PIS and Introduction

Quantifying Requirements Technical Debt: an Anonymous Online Questionnaire for Practitioners

Participant Information Sheet (PIS)

Project Title: 'Quantifying Requirements Technical Debt: An Anonymous Online Questionnaire for Practitioners'

Research Team:

Associate Professor Ewan Tempero (Principal Investigator/supervisor), School of Computer Science, Faculty of Science, The University of Auckland

Judith Perera (Doctoral researcher), School of Computer Science, Faculty of Science, The University of Auckland

Associate Professor Kelly Blincoe (co-investigator/supervisor), Department of Electrical, Computer and Software Engineering, The University of Auckland

Dr Yu-Cheng Tu (co-investigator/supervisor), School of Computer Science, Faculty of Science, The University of Auckland

Researcher Introduction: I am Judith Perera, a Doctoral researcher at the School of Computer Science, Faculty of Science at The University of Auckland. My supervisors are AP Ewan Tempero, AP Kelly Blincoe, and Dr Yu-Cheng Tu. We would like to invite you to take part in our **anonymous online questionnaire**. Before deciding to participate, you need to understand the purpose of this study and how you will be involved as a participant. Please take time to read the following information carefully. Reach out to us if you need further clarifications (see contact details at the bottom) – these correspondences will not be associated with the questionnaire responses.

Purpose of the study: Software Requirements are crucial in developing a software product. Software practitioners can make sub-optimal decisions (either deliberately or inadvertently) when dealing with software requirements while performing their tasks.

For example, requirements engineers or business systems analysts might not capture essential user needs or specify software requirements ambiguously during Requirements Engineering (RE) activities. Software architects may make sub-optimal design decisions with respect to satisfying the requirements when designing the software architecture. Similarly, software engineers and developers may implement software features inadequately (partially or incorrectly) without satisfying all the requirements.

***Requirements Technical Debt (RTD) captures the consequences of such sub-optimal decisions made concerning requirements*.** Unless managed, RTD can cause a software project to deviate from customer expectations or take longer than expected to meet them, causing significant cost overruns.

The goal of this **anonymous online questionnaire** is to understand whether practitioners formally or informally quantify (or measure or estimate) ***RTD***, if so, how they quantify it, and how quantifying RTD could support decision-making for better managing RTD. We want to understand this with respect to software requirements in general and for software requirements concerning veracity. --- Veracity requirements are a specific type of software requirements related to truth, trust, authenticity, provenance, and integrity of data and human interactions in software systems.

Participant involvement

- You have been invited to take part in this study because you are a practitioner involved in requirements engineering (RE), software architecture design, or software implementation activities. You will answer questions based on your experience working in your role.
- Each section contains a pre-text to help you familiarize yourself with the topic the questions of the section are based on. The questions are of type Multiple Choice (MC), 5-point Likert scale, and a few are open-ended. The questionnaire will take approximately 25 minutes to complete.

- The questionnaire is hosted on the Qualtrics online platform for data collection for five months from the date of the ethics approval.

Informed consent and right to withdraw

- **Since the questionnaire is anonymous, submission of the questionnaire will be taken as consent to participate.**
- Participation in this research is entirely voluntary. Questions are optional; you may choose to answer only some of the questions.
- Once you have submitted your responses, we will not be able to remove them from the dataset as they are anonymous.

Minimization of harm

- There is no risk or harm to participants from this study as it is conducted online and anonymously. The contact information of the Research Team is provided below if you would like to ask questions or have any concerns before participating.

Confidentiality and anonymity

- Your personal identification information, such as name, email address, or IP address, will not be collected or recorded. Some demographic data may be collected, but only related to your current role/ company and experience.

Data storage and management

- The responses gathered during the questionnaire will be strictly confidential and not disclosed or shared with external parties. The data will be stored securely in the PhD researcher's university-approved storage for the duration of the study and will be accessible only to the research team.
- After the completion of the research, all the data will be moved securely to the PI's university-approved storage and retained for six years to maintain the integrity of the research.
- Study results may be shared through the PhD researcher's presentations, reports, scholarly articles, and thesis.

Applying for a summary of findings

- A link to a Google form that is entirely separate from the Qualtrics platform is provided at the end of the questionnaire. If you wish to receive a summary of findings or an invitation for a presentation or webinar, you can enter your email address in that form. We will not be able to trace you back to your responses on Qualtrics using the details you provide in the Google form.

