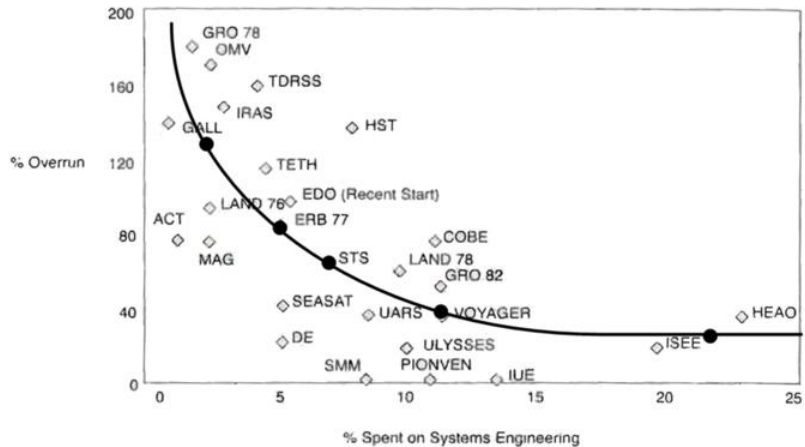


Using Natural Language Parsing (NLP) for Automated Requirements Quality Analysis

Chris Ritter, F. David Ayhan

Why are Quality Requirements Important?

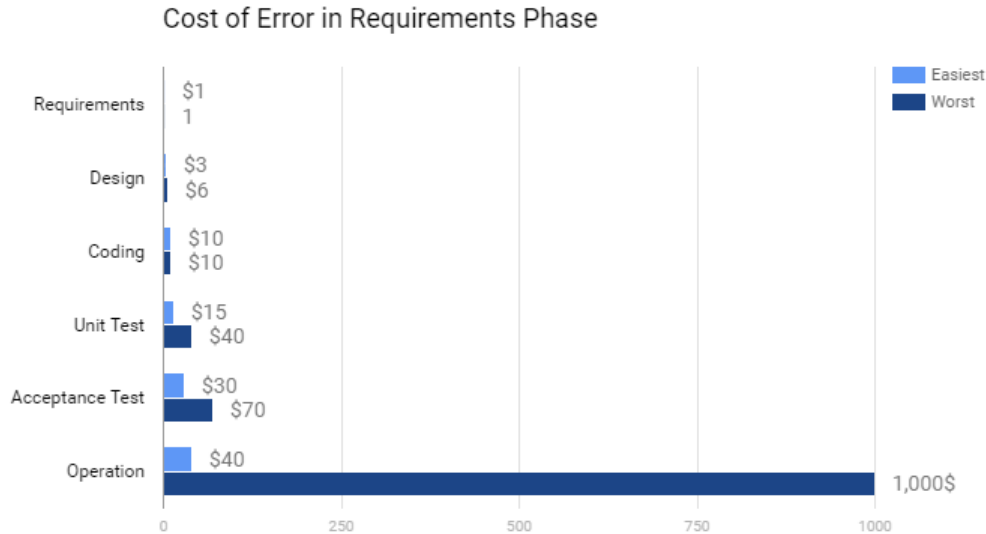
Cost Overrun vs Systems Engineering Effort



*Source: Werner M. Gruhl, Chief Cost & Economics Analysis Branch, NASA Headquarters

NASA found a smaller investments in systems engineering proportional to cost overrun

Cost of Error in Requirements Phase



("Extra Time Saves Money", Warren Kuffel)

Expect a **10 to 1000x increase** in cost for errors missed in the requirements/ design phase

Where are my Requirements?

Why not Microsoft Office?

- **Many Versions** - Often documents are emailed around and managing consistency is difficult
- **Change Tracking** - Change logs can be kept manually but often finding the rationale for a requirement change is impossible
- **Traceability** - Maintaining traceability between requirements and artifacts becomes a costly and manual process

Why start with models?

- Problem:
 - Natural language requirements can be difficult for humans to read and are often ambiguous
- Current Solution:
 - Models can be executed in a **simulation** environment for verification and validation
 - Models can generate **consistent** documentation
 - Models can **automatically generate** necessary views (requirements document, hierarchy chart, requirements diagram etc.)
- New Problem
 - Models almost always become **documents**
 - How do we ensure quality of natural language documents?

What Makes a Good Requirement?

