REQUIREMENTS ENGINEERING RESEARCH IN A CHANGING WORLD

International Requirements Engineering Conference
Keynote: August 22\textsuperscript{nd}, 2018
Banff, Canada

Jane Cleland-Huang, PhD
University of Notre Dame
Dept. of Computer Science and Engineering
http://sarec.nd.edu
JaneClelandHuang@nd.edu

Work described in this talk is funded by the US National Science Foundation under Grants CCF-0959924, CCF-1265178, and CCF-1741781
What I’m going to talk about today

- Back to the future
- Disruptive change
- Our Response

Confessions of a Traceability Researcher!
Back to the future
Requirements engineering is the branch of software engineering concerned with the real-world goals for, functions of, and constraints on software systems. It is also concerned with the relationship of these factors to precise specifications of software behavior, and to their evolution over time and across software families. Zave – 1982.

RE is often regarded as a front-end activity in the software systems development process. Nuseibeh and Easterbrook – Roadmap

**Historical Focus**

- Eliciting Requirements
- Modeling and Analyzing Requirements
- Communicating Requirements
- Agreeing Requirements
- Evolving Requirements
- Integrated Requirements Engineering

**Radical Changes of the previous period**

- Modeling and analysis within social context
- Shift from modeling information flow and state modeling goals and scenarios.
- Acknowledging that RE must accept inconsistencies, uncertainties, and conflicts.

**Roadmap:**

New techniques for formally modeling environment, bridging the gap between elicitation based on contextual enquiry and formal modeling, richer models for capturing NFRs & a focus on architectural impact, reuse of requirements models, and multidisciplinary training for practitioners.

**ABSTRACT**

This paper presents an overview of the field of software requirements engineering (SRE) and describes the main areas of RE research, and highlights some key issues and open questions.

**RE** is often regarded as a **front-end activity** in the software systems development process.
Research Directions in Requirements Engineering (2007)

Requirements engineering is about defining precisely the problem that the software is to solve. … RE Activities may be more iterative, involve many more players…, requirement more extensive analyses of options, and call for more complicated verification of more diverse components.

State of the Art

- Elicitation
- Modeling
- Requirements Analysis
- Validation and Verification
- Requirements Management

Research Strategies

- Paradigm Shift
- Leverage other disciplines
- Leverage technology
- Evolutionary research
- Domain-specific
- Generalization
- Evaluation

Hot Spots

- Paradigm Shift
- Leverage other disciplines
- Scale; security; tolerance; environmental dependency; self-management; globalization;
  methodologies, patterns, and tools; requirements management; Technologies.
The year 2002...

Requirements Engineering Conference, 2002, Essen. Who believes that Agile is more than a passing trend?
We are bombarded with change on every side...

Elicitation vs. invention

Requirements discovery by means of feature mining and app reviews.

User Stories, Sprints, Scrum!

The rise of CPS...

Where requirements are feature requests and bug reports!
Safety Critical projects seek increased agility..
Is Requirements Engineering as we know it a relic of the past?
Or is this a Goldilocks Moment?

“Who’s been sleeping in my bed?” said Papa Bear.
“Who’s been sleeping in my bed?” said Mama Bear.
“Look, there’s someone in my bed!” said Little Bear. “And there she is!”
The Goldilocks Principle

Conditions need to be ‘just right’ for transformative change.
Innovation
Paradigm shifts
Leverage emergent technologies
Break throughs
Traceability in a nutshell..

The ability to **interrelate any uniquely identifiable** software engineering artifact to any other, **maintain** required links over time, and **use the resulting network** to answer questions of both the software product and its development process.

- **CoEST Definition**

Required by many regulatory bodies and standards..

*But hard to achieve in practice.*
Accurate & Complete Traceability is challenging

Based on over a decade of traceability engagements in industrial projects we have observed a \textit{traceability gap} between what is prescribed and what is delivered:

Our study of Medical Device submissions to the FDA showed \textbf{incomplete and sometimes entirely missing}, inaccurate, redundant trace links – delivered as a big bang!

A \textbf{formal comparison of five safety-critical software systems} which claimed to follow various standards and guidelines showed \textbf{similar traceability problems}. 
Margaret Hamilton’s Keynote @ ICSE

When we ask developers who are looking for a better way to develop software and we ask them what their most pressing issues are:

- Integration too late if at all
- Lack of traceability, flexibility and evolvability
- Reuse methods ad hoc and error prone
- Software unreliable even with extensive testing
- Costs too much. Takes too long

Why still? Not unlike 50 years ago when the field was brand new*. What to do...


