



Justification diagrams in a DevOps context

Sébastien Mosser, Mireille Blay-Fornarino
(MDE + RE) CNRS working group
08/12/2021, Toulouse (virtually)

UQÀM | Département d'informatique

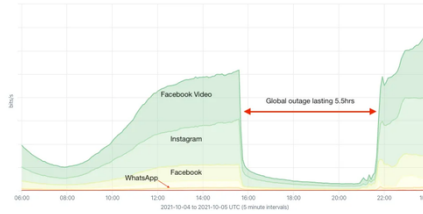
credits photos: Pixabay

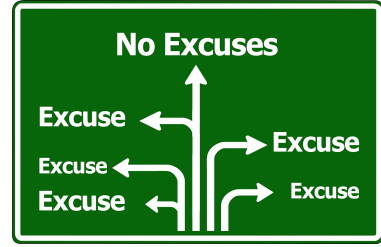
ace

CC BY NC SA

Bird-eye view of the **problem**

Top OTT Service by Average bits/s
Global outage 4-Oct-2021





No Excuses

Excuse

Excuse

Excuse

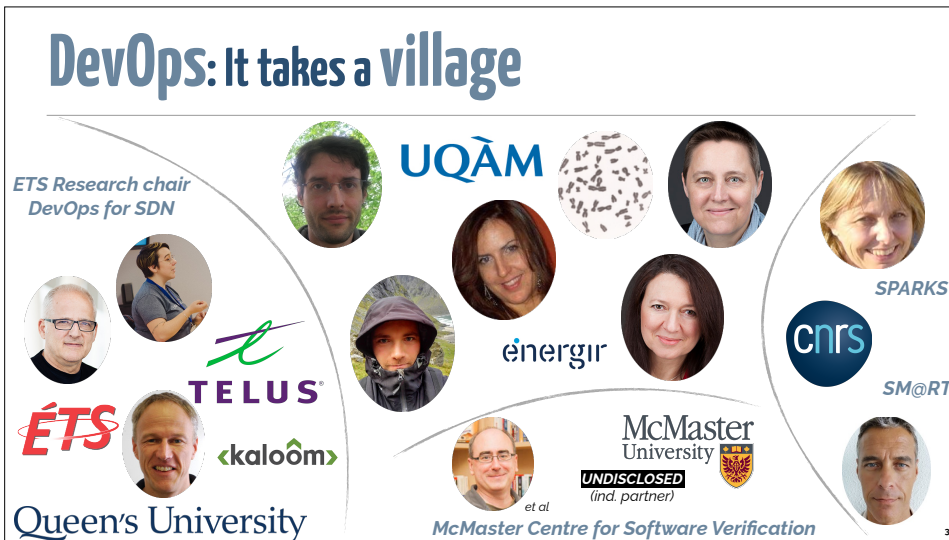
Excuse

Facebook outage (04/09/2021)

UQÀM | Département d'informatique

https://en.wikipedia.org/wiki/2021_Facebook_outage

DevOps: It takes a village



ETS Research chair
DevOps for SDN

UQÀM

SPARKS

energir

cnrs

SM@RT

TELUS

McMaster University

UNDISCLOSED (ind. partner)

et al

McMaster Centre for Software Verification

ETS

Queen's University

kaloôm

1 DevOps w.r.t. MDE & RE ?

2 Justifying or Explaining?

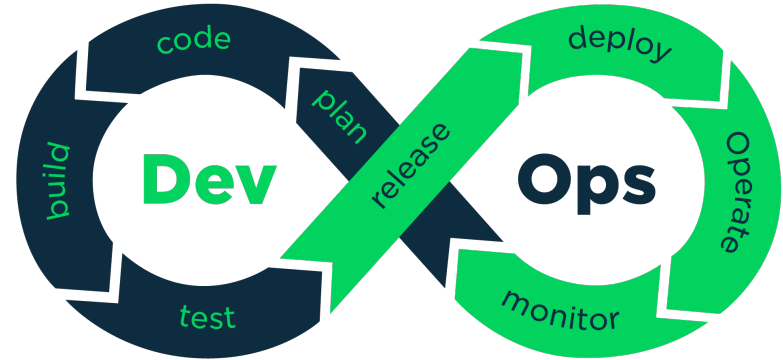
3 Preliminary results

DevOps w.r.t. MDE & RE



Context: DevOps lifecycle

Wait, this is not the "Software Velocity" working group! 🤔



Blatant advertisement (RE)

2021 RE 29th IEEE International Requirements Engineering Conference
 Notre Dame, South Bend, USA
 September 20-24, 2021



UQÀM
 Université du Québec à Montréal



Université Fédérale
 Toulouse
 Midi-Pyrénées

Requirements Engineering in the DevOps Era

Jean-Michel Bruel (Univ. Of Toulouse)
 Sébastien Mosser (UQAM)
 RE 21 - Notre Dame (US) / Virtual

Have I said blatant (MDE)?