Funding and third-party involvement

- This research is funded by the New Zealand Ministry of Business, Innovation, and Employment via the Science for Technological Innovation (SfTI) National Science Challenge Program's Veracity Technology Spearhead research project (https://veracity.wgtn.ac.nz/) hosted by Callaghan Innovation.
- The data will be shared with the Veracity Technology Spearhead main research project for reporting purposes.

Contact the research team.

If you have any questions before participating, please get in touch with us directly – these correspondences will not be associated with the questionnaire responses.

Associate Professor Ewan Tempero (PI/ supervisor): e.tempero@auckland.ac.nz

Miss Judith Perera (Doctoral researcher): jper120@aucklanduni.ac.nz

Associate Professor *Kelly Blincoe (co-investigator/ supervisor): k.blincoe@auckland.ac.nz*

Dr Yu-Cheng Tu (co-investigator/ supervisor): yu-cheng.tu@auckland.ac.nz

Head of Department

Professor Giovanni Russello, Science Centre 303 - Bldg 303, 38 Princes St, Auckland Central, Auckland, 1010, New Zealand.

email: g.russello@auckland.ac.nz

UAHPEC Chair

For any queries regarding ethical concerns, you may contact the Chair, The University of Auckland Human Participants Ethics Committee, Office of Research Strategy and Integrity, The University of Auckland, Private Bag 92019, Auckland 1142. Telephone 09 373-7599 ext. 83711.

Email: humanethics@auckland.ac.nz.

Approved by the University of Auckland Human Participants Ethics Committee on 13/11/2023 for three years. Reference Number **UAHPEC26580.**

I have read the Participant Information Sheet (PIS), take me to the questionnaire.

Demographics I

Section 1: Demographics - Current Role and Experience

For each question in this section, please select the option that best describes your current primary role in your company/ organization.

What is your **current primary role** in your company? *If it is not listed, please select 'Other' and specify.*

- Product Owner (PO)
- Business Analyst/ Requirements Engineer
- Systems Architect/ Software Architect/ Tech Lead
- Software Engineer/ Systems Engineer/ Developer
- Project Manager/ Product Manager/ Team Lead
- QA Manager/ QA Lead/ Quality Assurance Engineer (QAE)
- Researcher in Industry
- Other (Please specify)

Select from below (all) **primary activities you perform** in your current role.

- Capturing user needs, e.g., through stakeholder interviews, user forums, and observations
- Documenting requirements, e.g., in a System Requirements Specification (SRS), as User Stories, as Use Case Diagrams
- Prioritizing requirements, e.g., in the System Requirements Specification (SRS) or as User Stories in a product backlog
- Validating requirements, e.g., through user feedback
- Making architectural design decisions to accommodate user and system requirements
- Writing or modifying software code to implement features that satisfy the requirements
- Implementing feature enhancements
- Fixing bugs
- Making improvements to the code, i.e., Refactoring
- Testing or writing tests, e.g., to validate that requirements have been implemented correctly, to test that the software functions
- Making decisions on behalf of the team, e.g., what requirements to implement, what to refactor
- Other (Please specify)

How many **years of experience** do you have working in your current role?

- Less than an year
- 1-2 years
- 2-5 years
- 5-10 years
- More than 10 years

Quantifying Requirements Technical Debt (RTD)

Section 2: Quantifying Requirements Technical Debt (RTD)

*This section focuses on quantifying RTD for **software requirements in general.** --- By 'quantifying,' we mean measuring or estimating.*

Please read the text below to prepare for this section.

Requirements Technical Debt (RTD) captures the consequences of sub-optimal decisions made concerning software requirements.

RTD can occur, for example, by not adequately capturing essential user needs or ambiguously specifying requirements during RE. Similarly, RTD can occur by inadequately (or partially or incorrectly) implementing requirements as features or design decisions during a software system's architectural design or implementation phases.

The presence of RTD can cause extra efforts or costs in terms of having to rework the software requirements, redesign the software architecture, and rework the software code, for example.

However, sub-optimal decisions can be fixed (or rectified or remediated). If chosen to fix, there can be a cost to fix and a benefit of fixing.

