



ERICSSON

EIFFEL

ACHIEVING ENTERPRISE SCALE
CONTINUOUS INTEGRATION
AND DELIVERY TRACEABILITY

STEW 2016

HELLO!



- › Daniel Ståhl
- › Ericsson AB
- › Continuous Integration
Subject Matter Expert

- › Original Eiffel Architect

- › Kristofer Hallén
- › Ericsson AB
- › Continuous Integration
Architect

- › Original Eiffel Architect

BASED ON A TRUE STORY



Achieving Traceability in Large Scale Continuous Integration and Delivery

(In press)

Daniel Ståhl, Kristofer Hallén, Jan Bosch

*How can traceability needs of large scale
software development in industry be
effectively addressed in a continuous
integration and delivery context?*

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

Achieving Traceability in Large Scale Continuous Integration and Delivery Deployment, Usage and Validation of the Eiffel Framework

Daniel Ståhl · Kristofer Hallén · Jan Bosch

Received: date / Accepted: date

Abstract The importance of traceability in software development has long been recognized, not only for reasons of legality and certification, but also to enable the development itself. At the same time, organizations are known to struggle to live up to traceability requirements, and there is an identified lack of studies on traceability practices in the industry, not least in the area of tooling and infrastructure. This paper presents, investigates and discusses Eiffel, an industry developed solution designed to provide real time traceability in continuous integration and delivery. The traceability needs of industry professionals are also investigated through interviews, providing context to that solution. It is then validated through further interviews, a comparison with previous traceability methods and a review of literature. It is found to address the identified traceability needs and found in some cases to reduce traceability data acquisition times from days to minutes, while at the same time alternatives offering comparable functionality are lacking. In this work, traceability is shown not only to be an important concern to engineers, but also regarded as a prerequisite to successful large scale continuous integration and delivery. At the same time, promising developments in technical infrastructure are documented and clear differences in traceability mindset between separate industry projects is revealed.

Keywords Continuous integration, continuous delivery, traceability, very-large-scale software systems

1 Introduction

Traceability in software engineering is widely recognized as an important concern, with substantial research and engineering efforts invested in solving its challenges

D. Ståhl · K. Hallén
Ericsson AB, Datalinjen 3, Linköping, Sweden
E-mail: daniel.stahl@ericsson.com, kristofer.hallen@ericsson.com

J. Bosch
Chalmers University of Technology, Lindholmspiren 5, Göteborg, Sweden
E-mail: Jan@JanBosch.com

SOFTWARE CENTER



Software Center

www.software-center.se

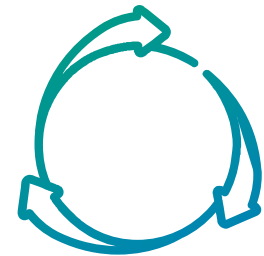
- › 10 companies
 - Ericsson
 - Saab
 - Volvo Group
 - Volvo Cars
 - Axis Communication
 - Tetra Pak
 - Grundfos
 - Jeppesen
 - Verisure
 - Siemens
- › 5 universities
 - Chalmers University
 - University of Gothenburg
 - Malmö University
 - Linköping University
 - Mälardalen University

TRACEABILITY



Traceability is hard, but necessary

- Scale
- Geographic diversity
- Technology and process diversity
- Continuous integration and delivery



THE MANUAL APPROACH



The way we used to do it.

- Put together a team of people.
- Let them work for a month.
- ... figuring out what actually changed.
- ... fixing up documentation.
- ... checking licenses and trade compliance.

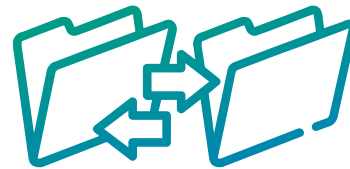


THE PROBLEM



In continuous integration and delivery

- Every day, every developer commits at least once.
- Every change is a release candidate.
- The latest approved release candidate is deployable at a moment's notice.
- Every change is fully documented, checked and analyzed.
- Not a few weeks later. Now.

