms written for academic publication were not necessarily scalable for large data sets nor maintainable for evolving data schemas and analysis purposes. Moreover, in industrial application there are new development artefacts to be managed, including learned statistical models, and training data sets. Since 2015, SE4ML has adapted conventional SE practices and technologies, and created new ones.

3.1 Usage Validation Goals for SE4ML

Many user problems solved by ML are only specified implicitly by training data sets, or by patterns embedded in live data streams. Such user needs are not explicitly specified, nor are they specifiable in the traditional sense. This is a disruptive challenge for conventional SE [7].

A long-standing concern is that personal data might be collected with no specific purpose, or be later reused for a different purpose. Individuals cannot reasonably provide informed consent in such circumstances. This is most obviously an ethical concern, but is also legally regulated in some jurisdictions. Data uses must be managed and auditable.

Ethical assessments require knowing the likely consequences of an action, but these user problems might never be explicitly defined nor understood. This makes ML solutions prone to "inadvertent algorithmic cruelty" [6, p. 20]. This can also be a legal risk. For example, ML can create illegal biases such as racial or sexual discrimination [15]. Technically, discrimination is the objective of machine learning, but many kinds of discrimination are legally prohibited or unethical.

Data analytics systems also have a new validation goal: model accuracy, also called statistical validity. Does a model created by ML really reflect the situation in the world? When reusing data, is sample population and data collection instruments that were used still appropriate? Accuracy is fundamental to validating user needs, but is also critical for ethical assessment and legal probity. Validating model accuracy can be complicated by difficulties with interpretation. In statistics, Simpson's paradox [14] is a well-known example where associations between variables can be reversed under different groupings. These threats can defeat validation.

3.2 Validation Impediments Under SE4ML

As discussed above, usage needs in data analytics can be unspecified or not understood. ML models emerge by mixing selected details of training data sets, and can implicitly encode selection biases of those data sets. Even for well-defined problems, the models are often inscrutable 'black boxes'. Whether or not the solution satisfies the need cannot be readily assessed by inspection and analysis of the model, but only by testing on representative cases. (Some ML models, e.g. decision trees, can be inspected.)

[1] These used to be called 'big data analytics systems'.

Many data analytics systems are based on online learning, where the models are continually updated using live data. These can initially be tested with carefully curated training sets. However, ongoing validation challenges remain—not only are the ML models continually changing, but the problem itself may be changing. As models are complex black boxes, it also unclear whether the learned models adequately reflect data about current problem situations, or are merely the ghosts of old training data sets.

Software reuse and abstraction are fundamental for effective SE, and for end-user programming. However, they are a challenge for ML [18, 7]. When different models are created for related sub-problems in a broader software system, the consistency of those models can depend on hidden relationships implicit in their training data sets, by the emergent functions that happen to be learned by the models, and by the contexts that those models are used in within the system. Even if an individual model is sufficiently valid, the combination of models may not be.

4. RESEARCH AGENDA

This section outlines our plans to investigate solutions to the devUsage and SE4ML challenges described above.

4.1 Risk-Based Governance Networks

To systematise devUsage we will adapt risk management (RM) and governance frameworks. RM is a way to identify, assess, mitigate, and monitor potential problems, and is standard in SE project management. However, it is also used in SE for product quality risks [8], ethical risks [5, 16], and legal compliance risks [2]. Continuous risk assessment is recommended within RM and for SE [5, 16], so is a good fit for continuous usage validation.

Although RM seeks input from relevant parties, it is often performed separately by each stakeholder. Governance frameworks instead focus on collective problems [3, 11]. Many data analytics systems are internet-connected multi-sided platforms. Most stakeholders for these systems are present as groups on public or enterprise social networks. We will investigate the use of these groups as loosely-coupled 'governance cells' [3] or governance 'nodal points' [11]. These are loci of interaction for these stakeholder groups to perform usage validation activities. Each cell will be provided with support systems to use RM as a framework to manage threats to user needs and social norms in use of the system. We hope to improve overall governance by sharing risk information and usage validation knowledge within the network. In particular this is expected to help to identify usage needs in conflict across stakeholder groups.