Sub-optimal decisions made concerning requirements *--- here onward we call them* ***"problems with requirements"***

Which of the **following problems** with requirements **have you encountered** within your current role? *Select all that apply.*

- Not capturing essential user needs while gathering requirements
- Not documenting requirements at all
- Insufficient documentation of requirements, i.e., information scattered in different places, requirements are specified only at a high level
- Ambiguities in requirements documentation e.g., in Systems requirements Specification (SRS) or User Stories or Use Cases
- Changes in the requirements documentation

Lack of requirements validation with the user

Trade-offs between what requirements to implement during prioritization of requirements e.g., de-scoping non-functional requirements

Inadequate (incorrect or partial) implementation of requirements

Requirements are not satisfied by the system design, i.e., the end-user satisfaction level is not met by the architectural design

Lack of requirements traceability

Other (Please specify)

Think about **the most recent incident that created a significant impact on your project (or company), due to problems with requirements**. Briefly describe (in one to three sentences) what happened.

Think about the **incident you described**. What was **the most significant problem (or the problems) with requirements** that created the **significant impact on your project (or company)**?

Not capturing essential user needs while gathering requirements

Not documenting requirements at all

Insufficient documentation of requirements, i.e., information scattered in different places, requirements are specified only at a high level

Ambiguities in requirements documentation e.g., in Systems requirements Specification (SRS) or User Stories or Use Cases

Changes in the requirements documentation

Lack of requirements validation with the user

Trade-offs between what requirements to implement during prioritization of requirements e.g., de-scoping non-functional requirements

Inadequate (incorrect or partial) implementation of the requirements

Requirements are not satisfied by the system design, i.e., the end-user satisfaction level is not met by the architectural design

Lack of requirements traceability

Other (Please specify)

When did you fix those problems with requirements?

Before designing or implementing the software system

During designing the software architecture

During implementing software features

During testing the software features

A quick fix just before delivering the software

We fixed them only when the end user complained after delivery

Other (Please Specify)

We DID NOT fix them

When you fixed those problems with requirements, **did you formally or informally quantify the effort or cost to fix**?

By 'quantifying', we mean measuring or estimating.

Yes, formally quantified the effort or cost to fix, e.g., in person-hours/ story points/ dollars or other currency

Yes, informally quantified the effort or cost to fix, e.g., had some expectation of cost in terms of effort spent

NO, did not formally or informally quantify the effort or cost to fix at all

What was the **primary reason for not quantifying the effort or cost to fix?**

There are NO reliable ways to quantify the effort or cost

There are reliable ways, but they are expensive

There are reliable ways, but they are time-consuming

Some decisions are made by experience (expert judgment) rather than quantifying efforts or costs

I am not the decision-maker; higher management roles make decisions. So I did not need to quantify the effort or cost to fix

Other (Please specify)

When you fixed those problems with requirements, did you **formally or informally quantify the benefit of fixing**?

By 'quantifying', we mean measuring or estimating.

Yes, formally quantified the benefit of fixing, e.g., in terms of market share or expected profit or ROI

Yes, informally quantified the benefit of fixing, e.g., had some expectation of benefit in terms of project completion time

NO, did not formally or informally quantify the benefit of fixing at all

What was the **primary reason for not quantifying the benefit of fixing?**

There are NO reliable ways to quantify the benefit

There are reliable ways, but they are expensive

There are reliable ways, but they are time-consuming

Some decisions are made by experience (expert judgment) rather than by quantifying benefits

I am not the decision-maker; higher management roles make decisions. So I did not need to quantify the benefit of fixing

Other (Please specify)

Were there **consequences (or negative impacts)** in terms of extra efforts or costs later in the project due to **NOT fixing or delaying to fix those**

problems with requirements? *Select all that apply.*

- Yes, there were rework design costs, e.g., redesigning the software architecture
- Yes, there were new design costs, e.g., adding new components to the software architecture
- Yes, there were rework code costs, e.g., refactoring software code
- Yes, there were new code costs, e.g., writing new software code
- Yes, there were rework Requirements Engineering (RE) costs, e.g., refining the System Requirements Specification (SRS)
- Yes, there were new Requirements Engineering (RE) costs, e.g., having to conduct new user interviews
- Other (Please specify)

- NO, there were no consequences

Did you **formally or informally quantify such consequences**, i.e., extra efforts or costs incurred in the project, due to **NOT fixing or delaying to fix those problems with requirements**?
By 'quantifying', we mean measuring or estimating.

- Yes, formally quantified the extra efforts or costs, e.g., in person-hours/ story points/ dollars or other currency
- Yes, informally quantified the extra efforts or costs, e.g., had some expectation of extra costs in terms of extra effort spent
- NO, did not formally or informally quantify the extra efforts or costs at all

What was the **primary reason for not quantifying the consequences of NOT fixing or delaying to fix** the problems with requirements?