Requirements Engineering for the Ageing Population: a Teaching Perspective

Alexandra Lachapelle-Dagenais¹, Sébastien Mosser², Anne-Marie Pina-Drey³ and Mireille Blay-Fornarino⁴

¹Université du Québec à Montréal (UQAM), Montréal, Canada.
 email: [lastname.firstname]@uqam.ca
²Université Côte d'Azur, CNRS, Sophia Antipolis, France.
 email: [lastname.LASTNAME]@univ-cotedazur.fr

TABLE I: 65+ citizens in Canada [2]

| Year | 1976 | 1981 | 1986 | 1991 | 2001 | 2016 |
|--|-------|-------|-------|-------|-------|-------|
| Millions | 3.96 | 4.39 | 4.96 | 5.22 | 5.70 | 6.15 |
| Female | 2.00 | 2.22 | 2.42 | 2.52 | 2.70 | 2.85 |
| (a) Percentage of citizens from 1976 to 2016 | | | | | | |
| Year | 1976 | 1981 | 1986 | 1991 | 2001 | 2016 |
| Female | 24.6% | 26.0% | 26.4% | 26.6% | 26.7% | 26.9% |
| Female | 25.4% | 26.9% | 27.4% | 27.6% | 27.7% | 27.9% |
| (b) Estimated percentage in 2036 (three growth models) | | | | | | |

Abstract—Elders occupy an ample space among the potential end-users of the pieces of software we are designing. This situation is only increasing, and even the pessimistic projections forecast global aging in the World population. In this context, it is our duty as software engineers to design and implement software adapted to elders. However, this particular problem is too often bypassed in the classical software engineering curriculum. In this paper, we report about a set of courses dedicated to requirements engineering for the ageing population, currently implemented at Université Côte d'Azur (C.A. France) and Université du Québec à Montréal (UQAM, Canada). We identify challenges related to the ageing population that can be addressed in a course and report on how such courses are implemented in both institutions. The work reported in this paper is validated through an in-depth case study of CCA, based on 18 years of experience in the teaching of requirement engineering for specific populations.

the Azur-Well² nationwide network of centres of excellence regrouping 42 universities, 400 industrial partners and 200 researchers working together in the context of "technology and ageing". In addition to the research and product incubation areas, the network dedicates resources to support training

UQÀM
 Université du Québec à Montréal

Towards Modelling Acceptance Tests as a Support for Software Measurement

Alexandra Lapointe-Boisvert
 Université du Québec à Montréal
 Montréal, Canada
 lapointe-boisvert.alexandra@uqam.ca

Sébastien Mosser
 Université du Québec à Montréal
 Montréal, Canada
 mosser.sebastien@uqam.ca

Sylvie Trudel
 Université du Québec à Montréal
 Montréal, Canada
 trudel.s@uqam.ca

Abstract—The DevOps paradigm emphasizes the need for a measurable feedback loop, starting from requirements and going as far as deployment in an automated way. In this context, a modelling challenge is to leverage the existing requirement engineering approaches to support measurements. Unfortunately, measurement methods are slow and costly by definition, preventing precisely measured requirements from being used in the DevOps loop. As a result, developers have to deal with grossly estimated elements, e.g., using story points presented by agile methods. Thus, it is not possible to provide better support for the development team. We envision taking advantage of the artifacts that already exist in a DevOps context to provide better support for requirements measurement, making it available in an automated context such as the DevOps one. This paper focuses on the automated analysis of acceptance tests (e.g., expressed using the Gherkin language) to support functional measurement automation in a DevOps context. This proposition is illustrated by a scenario coming from an industrial partner, supporting the identification of four research challenges to be tackled.

are required and act as functional requirements that prevent duplication of requirements writing. Eventually, the DevOps paradigm reaches a paradox here, by emphasizing the need to improve while still relying on clumsy measurements to support such improvement at the requirements level. More precise way of measuring software exists (e.g., the COSMIC method [2]), with demonstrated benefits in terms of planning and project implementation [3]. Unfortunately, the effort necessary to measure a given piece of software according to these methods is tremendous, making it close to impossible to integrate such approaches in a DevOps context.

