Requirements Engineering Research in a Changing World

International Requirements Engineering Conference
Keynote: August 22nd, 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
A Requirements Engineering Road Map (2000)

Requirements engineering is the branch of software engineering concerned with the real-world goals, functions of, and constraints on software systems. It is also concerned with the relationship between users to precise specifications of software behavior, and the 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.
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

- Scale; security; tolerance; environmental dependency; self-management; globalization;
- methodologies, patterns, and tools; requirements
- Technologies.
We are bombarded with change on every side...
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...

User Stories, Sprints, Scrum!

The rise of CPS...

Elicitation vs. invention

Requirements discovery by means of feature mining and app reviews.

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 **traceability gap** between what is prescribed and what is delivered:

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

A **formal comparison of five safety-critical software systems** which claimed to follow various standards and guidelines showed **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...

* M. Hamilton, “What the Errors Tell Us”, IEEE Software - Special issue "50 years of Software Engineering"

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

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!
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.

How should I as a Requirements Engineering Researcher respond to these pain points?

Reference the paper

Empirical Software Engineering manuscript No.
(will be inserted by the editor)
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...
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.

Test Case T1
Test Case T2
Test Case T3
Test Case T4
Test Case T5
Test Case T6
Test Case T7
Test Case T8
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?

Research Directions
RD-1.1 Develop prototypes including scenarios of use
RD-1.2 Empirically validate the stakeholders

RD-3.2: Leave no exhaust
Develop techniques that monitor the environment and human activities – and use this information to infer new trace

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.

Trustings Links
As we cannot guarantee complete and accurate traceability, we should devise techniques for clearly communicating confidence levels to the stakeholders.

RD-1.5 Develop human-centric tools to support link vetting.
RD-1.6 Develop algorithms and supporting tools for automatically evaluating the correctness of existing trace links, whether created manually or with tool-support.
RD-1.7 Create visual dashboards to visualize the traceability quality of a project.

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 standard (Purpose).

RD-3.1 More Intelligence.....
Hypothesis: Real advancements, that make a difference to the traceability problem, will be achieved as we transition towards intelligent traceability solutions.

Knowledge Base

RD-3.3 Self Adapting Solutions

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.

Creating and Using Trace Queries

RD-6.1 Integrate traceability into existing development tools
RD-6.2 Provide intuitive forms of query mechanisms 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**.

**What goes through a domain expert’s mind as he or she performs the tracing task?**

**WIU** = Wayside Interface Unit and is located in a Highway Wayside Segment.

Broadcast is similar to transmit.

Automobile controller is part of the automobile.

Both artifacts involve Highway Wayside Segment **sending** wayside status message to automobiles.

WSM and wayside status message are equivalent.
Leverage Deep Learning Techniques

Source $s_1 \ s_2 \ ... \ s_m$

Target $t_1 \ t_2 \ ... \ t_n$

**Word Representation Mapping**

Source $v_{s1} \ v_{s2} \ ... \ v_{sm}$

Target $v_{t1} \ v_{t2} \ ... \ v_{tn}$

**Sentence Semantic Representation**

Source $V_s$

Target $V_t$

**Trace Link Evaluation**

$P_{link}$

**Leverage advances in other fields!**

*Lexical semantic knowledge*

*Variability of linguistic expression*

*Informal reasoning*

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.
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

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

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

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

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

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

Obvious solution is to build industry collaborations...
Dronology Project @ Notre Dame

http://sarec.nd.edu/pages/Dronology.html
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...

http://sarec.nd.edu/pages/Dronology.html

River rescue

Highway Chemical Spill
Team Work

Michael Vierhauser
Architect

Jane Wyngaard
Hardware

Jinghui Cheng
Post Doc / UI

James, Patrick, Michelle
MS student

Joshua Huseman
MS student

Sean Bayley
PhD Student

Alex Madey
Undergraduate

Quinlan McMillan
Undergraduate

Greg Madey
Frank
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.
River Rescue Demo with Dronology
Testing an AED drop
Testing an AED drop
Dronology: Our Crashes

- Bent prop drone
- Rescued Drone
- Trajectory challenged, upside down drone
- Broken leg drone
- Drowned and missing drone
A Near Catastrophe
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.
Artifact tree generated automatically from Jira and Github showing traceability from hazard to code.
Immersive Requirements Engineering