- There are NO reliable ways to quantify the consequences
- There are reliable ways, but they are expensive
- There are reliable ways, but they are time-consuming
- Some decisions are made by experience (expert judgment) rather than by quantifying the consequences
- I am not the decision-maker; higher management roles make decisions. So I did not need to quantify the consequences of NOT fixing or delaying to fix

Other (Please specify)

Generally, **how often do you fix problems with requirements** in your company?

- Before designing or implementing the software system
- During designing the software architecture
- During implementing software features
- During testing the software features
- Quick fixes just before delivering the software
- Other (Please Specify)
- It's NOT a common practice to fix problems with requirements unless the end user complains

Please **rate how much you agree or disagree with the following statements** regarding the **usefulness of quantifying costs and benefits** for **making the decision to fix problems with requirements**.

Briefly explain (in one or two sentences) the reason for your rating in the optional text boxes.

By 'quantifying', we mean measuring or estimating.

	Strongly agree	Somewhat agree	Neither agree nor disagree	Somewhat disagree
Quantifying the **cost to fix problems with requirements** supports making an **informed decision if and when to fix**	○	○	○	○

	Strongly agree	Somewhat agree	Neither agree nor disagree	Somewhat disagree
Quantifying the **benefit of fixing problems with requirements** supports making an **informed decision if and when to fix**	○	○	○	○

	Strongly agree	Somewhat agree	Neither agree nor disagree	Somewhat disagree
Quantifying the consequences of NOT fixing or delaying to fix problems with requirements supports making an **informed decision if and when to fix**	○	○	○	○

	Strongly agree	Somewhat agree	Neither agree nor disagree	Somewhat disagree

	Strongly agree	Somewhat agree	Neither agree nor disagree	Somewhat disagree
Quantifying the cost to fix problems with requirements is insufficient for making an informed decision; **the benefit also must be quantified**.	○	○	○	○

	Strongly agree	Somewhat agree	Neither agree nor disagree	Somewhat disagree
Quantifying the consequences of not fixing problems with requirements is insufficient for making an informed decision; **the costs and benefits of fixing must also be quantified.**	○	○	○	○

What tools or techniques may help in quantifying costs, benefits and consequences related to RTD?

What other factors apart from costs, benefits and consequences are useful to consider when deciding to fix problems with requirements?

Quantifying RTD for software requirements concerning veracity

Section 3: Quantifying RTD for software requirements concerning veracity

*This section focuses on quantifying RTD for **software requirements concerning veracity**. --- By 'Software requirements concerning veracity,' i.e., veracity*

requirements, we mean software requirements related to ***truth, trust, authenticity, provenance, and integrity of data and human interactions****.*

Please read the text below to prepare.

Veracity requirements can be functional, non-functional, or both, depending on the system under development and its context. In some cases, veracity requirements can be functional requirements that are developed as software features, such as the functionality for user authentication. In some cases, veracity requirements can be non-functional requirements similar to system performance, availability, and security that may be satisfied by the system design and the implementation of the software architecture. One such example is the '*trustworthiness*' of a software system or an Artificial intelligence (AI) system.

Software practitioners could make sub-optimal decisions regarding veracity requirements when gathering such requirements, designing the system, implementing the software architecture, and when implementing software features to satisfy the veracity needs of end-users.

Veracity Requirements Technical Debt (VRTD) captures the consequences of such sub-optimal decisions made concerning veracity requirements.

Our previous work found that there can be **multiple dimensions (types) of veracity**. These dimensions include **regulatory, data, financial data, process, and cultural.**

- Regulatory Veracity requirements: compliance with standards and regulations
- Data Veracity requirements: provenance, transparency, sovereignty, integrity, accuracy, and correctness of data entered into software systems

- Financial Data Veracity requirements: correctness of banking, tax, and accounting data
- Process Veracity requirements: adherence to organizational policies, processes, methods, and practices
- Cultural Veracity requirements: requirements related to, e.g., trust within society, culture, brand

We describe an illustrative example of veracity requirements on the next screen.

Below is an **illustrative example of Veracity requirements from the New Zealand organic certification domain**. Veracity requirements can apply to other domains as well.

The Organic certification domain in New Zealand is a domain where veracity is a primary concern for the certification of organic produce such as vegetables, dairy products, honey, grapes, and wine, as well as their production and supply chain processes.