The programmer who wrote much of the code that took Apollo to the moon!

HARD QUESTIONS
Question #1: Am I addressing an important problem?

About 80% accuracy.
5 things industry told me when I listened

- Traceability is one of the most pressing problems we face.
- Traceability is a made-up problem!
- You are solving the ‘wrong’ part of the problem.
- Your solutions don’t scale.
- You are the expert. Give us ready-to-use tools, now!
Others have listened too..

Lessons learned
- Lack of time
- Lack of experience of RE team members
- Weak qualification of RE team members
- Communication flaws between project team and the customer
- Requirements remain too abstract
- Changing business needs
- Customer does not know what he wants
- Missing direct communication to customer
- Language barriers
- Strict time schedule by customer

Claims:
- Provides insights into industrial RE problem trends
- Helps to lay the foundation to steer academic and industrial research in a problem-driven manner where scientific contributions to RE can be put in tune with practically relevant problems.
Seminal work in 2001 launched a new research direction – the quest to automate the traceability process – followed shortly thereafter by work at RE by Jane Hayes and Alex Dehktyar.

Results reported in TSE in 2002.
Hazard H2: Moving the patient’s arm at excessive velocity.

Fault F2.1: Velocity sensors fail to sense excessive velocity.

R1: A system test must be run prior to each use to check that the sensors are operating correctly.

R2: All sensors must be duplicated.

R3: The robotic arm must stop automatically if arm sensors disagree on current velocity by more than x mps.

Fault F2.2: Configuration component fails to update correct velocity constraints.

R9: Current velocity constraint is displayed on the monitor.

R10: Movement must be disabled if current velocity constraint fails to match patient’s personal record.

R11: Current velocity constraint must fall under maximum allowed velocity.
Roadmaps help set directions
Open Research Questions

Planning and Managing

Planning and managing is at the heart of all other areas of the traceability life-cycle.

What tasks do people need traceability to support?

What is the role of traceability in each of those tasks?

Knowledge Reuse

RD-2.1 Identify ingredients for through-life traceability success in different contexts, from a thorough understanding of industry best and worst practice, and then use this knowledge to establish a process and standard. (Purpose)

RD-3.3 Self-Adapting Solutions

Self-aware systems are able to modify their own behavior in an attempt to optimize performance. Such systems can self-diagnose, self-repair, adapt, add, or remove software components dynamically.

Maintaining Trace Links

While traceability is touted for its ability to support change, trace maintenance actually adds work and can impede change. Furthermore, trace links are brittle and break easily.

RD-4.1 Understand patterns of change across time.

Creating and Using Trace Queries

RD-6.1 Integrate traceability into existing development tools

RD-6.2 Provide intuitive forms of query mechanism including visual languages and natural language interfaces.

Visualizing and Understanding Results

Enormous advances have been made in popular techniques and tools for information and knowledge visualization.

Visual analytics are now a common form of support for decision-making activities in many fields of endeavor.

Despite some pockets of research, our field has been slow to keep pace, and must re-examine its information-seeking needs and mechanisms.

RD-7.1 Construct a taxonomy of available visualizations and fundamental traceability tasks.

Bridge these by exploring the basic visualization principles that they either provide or demand.

RD-7.2 Gather and share user-based empirical data to evaluate trace visualizations.

RD-7.3 Perform in-situ user studies to evaluate and understand task-specific needs for trace information.
The Grand Challenge of Traceability

**Inherent.** Traceability is always there, without having to think about getting it there. Traceability is neither consciously established nor sought; it is built-in and effortless. It has effectively ‘disappeared without a trace.’

How do we measure that?

Total automation of trace creation and trace maintenance, with quality and performance levels superior to manual efforts.
Automated Trace Link Creation and Maintenance

Requirement
Stop movie at request or at the end (no auto restart)
Start playing movie within 1 sec after selection

Behavior
select
play
stop

Structure
Display
Streamer
select()
play()
stop()
stream()
wait()

Code

Figure courtesy of Alex Egyed, University of Vienna.
Establish Measurable goals

We need to know where we are going!

Define meaningful measures!