Any research effort in this direction should take into consideration the state of practice: while relying on informal artifacts as inputs, there is a need for better measurements to support the DevOps feedback loop. As a consequence, the global research question to be addressed is the following:

DevOps@MODELS workshop series



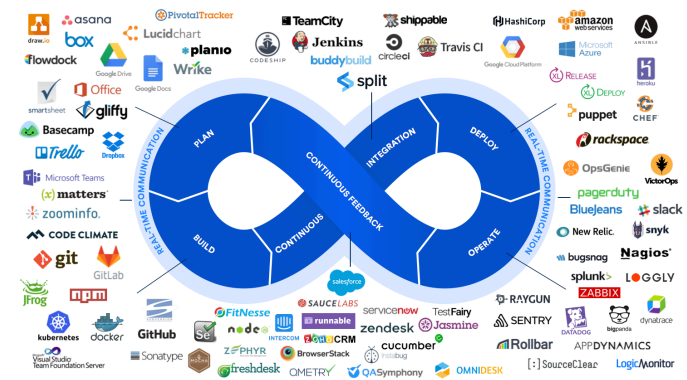
- 2019:
 - 35 participants
- 2020: **Cancelled (COVID-19)**
- 2021:
 - 60 participants
 - 2 industrial keynotes



UQÀM | Département d'informatique



What people see when talking about DevOps

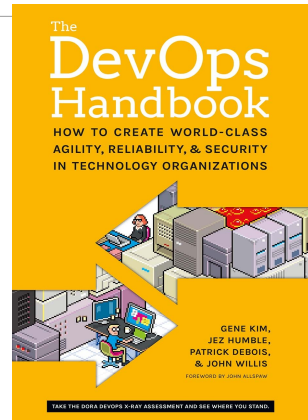


UQÀM | Département d'informatique

Plot twist!

Myth—DevOps is Just “Infrastructure as Code” or Automation: While many of the DevOps patterns shown in this book require automation, DevOps also requires cultural norms and an architecture that allows for the shared goals to be achieved throughout the IT value stream. This goes far beyond just automation.

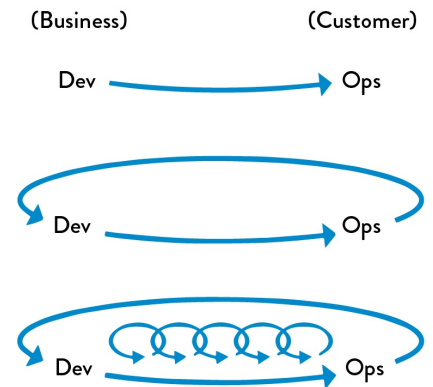
“DevOps isn’t about automation, just as astronomy isn’t about telescopes.”



UQÀM | Département d'informatique

The Three Ways of DevOps

- **Flow**
 - From Development to Operations
 - Identify Value Streams
 - Maximize / Speedup streams
- **Feedback**
 - Amplify feedback
 - Problems must not happen again
- **Continuous Learning**
 - Disciplined approach
 - Support risk taking / experimentation



UQÀM | Département d'informatique

[DevOps Handbook]

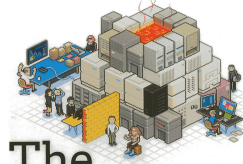
“DevOps isn’t about automation, just as astronomy isn’t about telescopes.”

- John Willis

13

Dealing with a Release failure

From the authors of *The Visible Ops Handbook*



The Phoenix Project

A Novel About IT, DevOps, and Helping Your Business Win

Gene Kim, Kevin Behr, and George Spafford

UQÀM | Département d’informatique



14

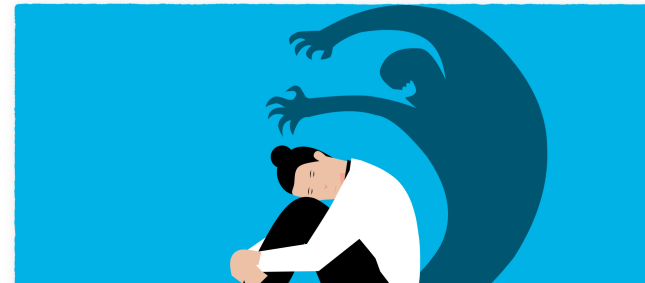
Justifying or Explaining?
(the RELAI project)