Attributes of Quality Requirements

- **Correct** – i.e., describes the user's intent and legally possible
- **Complete** – i.e., expresses a whole, single idea, and not portions of one or many
- **Clear** – i.e., explicit and not confusing to readers
- **Consistent** – i.e., does not conflict with other requirements
- **Verifiable** – i.e., proves within realistic cost and schedule that the architecture meets the requirement

Attributes of Quality Requirements

- **Traceable** – i.e., uniquely identify, and able to be tracked to predecessor and successor lifecycle items/objects, such as functions or components
- **Feasible** – i.e., implement with existing or projected technology and within cost and schedule
- **Modular** – i.e., changes without excessive impact on other requirements
- **Design** – i.e., does not impose a specific solution (“what” not “how”)

Requirement Pitfalls

- DON'T use:
 - ambiguous language
 - use bullet lists; use numbered lists instead
 - jargon
 - language that provides an escape clause Ex: “The user shall be able to access the Internet as often as is practicable”
 - write long, rambling sentences
 - two requirements in one sentence; e.g., “The system shall ... and ...”
 - vague terms -- Ex: “user-friendly”
 - include suggestions or possibilities – Ex: “may”, “should”, “ought”
 - wishful thinking – Ex: “The system shall be 100% reliable”

Natural Language Processing

- Definition: “a field of computer science, artificial intelligence, and linguistics concerned with the interactions between computers and human (natural) languages”
- Innoslate uses this technology to break down sentences into nouns, verbs, and adjectives to identify when conjunctions are used (clear), specific types of hardware/software specified (design), and other parameters that affect the requirement’s quality

Part of Speech Tagging

Given:

The system shall identify the part based on either a Serial Number or a Part Name.

Tagged as:

The **(Determiner)** system **(Noun)** shall **(Modal)** identify **(Verb)**
the part based on either a Serial Number **(Noun)** or a Part
Name **(Noun)**.

Clear - Multiple modals; conjunctions

Given:

The system shall start within 5 seconds **but** can take 10 seconds.

The screenshot displays a software interface for managing requirements. The main window shows a requirement form with the following fields:

- Number:** 1
- Description:** The system shall start within 5 seconds but can take 10 seconds.
- Quality Score:** 22% (indicated by a red progress bar)
- Rationale:** (empty)
- Clear:** A dropdown menu currently set to "No". A tooltip is visible over this field, stating: "Clear represents if this Requirement is unambiguous and not confusing." and listing two criteria:
 - Use of more than one modal verb may lead to ambiguity or confusion.
 - Contains a conjunction; consider separating into two statements.
- Complete:** Yes
- Consistent:** No

On the right side of the interface, there is a list of relationships with "Add" buttons:

- decomposed by Children
- decomposes Parents
- references Artifact
- sourced by Artifact 1
- rer (with a red 'X' button)
- traced from Statement
- traced to Many

At the bottom right, the SPEC INNOVATIONS logo is visible.

Complete - Sentence Detection

Given:

The system shall start
within 5 seconds

The screenshot displays a web-based interface for managing requirements. The main content area shows a requirement with the following details:

- Number:** 1
- Description:** The system shall start within 5 seconds
- Quality Score:** 33% (indicated by a red progress bar)
- Rationale:** (Empty text area)
- Clear:** Yes
- Complete:** No (dropdown menu is open, showing a tooltip: "Complete represents if this Requirement expresses a whole idea." and a bullet point: "Requirement does not end with a period.")
- Consistent:** No
- Correct:** No
- Design:** Yes

On the right side, there is a list of relationships with "Add" buttons:

- decomposed by Children
- decomposes Parents
- references Artifact
- sourced by Artifact 1
- rer (with a red "X" button)
- traced from Statement
- traced to Many

At the bottom right, the logo for SPEC INNOVATIONS is visible.

Complete - Run-On Detection

Given:

The system shall identify the part based on **either a Serial Number Part Name** that are also acceptable input.

The screenshot displays a web-based requirement management tool. The top navigation bar includes 'MENU', 'Dashboard', 'Database', 'Requirements', and 'Diagrams'. The user is logged in as 'John Doe'. The main content area is titled 'Requirement' and shows the following details:

- Id:** e1721
- Class:** Statement/ Requirement
- Modified:** 10/10/2016 by John Doe
- Created:** 10/10/2016 by John Doe

The 'Attributes' section contains the following fields:

- Name:** (empty text box)
- Number:** 1
- Description:** The system shall identify the part based on either a Serial Number or a Part Name.
- Quality Score:** 33% (with a progress bar)
- Rationale:** (empty text box)
- Clear:** No (with a tooltip: 'Contains a conjunction; consider separating into two statements.') (dropdown menu)
- Complete:** Yes (dropdown menu)

The 'Relationships' section lists several relationships with 'Add' buttons:

- decomposed by Children
- decomposes Parents
- references Artifact
- sourced by Artifact 1
- traced from Statement
- traced to Many

A 'Clear' button is also present with a tooltip: 'Clear represents if this Requirement is unambiguous and not confusing.'