Discovering, Analyzing, and Managing Safety Stories in Agile Projects

Jane Cleland-Huang and Michael Vehrlauer
Department of Computer Science and Engineering
University of Notre Dame
Notre Dame, IN

JaneClelandHuang@nd.edu, michaelvehr@nd.edu

Abstract... Traditionally, safety-critical projects have been viewed using the waterfall process. However, this makes it both costly and challenging to instrumentally introduce new features and to certify the modified product for use. As a result, there has been increasing interest in applying agile development methods to early phase support systems to reduce the number of changes that are needed throughout the project. In this paper, we address the peculiar challenges of discovering, analyzing, and managing safety requirements within the agile software process. We propose a novel approach for identifying safety-related issues in software development that enhances the traditional approach by introducing a new layer of technical expertise and the motion of safety-related issues to be incrementally tracked on the current status of agile projects. We demonstrate the viability of safety stories for managing safety issues in an agile development environment by modeling it as a product in which our arbitrary entity, the safety analyst, creates the safety story. The paper is intended to represent a significant contribution to the safety analyst in modeling a novel approach in the agile development process.

1 Introduction

Systems operating in safety-critical domains, where failures can cause harm or injury, must not only deliver prescribed functionality, but must do so in a way that ensures the safety and health of people. Safety-critical development may be defined as software whose operation must not result in severe harm or death if the system fails to function correctly [1]. Safety-critical systems must meet strict guidelines in order to ensure proper certification [2]. The strict guidelines of the certification process as well as constraints introduced by the rigid deadlines imposed by hardware components have led many organizations to follow a traditional waterfall approach—often resulting in the process of change being unable to meet deadlines set by deadlines set by the project [2]. The apparent cost of change is offset by delivering a product that is difficult to introduce to customers and, in turn, the ability to provide new features or respond to customer needs.

Agile techniques have traditionally been viewed as an alternative to the traditional waterfall development process [3]; however, recently the idea of leveraging agility into the safety-critical development process, and there are numerous accounts reporting its effectiveness and its deployment [4, 5]. Due and Kelly [6] reported findings from a multi-country survey with a total of 31 participants, 67% of whom had experience in safety-critical software development, and 17% with practical experience using a broad range of agile methods. Their survey revealed several insights about particular issues in the requirements process. Respondents strongly suggested that effective management of safety-related requirements is an essential ingredient for successful development of software-intensive systems [7]. In fact, during the development of software-intensive systems, the role of safety stories in the requirements process is often neglected. Therefore, the objective of this paper is to address the need for early phase support systems that avoid managing safety requirements within the agile software process. We propose a novel approach for identifying safety-related issues in software development that enhances the traditional approach by introducing a new layer of technical expertise and the motion of safety-related issues to be incrementally tracked on the current status of agile projects.
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>Role</th>
<th>Activity</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Researcher</strong></td>
<td>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.</td>
</tr>
<tr>
<td><strong>Researcher</strong></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><strong>Instructor</strong></td>
<td>Creates an assignment using a challenge as an exercise.</td>
</tr>
</tbody>
</table>

### Eyrie Research Incubator Projects & Resources

#### Projects

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Dronology</strong></td>
<td>Software Artifacts&lt;br&gt;Physical Elements&lt;br&gt;Runtime Environment&lt;br&gt;Development Environ.</td>
</tr>
<tr>
<td><strong>SafeWalk</strong></td>
<td>Software Artifacts&lt;br&gt;Physical Elements&lt;br&gt;Runtime Environment&lt;br&gt;Development Environ.</td>
</tr>
</tbody>
</table>

#### Challenges

<table>
<thead>
<tr>
<th>Static Artifact Challenges</th>
<th>Runtime Challenges</th>
</tr>
</thead>
<tbody>
<tr>
<td>Formally specify requirements for CPS User Interface.</td>
<td>Student exercise to create and evaluate collision avoidance algorithms.</td>
</tr>
<tr>
<td>Model safety and check safety properties associated with a UI.</td>
<td>Present runtime data in ways that support human decision making.</td>
</tr>
</tbody>
</table>

#### Future Incubator Projects

- Eyrie Website
- Advisory Board
New Trajectories bring new challenges

Semantic Solutions

Evolving Links

Mine Links

Safety Critical Trace..

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

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

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

Addresses real problem ✓
Leverages Goldilocks ✓
Well defined benchmarks ✓
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