During engagements with the stakeholders in the domain, we found that there can be multiple dimensions of veracity. These dimensions include *regulatory, data, financial data, process,* and *cultural*.

When developing software systems for the auditing and certification of organic produce and processes, there can be functional and non-functional software requirements pertaining to these different dimensions of veracity.

- ***Regulatory Veracity requirements*** refer to compliance with regulations, such as organic product regulations, standards, and other regional, local, or international laws.
- ***Data Veracity requirements*** refer to the veracity-related properties of the data entered into the software system, such as provenance, transparency, sovereignty, integrity, and accuracy or correctness of the data. An example from our above case study is that it should be possible to verify that the results of a particular soil test come from testing the soil of the specific farm or vineyard that uploads the test results to the organic certification agencies' software system. This requires implementing the functionality for verifying test results and the provenance of the data in the software system.
- ***Financial Data Veracity requirements*** refer to the correctness of banking, tax, sales, and accounting data. For example, records of inventory management must match the inventory. The ability to verify this is another required functionality for the certification software.
- ***Process Veracity requirements*** refer to the adherence to internal organizational policies, processes, methods, and practices. An example from our case study is the practice of having prior approval from the organic certification agency to make specific changes to the Organic Management Plan (OMP). The system should have the functionality to verify that this approval has been obtained.
- ***Cultural Veracity requirements*** refer to the trust within the society and culture (human interactions) and trust for a brand, for example, cultural aspects pertaining to the trust of indigenous communities in New Zealand.

Sub-optimal decisions concerning veracity requirements (i.e., problems with veracity requirements), can lead to inadequate satisfaction of the veracity requirements pertaining to these different dimensions of veracity. This could, in turn, lead to building the wrong product in terms of the desired level of veracity for the end-users.

Consider the application domain and the technologies you work on. **What types of veracity requirements are usually supported by the software solutions** developed by your organization? *Select all that apply.*

- Regulatory Veracity requirements: compliance with standards and regulations
- Data Veracity requirements: provenance, transparency, sovereignty, integrity, accuracy, and correctness of data entered into software systems
- Financial Data Veracity requirements: correctness of banking, tax, and accounting data
- Process Veracity requirements: adherence to organizational policies, processes, methods, and practices
- Cultural Veracity requirements: requirements related to, e.g., trust within society, culture, brand
- Other (Please specify)

Recall **the incident related to problems with software requirements that you described** in the previous section of the questionnaire. Was it related to veracity requirements?

- Yes, that was precisely veracity requirements
- Yes, that included veracity requirements
- NO, the problem was not related to veracity requirements at all

What types of veracity requirements were the **most relevant in the incident you described** in the previous section of the questionnaire? *Select all that apply.*

- Regulatory Veracity requirements: compliance with standards and regulations
- Data Veracity requirements: provenance, transparency, sovereignty, integrity, accuracy, and correctness of data entered into software systems
- Financial Data Veracity requirements: correctness of banking, tax, and accounting data
- Process Veracity requirements: adherence to organizational policies, processes, methods, and practices

Cultural Veracity requirements: requirements related to, e.g., trust within society, culture, brand

Other (Please specify)

Think about the **most recent incident that created a significant impact on your project (or company) due to problems with veracity requirements.** Briefly describe (in one to three sentences) what happened.

What types of veracity requirements were **most relevant** in **the incident you described?** *Select all that apply.*

Regulatory Veracity requirements: compliance with standards and regulations

Data Veracity requirements: provenance, transparency, sovereignty, integrity, accuracy, and correctness of data entered into software systems

Financial Data Veracity requirements: correctness of banking, tax, and accounting data

Process Veracity requirements: adherence to organizational policies, processes, methods, and practices

Cultural Veracity requirements: requirements related to, e.g., trust within society, culture, brand

Other (Please specify)

Please rate **how much you agree or disagree with the following statements** regarding **the usefulness of quantifying costs, benefits, and consequences for making the decision to fix problems with veracity requirements**.

Briefly explain (in one or two sentences) the reason for your rating in the optional text boxes.

By 'quantifying,' we mean measuring or estimating.