• RD-3.1 Develop intelligent tracing solutions which are not constrained by the terms in source and target artifacts, but which **understand domain-specific concepts**, and can reason intelligently about relationships between artifacts.

• RD-3.1 Deliver **prospective trace capture solutions** that are capable of monitoring development environments, including artifacts and human activities, to infer trace links.

• RD-3.3 Adopt **self adapting solutions** which are aware of the current project state and **reconfigure** accordingly in order to optimize the quality of trace links.
Question #3: Is there a trajectory to a real solution?
Our response

Yes – but we are probably not on it!

Semantic Solutions

Evolving and Managing Links

Traceability for Safety Critical...
(1) Solution 1: Semantic Traceability

**Hypothesis:** Real advancements, that make a difference to the traceability problem, will only be achieved as we transition towards more intelligent traceability solutions.
Why a semantic approach?

Status of field elements is consumed by the **WIU**, which in turn creates a **wayside status message** and **broadcasts** that **message** out to any **automobile** within range.

The **Highway Wayside Segment** shall **transmit** information to the **automobile controller** in the form of **WSMs**.

**WIU** = Wayside Interface Unit

**Broadcast** is similar to transmit

**Automobile controller** is part of the **automobile**

**WSM and wayside status message** are equivalent.

Both artifacts involve **Highway Wayside Segment** **sending** wayside status message to automobiles.
Semantically enhanced Software Traceability using Deep Learning techniques.

Jin Guo, Jinghui Cheng, Jane Cleland-Huang: ICSE 2017: 3-14
Automated approaches that generate trace links from scratch, return imprecise results.

They are useful for supporting tasks such as Impact analysis, but not currently sufficiently reliable on their own.
Solution 2: Evolving and Discovering Links

1. Software artifacts changed across versions

2. Tools for detecting changes in code and requirements.

3. TLE tools and algorithms for recognizing change patterns and evolving trace links.

- Requirements $R_i$
- Source Code $C_i$
- Trace Links $T_i$

- Requirements $R_{i+1}$
- Source Code $C_{i+1}$
- Trace Links $T_{i+1}$ Evolved

- srcML
- Java-Call Graph
  - Source Code Processing

- Refactoring Crawler

- Similarity Computations using IR Methods

- Change Scenario Detection Heuristics
  - Change Scenarios S1-S18 detected

- Trace Link Heuristics Applied against Scenarios S1-S24 and Trace Links $T_{i+1}$ Generated
Evolving Links

1. Identify types of changes that could invalidate existing trace links.

2. Define properties to detect when the change has occurred.

3. Define trace link generation heuristics
In our experiments the effort required by humans to confirm or deny TLE links was minimal – with few decision points per day.

Leverage the Project Environment

Traceability in the wild: automatically augmenting incomplete trace links.
Michael Rath, Jacob Rendall, Jin L. C. Guo, Jane Cleland-Huang, Patrick Mäder: ICSE 2018: 834-845
New Trajectories bring new challenges

Semantic Solutions
Addresses real problem ✓
Leverages Goldilocks ✓
Well defined benchmarks ✓
Adequate data sets X

Evolving Links
Addresses real problem ✓
Leverages Goldilocks ✓
Well defined benchmarks ✓
Adequate data sets X

Mine Links
Addresses real problem ✓
Leverages Goldilocks ✓
Well defined benchmarks ✓
Adequate data sets ✓

Safety Critical Trace..
Addresses real problem ✓
Leverages Goldilocks ✓
Well defined benchmarks ✓
Adequate data sets X

Obvious solution is to build industry collaborations...
Why real projects?

- Strengthens the believability of Dronology as an ‘industrial strength’ project – helping us to achieve our original goal of developing a ‘research incubator’.

- Opens up *new* and interesting research questions. Advances state of the art in small UAV applications.

- Addresses a humanitarian need.

Safety Critical, Cyber Physical System.
Dronology Project @ Notre Dame

A platform for coordinating the flight of UAVs. Supports research in safety assurance, runtime monitoring, & adaptation.

*Developed by the Notre Dame Team: Michael Vierhauser, Jane Wyngaard, Jinghui Cheng, Sean Bayley, Greg Madey, Joshua Huseman, Jane Cleland-Huang, & more...*
Dronology Stakeholders

Requirements Engineering researchers interested in using Dronology to support research into traceability, forensic requirements, goal modeling, runtime adaptation....