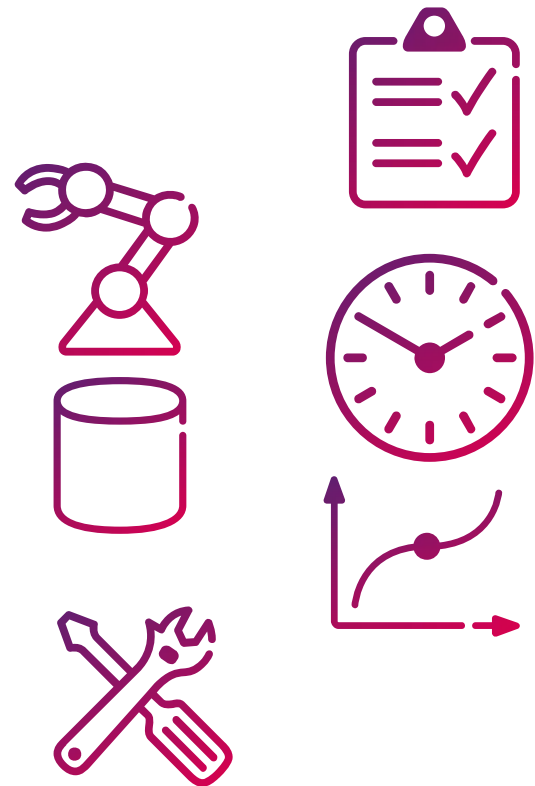


THE SOLUTION



For every step in the software production pipeline

- We need to document what we did, who did it, why we did it and how we did it.
- We need to do it automatically.
- We need to do it in real time.
- We need to store it.
- We need to analyze and visualize it.
- We need to do it regardless of your tools, your technology domain, your location, your organization, your process.

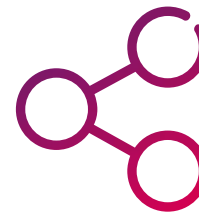
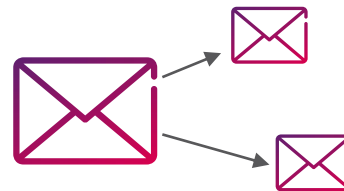