	Strongly agree	Somewhat agree	Neither agree nor disagree	Somewh disagree
Quantifying the **cost to fix problems with veracity requirements** supports making an **informed decision if and when to fix.**	○	○	○	○

	Strongly agree	Somewhat agree	Neither agree nor disagree	Somewh disagree
Quantifying the **benefit of fixing problems with veracity requirements** supports making an **informed decision if and when to fix.**	○	○	○	○

	Strongly agree	Somewhat agree	Neither agree nor disagree	Somewh disagree
Quantifying the **consequences of NOT fixing or delaying to fix problems with veracity requirements** supports making an **informed decision if and when to fix.**	○	○	○	○

	Strongly agree	Somewhat agree	Neither agree nor disagree	Somewhat disagree
Quantifying the cost to fix problems with veracity requirements is insufficient for making an informed decision; **the benefit also must be quantified**.	○	○	○	○

	Strongly agree	Somewhat agree	Neither agree nor disagree	Somewhat disagree
Quantifying the consequences of not fixing problems with veracity requirements is insufficient for making an informed decision; **the costs and benefits of fixing must also be quantified.**	○	○	○	○

In your opinion, **for what type of veracity requirements** is it **most useful to quantify the cost to fix** a problem**?**

Give reasons in the optional text box that follows.

Regulatory

Data

Financial Data

Process

Cultural

Other (Please specify the type and why)

In your opinion, **for what type of veracity requirements** is it **most useful to quantify the benefit of fixing** a problem?

Give reasons in the optional text box that follows.

Regulatory

Data

Financial Data

Process

Cultural

Other (Please specify the type and why)

In your opinion, **for what type of veracity requirements** is it **most useful to quantify the consequences of NOT fixing or delaying to fix a problem**?

Give reasons in the optional text box that follows.

Regulatory

Data

Financial Data

Process

Cultural

Other (Please specify the type and why)

Please rate **how much you agree or disagree with the following statements** regarding **the consequences of NOT fixing** or delaying to fix **problems with veracity requirements compared to other software requirements**.

Briefly explain (in one or two sentences) the reason for your rating in the optional text boxes.

By 'quantifying', we mean measuring or estimating.

Impact on the software architecture

	Strongly agree	Somewhat agree	Neither agree nor disagree	Somewh disagree
The consequences of not fixing a problem with veracity requirements can have a much higher impact on the software architecture, for example, causing rework on the software architecture or the data infrastructure.	○	○	○	○

Impact on the end-user

	Strongly agree	Somewhat agree	Neither agree nor disagree	Somewh disagree
The consequences of not fixing a problem with veracity requirements can have a much higher impact on the end-user when the desired level of veracity is not met.	○	○	○	○

Impact on the software company

	Strongly agree	Somewhat agree	Neither agree nor disagree	Somewh disagree
Consequences of not fixing a problem with veracity requirements can have a much higher impact on the software company, e.g., in terms of potential ROI, product success, and company reputation.	○	○	○	○

What tools or techniques may help in quantifying costs, benefits and consequences related to **Veracity RTD**?

What other factors apart from costs, benefits and consequences are useful to consider when deciding **to fix a problem with veracity requirements compared to other software requirements**?

Demographics II

Section 4: Demographics - Current Company/ Organization

Please select the option that best describes your current company/

organization for each question in this section.

In which **country** is your **current company/ organization** located?

How would you describe the **scale of your current organization**? SMEs are firms with 6 to 49 employees, and large businesses are firms with 50+ employees.

What is (are) the **primary application domain (s) of the products and services** you work on in your current company/ organization?

- Agriculture
- Viticulture/ Organic Viticulture
- Growing/ Farming other Organic products
- Automotive
- Avionics
- Compliance, Audits and Certification
- Cultural organizations
- Distribution / Transportation/ Logistics
- e-Commerce
- Education
- Energy
- Enterprise Resource Planning (ERP)
- Export Industry

Finance

Government Regulations / Policies

Healthcare

Human Resources

Insurance

Manufacturing

Winemaking/ Organic Winemaking

Manufacturing other Organic products

Railway

Security

Telecommunication

Other (Please specify)

What is the **software development methodology** practiced in your current company/ organization?

Traditional software development, i.e., Waterfall

Agile software development, i.e., Continuous Software Engineering

A hybrid approach of both traditional and agile approaches

Other (Please specify)

Closing Question

Please comment on **any additional thoughts** you have regarding our online questionnaire.

Request Summary of Results

If you wish to receive a summary of the study results, please click **HERE** to go to a **separate Google form** where you can enter your email address.

*There is **no linkage between the details you enter in the Google form and your responses to the questionnaire**. We will not be able to trace you back.*