End users interested in deploying Dronology to support specific scenarios of use.

Stakeholders’ needs ultimately drive feature prioritization and variability points.

Students
General public
Funders

http://Dronology.info
River Rescue Demo with Dronology
Testing an AED drop
Testing an AED drop
Dronology: Our Crashes

- Bent prop drone
- Trajectory challenged, upside down drone
- Broken leg drone
- Rescued Drone
- Drowned and missing drone
What do we gain?

An entirely new controllable environment for experimenting with Software Traceability across multiple artifact types and multiple versions.

A system that focuses on an agile, yet safety-conscious, more requirements-centric process.
An Ecosystem of Traceability Artifacts

Artifact tree generated automatically from Jira and Github showing traceability from hazard to code.
Discovering, Analyzing, and Managing Safety Stories in Agile Projects

Jane Cirelli-Huang and Michael Vierheller
Department of Computer Science and Engineering
University of Notre Dame
Notre Dame, IN

JaneCirelliHuang.michaelv@nd.edu

Abstract—Traditionally, safety-critical projects have been
monitored using a waterfall process. However, this makes
sense for new and challenging to implement changes are
fast-moving, and to swiftly implement any issues. In
this paper, we present the specific problems of discovering,
monitoring, tracking, and managing safety requirements
throughout the Agile process. We propose a Visual
Requirements and Risk Analysis (Vera) framework for
enabling an agile development process to manage
safety-critical requirements. We apply our framework to
a recent NASA NASA Mission Planning scenario.

1. Introduction

Systems operating in safety-critical domains, where
failure can cause harm or death, must not only deliver
required functionality, but must do so in a way that ensures
the safety of the users. Safety-critical systems must operate
without failures that could lead to harm. To ensure
time-critical systems must meet stringent performance
to meet the system's requirements in a timely fashion.

The strict requirements of the certification process as
well as constant evolution by the regulatory authorities
impose a significant challenge on the development of
such systems. Therefore, the need for an agile approach
in developing safety-critical systems has gained
increasing attention in recent years. The introduction
of agile methodologies has led to a more efficient
and effective process for developing safety-critical
systems. This paper presents a framework for
managing safety-critical requirements in an agile
development environment. We describe the
framework and its application to a recent NASA
Mission Planning scenario.

1.1. Immersive Engineering

Immersive engineering involves providing a realistic
environment to the user that simulates the real
world. This technique is used in various industries,
including aerospace, healthcare, and manufacturing.
Immersive engineering is particularly useful in
safety-critical domains, where the risk of error can
have severe consequences. It allows developers to
experience the system in a more realistic and
engaging way, facilitating better decision-making
and problem-solving.

1.2. Safety Stories

Safety stories are a method used in agile development
projects to document and manage safety-related
requirements. They provide a structured way to
record and track safety-critical issues throughout
the development process. Safety stories allow
stakeholders to easily understand and address
safety concerns, ensuring that the final product
meets the required safety standards.

The framework presented in this paper enables
the systematic discovery, analysis, and management
of safety stories in an agile development environment.
It facilitates the tracking of safety-critical issues,
ensuring that they are addressed promptly and
adequately. The Vera framework provides a
structured approach to managing safety-critical
requirements in an agile context, aligning with the
principles of agility and continuous improvement.

2. Related Work

Several frameworks and methods have been developed
for managing safety-critical requirements in
agile environments. One notable approach is the
Safety in Agile Concepts (SiAC) framework, which
provides a method for integrating safety into the
Agile process. SiAC emphasizes the importance
of risk management and continuous assessment
to ensure that safety-critical requirements are
correctly addressed throughout the development
process.

Another approach is the Safety in Agile Concepts
(SiAC) framework, which integrates safety into the
Agile process through the use of safety stories.
These stories capture safety-critical issues and
are continuously reviewed and updated as the
project progresses. This approach ensures that
safety is considered throughout the development
process, maintaining a focus on safety-critical
requirements.

3. Methodology

The methodology used in this study involved a case
study approach, focusing on a specific NASA
Mission Planning scenario. The Vera framework
was applied to this scenario to demonstrate its
effectiveness in managing safety-critical
requirements in an agile environment.