Conventional ethics governance uses a consensus of ethicists to define principles for system usage [1]. For multi-sided systems, it may not be clear to every stakeholder that a central group is trustworthy and recognises their stake. Instead, we will manage the validation of ethical principles and policies independently through separate stakeholder groups. Using shared RM information, we will explore approaches to promote the reuse, harmonisation, and cross-validation of ethics-related principles and policies across the groups.

Some key stakeholders (e.g. statutory regulatory bodies) may not have social networking groups to represent their opinions. So outputs from their traditional committee work must be integrated with outputs from social networking and crowdsourcing in the governance network.

4.2 Instrumentation and Governance Support

Effective governance for data analytics systems needs machine support for instrumentation and partial automation. This functionality has its own requirements, implementation, and validation and gives rise to performance overheads in the use of the primary system [13], which can in turn affect usage validation of that system. Our research will identify requirements for governance support systems, and develop prototype solutions for evaluation. Requirements are expected to cover policy definition, monitoring, reporting, and alerting, but also include support for the operation of governance networks (section 4.1), and integrating cross-validation activities (section 4.3).

4.3 New Usage Validation Techniques

No single approach can provide unequivocal usage validation. Just as with qualitative research methods, we need to use cross-validation, or 'triangulation', to combine different kinds of studies of individual validation goals. For example, to validate performance goals, we could combine system performance monitoring with observational studies and online questionnaires. We will also include trials of usage validation techniques enabled by new technologies, as below.

4.3.1 *Autonomic Experimentation*

The last decade has seen model-driven e-science become more widely used, especially for data sharing, data integration, and provenance. We will tailor these models to represent validation goals, validation techniques, and the results of validation studies. By integrating these models with ongoing work in computational creativity [10], we plan to explore the automatic creation of usage validation hypotheses, and the automatic design of usage validation experiments to be run within stakeholders' social networks groups. Even if statistical models are inscrutable, it seems likely a case-based approach can be used to define and monitoring ethical rules [1] and principles for usage needs. These will not fully define system usage, but can be explicit simple tests, and be derived from abstract ethical principles or usage goals identified in governance cells. Autonomic experimentation cannot provide fully automatic governance, but is expected to provide machine support for the governance network.

4.3.2 *Cyber-Physical Data Streams*

There are now massive data streams flowing from cyber-physical systems. The most common sources of these are from the Internet of Things and Augmented Reality interaction data streams. Third party data streams support various functional purposes, but we can also use them to characterise physical situations associated with the usage situations of our own data analytics systems. The needs satisfied by our data analytics systems often have a physical manifestation, and observing this and the usage context can help in cross-validation. For example, validation of shopping recommendation systems requires user feedback about bought products, but this feedback may be best gathered during or shortly after the physical use of those products. A challenge is to identify relevant physical contexts, but this is a core capability of Augmented Reality systems, and so they will form part of our research.

4.3.3 *Online Discursive Feedback*

Personalised natural language dialog systems from Apple, Google, Microsoft, and SoundHound are now embedded in the activities of daily living of millions of people. Increasingly these systems are offered as customizable platforms for dialog services. We will use these to capture live feedback about the adequacy of system usage. We expect this to be useful for feedback on user needs. For example, validation of shopping recommendation systems also requires user feedback about the timeliness of recommendations, and this may be able to be gathered with online spoken dialog systems. However, dialog systems can offer online help about concerns or risks related to the use of the system which we can use to identify and validate ethical risks.

As well as live discursive feedback, we may be able to use automated structured interviews with stakeholders to help inductively define ethical principles (or other validation goals) by reference to hypothetical cases [1].

5. CONCLUSIONS

The value of a software system comes from solving real-world problems faced by its users. Systems might be designed to meet those needs, but users' needs are not always well understood and inevitably change over time. It is critical to continually validate the usage of a system, for its main functional requirements, but also to ensure that the system is ethical and legal. For data analytics systems, continuous usage validation is complicated by numerous challenges, some of which break fundamental assumptions for conventional SE. Some ML problems are defined only by training data, and are not specifiable in the conventional sense. ML models are usually black boxes which we cannot meaningfully inspect for static analysis or test coverage. Because abstraction is hard for ML, even if individual models are separately validated, their combination may not be jointly valid nor valid within a context of use that does not match their training data. Nonetheless, we have identified some approaches that may partly address these challenges. We will investigate validation techniques using emerging technologies including autonomic experimentation, augmented reality, and online discursive feedback platforms. To integrate and cross-validate results from these and traditional approaches, we will use social networks to support a network of loosely-coupled usage governance cells, sharing RM and validation knowledge between stakeholder groups.