Design - Predetermined Solution

Given:

The system shall use
**Microsoft SQL
Server 2008R2.**

The screenshot displays a software requirement management interface. The top navigation bar includes 'MENU', 'Dashboard', 'Database', 'Requirements', and 'Diagrams'. The main content area shows a requirement with the following details:

- Change:** Requirement
- Id:** e1721
- Class:** Statement/ Requirement
- Modified:** 10/10/2016 by John Doe
- Created:** 10/10/2016 by John Doe

The requirement is categorized under 'Labels' with the following options:

- DoDAF Product
- Environmental Requirement
- Functional Requirement
- Interface Requirement
- Performance Requirement
- Reliability Requirement
- Safety Requirement

The requirement's properties are as follows:

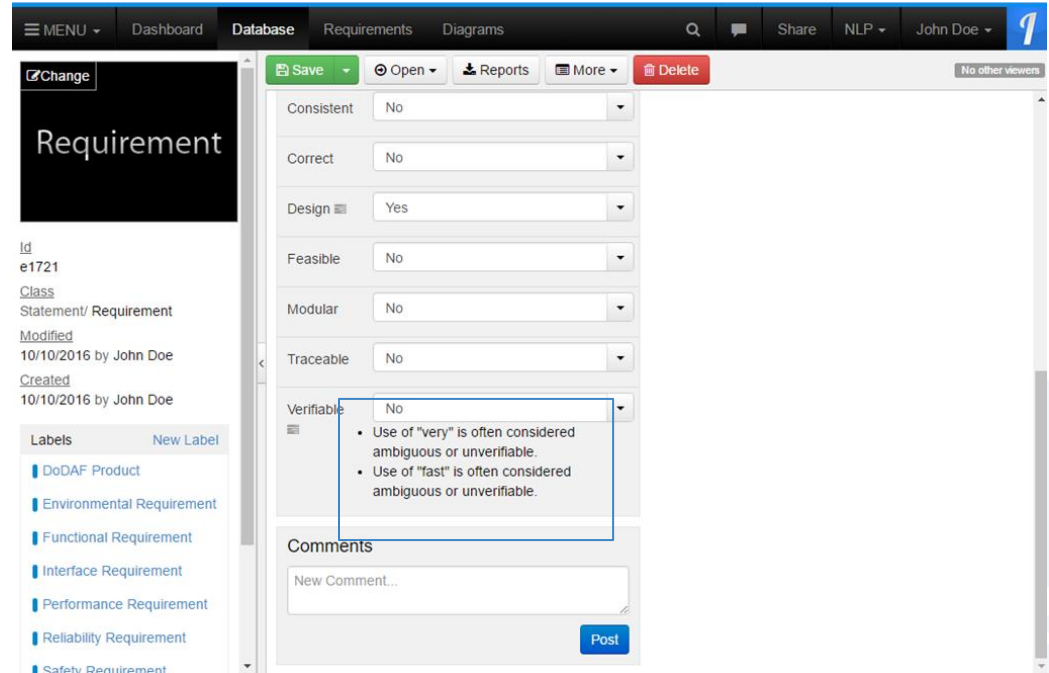
- Complete: Yes
- Consistent: No
- Correct: No
- Design: No (highlighted with a blue box containing the text: 'Using Microsoft is not a necessary requirement.' and 'Using SQL is not a necessary requirement.') (A tooltip for 'Design' explains: 'Design represents if this Requirement does not impose a specific solution on design; says "what", not "how".')
- Feasible: No
- Modular: No
- Traceable: No
- Verifiable: No (A tooltip for 'Verifiable' explains: 'Use of "use" is often considered ambiguous or unverifiable.')

The 'Comments' section is currently empty.

Verifiable - Ambiguous Words

Given:

The system shall be
very fast.



The screenshot shows a web-based interface for managing requirements. The top navigation bar includes 'MENU', 'Dashboard', 'Database', 'Requirements', and 'Diagrams'. The main content area is titled 'Requirement' and displays the following details:

- Id:** e1721
- Class:** Statement/ Requirement
- Modified:** 10/10/2016 by John Doe
- Created:** 10/10/2016 by John Doe

Below the details is a 'Labels' section with a 'New Label' button and a list of labels: DoDAF Product, Environmental Requirement, Functional Requirement, Interface Requirement, Performance Requirement, Reliability Requirement, and Safety Requirement.

The main form contains several dropdown menus for property selection:

- Consistent: No
- Correct: No
- Design: Yes
- Feasible: No
- Modular: No
- Traceable: No
- Verifiable: No

A tooltip is displayed over the 'Verifiable' dropdown, containing the following text:

- Use of "very" is often considered ambiguous or unverifiable.
- Use of "fast" is often considered ambiguous or unverifiable.

At the bottom of the form is a 'Comments' section with a text input field labeled 'New Comment...' and a 'Post' button.

Coming Soon

- Fragment Detection
- Active Voice Detection
- Multiple Negative Detection
- Exact Duplicate Identification
- Near Duplicate Identification

Next Generation

- Utilize machine learning to develop self-improving algorithms
 - Gather large data sets of requirements
 - Process requirements with neural networks
 - Analyze network to identify good patterns and common pitfalls

Contact Information

- Christopher Ritter
- 301-910-1818
- SPEC Innovations
- chris.ritter@specinnovations.com