The case study was conducted in collaboration
with NASA engineers, who provided insights
and feedback on the application of the Vera
framework. This collaboration facilitated a
better understanding of the practical challenges
and potential benefits of using the framework
in a real-world context.

4. Results

The application of the Vera framework to the NASA
Mission Planning scenario revealed several
benefits. The framework facilitated the discovery
and analysis of safety-critical issues, enabling
more effective decision-making and resource
allocation. Additionally, the continuous tracking
and monitoring of safety stories ensured that
risks were identified and addressed promptly.

The results demonstrated the potential of the Vera
framework in improving the management of
safety-critical requirements in agile environments.
This framework could be further evaluated and
refined through additional case studies to
enhance its applicability and effectiveness.

5. Conclusion

The Vera framework offers a promising solution
for managing safety-critical requirements in
Agile environments. By integrating safety stories
to systematically document and track safety
critical issues, the framework provides a
structured approach to ensuring that safety
requirements are effectively addressed throughout
the development process. The framework's
efficiency and adaptability make it a valuable
tool for projects seeking to incorporate safety
considerations into their Agile processes.

Acknowledgments

The authors would like to thank the NASA engineers
who provided valuable insights and feedback during
the course of this study. Their contributions were
instrumental in shaping the framework and ensuring
its effectiveness in practical application.

References

from https://www.siacframework.org/

framework for managing safety-critical requirements
in agile environments. IEEE Transactions on Software
Engineering, 49(3), 377-393.

from https://nasa.gov/missionplanning

This framework has been developed in collaboration
with NASA engineers, ensuring its practicality and
relevance in real-world applications.

The Vera framework provides a structured approach
to managing safety-critical requirements in
agile environments, facilitating better decision-making
and problem-solving. It offers a promising solution
for ensuring that safety is integrated into the
development process, aligning with the principles of
agility and continuous improvement.

Keywords: Safety-Critical Systems, Agile
Methodologies, Safety Stories, Vera Framework,
NASA Mission Planning Scenario.
What is a Research Incubator?

An incubator enables researchers to experiment with a theory or hypothesis in a controlled environment, and to progressively develop the idea until it is ready for testing and deployment in a fully industrial environment.

- Software and Systems Requirements
- Safety Assurance
- Product Lines
- Software & Systems traceability
- Runtime monitoring
- Human studies
## The Eyrie Research Incubator in collaboration with Robyn Lutz

### CISE Users

<table>
<thead>
<tr>
<th>Researcher</th>
<th>Clones one of the incubator projects and uses the runtime environment to support and/or evaluate their own research agenda. e.g., self-adaptation algorithms.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Researcher</td>
<td>Uses prepared bundle of static artifacts to address an open research challenge. e.g., evolution of safety assurance cases across versions of a product.</td>
</tr>
<tr>
<td>Instructor</td>
<td>Creates an assignment using a challenge as an exercise.</td>
</tr>
</tbody>
</table>
New Trajectories bring new challenges

Addresses real problem ✔ Addresses real problem ✔ Addresses real problem ✔ Addresses real problem ✔
Leverages Goldilocks ✔ Leverages Goldilocks ✔ Leverages Goldilocks ✔ Leverages Goldilocks ✔
Well defined benchmarks ✔ Well defined benchmarks ✔ Well defined benchmarks ✔
Adequate data sets ✗ Adequate data sets ✗ Adequate data sets ✔

This is a starting point. It is a long “game”...
We *need* strong collaborations between industry and academic research.
As a community of practitioners and researchers:

**Individually:**
- Be visionary
- Be courageous -- to ask the hard questions.
- Constantly evaluate progress.
- Tackle important questions with potential for real impact.

**As a community:**
- Set vision!
- Be open-minded to innovative work
- Nurture collaborations.
- Tackle inhibitors head on.
- Maintain open communication channels between industry and academia.
Is Requirements Engineering as we know it a relic of the past?

Not unless we stand by waiting for extinction!
REQUIREMENTS ENGINEERING RESEARCH IN A CHANGING WORLD

International Requirements Engineering Conference
Keynote: August 19th, 2018
Banff, Canada

Jane Cleland-Huang, PhD
University of Notre Dame
Dept. of Computer Science and Engineering
http://sarec.nd.edu
JaneClelandHuang@nd.edu

Work described in this talk is funded by the US National Science Foundation under Grants CCF-0959924, CCF-1265178, and CCF-1741781