Understanding Quality Requirements in the Context of Big Data Systems

Ibtehal Noorwali, Darlan Arruda, Nazim Madhavji

University of Western Ontario, London, Canada
Introduction

• With the advent of Big Data, its “V” characteristics are posing challenges w.r.t system security, performance, scalability, privacy and other quality requirements (Suwalska et al., 2013; Madhavji et al., 2015).
Introduction

• Among the challenges is how to handle quality issues and big data characteristics in the specification of requirements.

• Proposed idea:
  A specification technique that integrates:
  (i) traditional quality attributes (e.g., performance, reliability, availability, etc.) and
  (ii) big data characteristics
  in a unified requirement description.
The Proposed Idea

• The integration would permit various permutations of big data characteristic and quality attribute.

• Example:
  – variety $\times$ security
  – velocity $\times$ performance
  – veracity $\times$ privacy
  – volume $\times$ scalability

• The full requirement description must specify the desired permutation.
Example

• A velocity $\times$ performance requirement:
  
  – The system shall use a stream-processing engine with a latency of 0.5 – 2.0 seconds (e.g., Storm, S4, Spark or Samza) to process data in real-time between global earthquake sensors and the data centre.

• In the context of safety critical systems (e.g., earthquake monitoring) performance is key.
  
  – The suggested stream-processing engines handle high velocity, real-time data while offering low latency and high throughput (i.e., performance).
Issues to Consider

1. A big data characteristic can intersect with more than one quality attribute.

- **Example:**
  A requirement that specifies the *veracity* (i.e., data consistency, trustworthiness, and integrity (Demchenko et al., 2014)) of the data in a big data system through *security* requirements.

  - The system must implement *security* mechanisms (e.g., encryption, authentication, etc.) to maintain the *veracity* of the data.
  
  - However, some big data security mechanisms may be more computationally expensive than others.

    - Thus, *Performance* becomes an issue for this “*veracity × security*” intersection
Issues to Consider

• It results in the following configuration: $\text{veracity} \times \text{security} \times \text{performance}$.

  – The system shall:
    • encrypt, query, and decrypt [security]
    • 50000 tweets in 2 seconds [performance]
    • using CMD (Computing on Masked Data) [veracity]
Issues to Consider

2. It is possible that a particular permutation (e.g., variety × availability) may not be valid.

– For example: when availability of the system is not affected by the variety of the data processed by the system.

– Caution: too many NULL intersections may indicate possible incomplete or missing quality requirements
Issues to Consider

3. A permutation may be more frequent in one domain than in another.

– Example: the *velocity* × *performance* permutation may be more relevant in time-critical big data system than a *variety* × *privacy* permutation.
4. **SRS structuring:**

Each permutation can be considered a *header* in a SRS that can include one or more quality requirements under it.

– **Potential benefit:** ease of understanding Big Data requirements in the midst of other system requirements.
Discussion Points

i) A **systematic approach** for handling big data characteristics and quality requirements.
   – In turn, this may **help to reduce** missing quality requirements while accounting for big data characteristics.

ii) **SRS structuring:**
   – Traditionally, SRS has headers for quality attributes (usability, performance, reliability, etc.)
   – In the big data context, the permutations described can serve as additional headers in a SRS. Ease of understanding.

iii) **Flexibility:**
   – Permutations allow zero or more quality requirements.
Implications for Research

• A conversation starter for discussing big data characteristics in requirement processes.

• Precipitate research on elicitation, specification, analysis, validation, and management of big data system requirements.
Implications for Practice

• **Improve specification** of big data and quality requirements; in turn, system quality.

• **Role of a requirements analyst:**
  – expected to be knowledgeable on big data technologies, tools, libraries, methods, and reference architectures and not just RE practices.
  • Without such knowledge, an analyst c/would be challenged in addressing big data requirements.
Future Work

• Empirical studies to better understand the utility of the approach, generalisability, and scalability.
Thank you!
References