EIFFEL

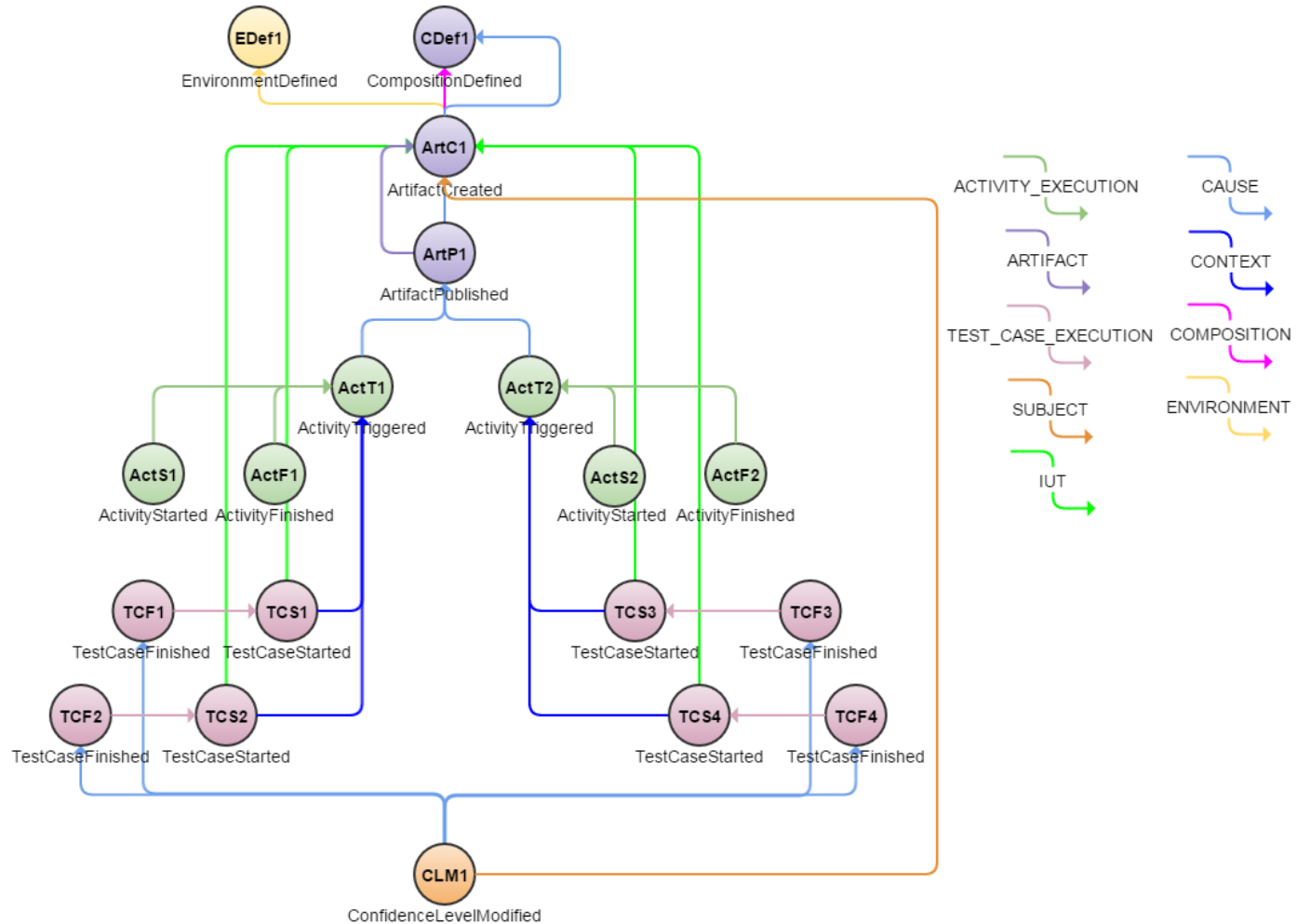


Eiffel is a continuous integration and delivery framework developed by Ericsson.

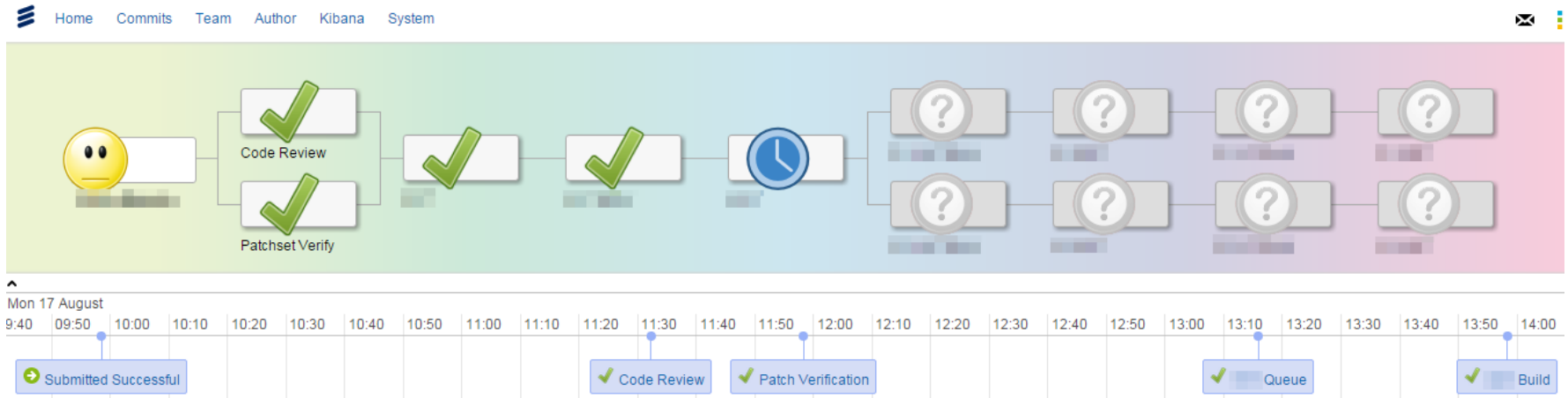
- Every action taken in continuous integration and delivery is an *event*.
- Every event is globally broadcast.
- Every event references other events.
- Every event is immutable.
- The events form a real time Directed Acyclic Graph of everything going on in your software production.



WHAT IT LOOKS LIKE



WHAT IT LOOKS LIKE



Author

Name	[Redacted]
Signum	[Redacted]
E-Mail	[Redacted]
Phonenumber	[Redacted]

Change information

[Redacted]

Tickets

Id	System
[Redacted]	[Redacted]

EVALUATION

We interviewed 15 senior engineers in 3 cases

- One case used Eiffel, two didn't.
- All had substantial traceability needs.
- Non-Eiffel users struggled a lot more!
- We found a significant difference in mindset.
- Developers using Eiffel have a more holistic perspective.



EVALUATION



How does Eiffel compare to the manual approach?

- We repeated an analysis conducted in 2011, prior to the introduction of Eiffel.
- Previous method: mails and phone calls.
- Eiffel method: database query.
- Difference in lead time: 15 days vs. 10 minutes



TESTIMONIES



And what do the engineers have to say?

- “We have achieved transparency in things: [whereas] previously there were a few individuals who could see the status of builds, now [...] you have access to data”
- “In my role it's [now] much, *much* easier to comprehend the situation.”
- “Now when I go into [a tool presenting Eiffel data] and look it's fairly simple. It's practically clickable, so I can drill fairly far down.”



SUMMARY



Continuous integration and delivery traceability is hard, but necessary.

- It can be solved.
- Let's solve it together.
- Solving it can change the way you think about your software pipeline.

JOIN THE FUN!



Eiffel is open source!

- Check out and contribute to the Eiffel protocol on GitHub:
<https://github.com/Ericsson/eiffel>.
- Join the Eiffel Summit 14 October
- Make use of forthcoming open source implementations





ERICSSON