6. ACKNOWLEDGMENTS

Mark Staples and Liming Zhu are also affiliated with the School of Computer Science and Engineering, UNSW. John Grundy is also affiliated with NICTA. NICTA is funded by the Australian Government through the Department of Communications and the Australian Research Council through the ICT Centre of Excellence Program.

7. REFERENCES

[1] M. Anderson and S. L. Anderson. Toward ensuring ethical behavior from autonomous systems: a case-supported principle-based paradigm. *Industrial Robot: An International Journal*, 42(4), 2015.

[2] S. M. Bainbridge. Caremark and enterprise risk management. Law-Econ Research Paper 09-08, UCLA School of Law, 2009.

[3] P. L. Bannerman. Software development governance: A meta-management perspective. In *Proceedings of the 2009 ICSE Workshop on Software Development Governance*, pages 3–8. IEEE Computer Society, 2009.

[4] L. Bass, I. Weber, and L. Zhu. *DevOps: A Software Architect's Perspective*. Addison-Wesley, 2015.

[5] B. W. Boehm. Value-based software engineering: Seven key elements and ethical considerations. In *Value-based software engineering*, pages 109–132. Springer, 2006.

[6] G. Booch. All watched over by machines of loving grace. *IEEE Software*, 32(2):19–21, 2015.

[7] L. Bottou. Two big challenges in machine learning, 2015. Slides presented at *International Conference on Machine Learning*, 6–11 July 2015. Available at: http://icml.cc/2015/invited/LeonBottouICML2015.pdf.

[8] Y. K. Chiam, L. Zhu, and M. Staples. Quality attribute techniques framework. In *Software Process Improvement*, pages 173–184. Springer, 2009.

[9] D. Gotterbarn, K. Miller, S. Rogerson, S. Barber, P. Barnes, I. Burnstein, M. Davis, A. El-Kadi, N. B. Fairweather, and M. Fulghum. Software engineering code of ethics and professional practice, 2001.

[10] K. Grace and M. L. Maher. Using computational creativity to guide data-intensive scientific discovery. In *Workshops at the Twenty-Eighth AAAI Conference on Artificial Intelligence*, 2014.

[11] M. Hufty. Investigating policy processes: the governance analytical framework (GAF). *Research for Sustainable Development: Foundations, Experiences, and Perspectives*, pages 403–424, 2011.

[12] M. Jackson. Specializing in software engineering. *IEEE Software*, 16(6):119–121, 1999.

[13] Y. Liu, L. Zhu, L. Bass, I. Gorton, and M. Staples. Non-functional property driven service governance: Performance implications. In *ICSOC 2007 Workshops*, pages 45–55. Springer-Verlag, 2009.

[14] J. Pearl. Understanding Simpson's paradox. *The American Statistician*, 68(1):8–13, 2014.

[15] D. Pedreshi, S. Ruggieri, and F. Turini. Discrimination-aware data mining. In *Proceedings of the International Conference on Knowledge Discovery and Data Mining*, pages 560–568. ACM, 2008.

[16] A. Rashid, K. Moore, C. May-Chahal, and R. Chitchyan. Managing emergent ethical concerns for software engineering in society. In *International Conference on Software Engineering (ICSE)*, volume 2, pages 523–526, May 2015.

[17] H. Rittel. On the planning crisis: Systems analysis of the 'first and second generations'. *Bedriftsøkonomen*, 8:390–396, 1972.

[18] D. Sculley, G. Holt, D. Golovin, E. Davydov, T. Phillips, D. Ebner, V. Chaudhary, and M. Young. Machine learning: The high interest credit card of technical debt. In *SE4ML: Software Engineering for Machine Learning (NIPS 2014 Workshop)*, 2014.

[19] M. Staples. Critical rationalism and engineering: ontology. *Synthese*, 191(10):2255–2279, 2014.

[20] W. A. Wulf. Keynote address. In *Emerging Technologies and Ethical Issues in Engineering: Papers from a Workshop*, pages 1–6. The National Academies Press, 2004.