2

UQÀM | Département d’informatique

15

Example: e-Health & DevOps



RELAJ: Where DevOps meet Mental (e-)Health

UQÀM | Département d’informatique

Sébastien Mosser
Université Côte d’Azur (virtually)
22/10/2021



New Frontiers in Research Fund
Fonds Nouvelles frontières en recherche



Marie-Jean Meurs

UQÀM | Département d’informatique

16

Vision: Frontline service for mental health



Collect an annotated corpus of textual conversations, from patients received by the emergency units in Switzerland and Belgium

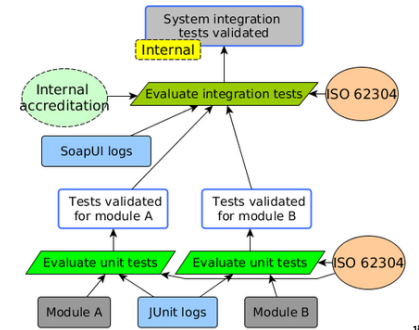
Train a model to support the diagnostic of suicidal risk

Challenge: Convincing Ethic committees

- Need to obtain **three different ethics committees certifications**
- **(Super)** Long & **(Super)** tedious process (*multi-days meeting in 🇨🇦, early 2020*).

• Looks like a **classical "certification"** process

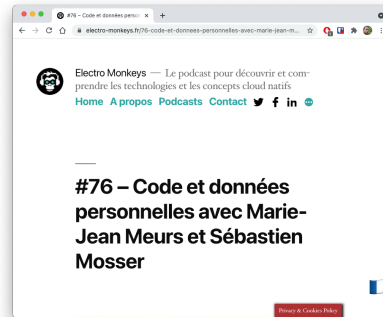
- **Insights (ongoing):**
- Strengthen the DMP
- Give confidence to defend it
- **Even if not executable!**



Challenge: Automated Support for developers

```
public void method store(Set<Entry> dataset) {
    Database db = RemoteStorage("...");
    for(Entry e: dataset) {
        db.save(e)
    }
}
```

Error:
Entry 'e' is not
anonymized



Enhance tooltips and error handlers in IDEs with **composable requirements** that are driven by external & evolving concerns

To "explain" or to "justify"?

explain verb

Save Word
explain | \ɪkˈspleɪn
explained, explaining, explains

Essential Meaning of explain

- 1 : to make (something) clear or easy to understand
// I need a lawyer to *explain* this contract to me.
// The professor *explained* the poem to the class.
See More Examples
- 2 : to tell, show, or be the reason for or cause of something
// Scientists could not *explain* the strange lights in the sky.
// I don't know how to *explain* the dog's strange behavior.
See More Examples

justify verb

Save Word
justify | \ˈjʌ-stə-fi
justified, justifying

Essential Meaning of justify

- 1 : to provide or be a good reason for (something) : to prove or show (something) to be just, right, or reasonable
// He tried to *justify* his behavior by saying that he was being pressured unfairly by his boss.
// The fact that we are at war does not *justify* treating innocent people as criminals.
// It's hard to *justify* the cost of a new car right now. = It's hard to *justify* spending money on a new car right now.
- 2 : to provide a good reason for the actions of (someone)
// Why should I have to *justify myself* [=to provide an explanation for my actions] when it was their fault?

The point is not to “explain”
the release process

It is clearly to “justify” it!

Preliminary Results

3

Justification diagram language



@startuml

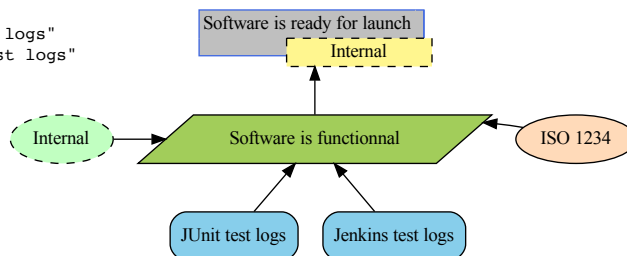
```

conclusion C = "Software is ready for launch" - "Internal"
strategy S = "Software is fonctionnal"
domain D = "Internal"
rationale R = "ISO 1234"
support A = "JUnit test logs"
support B = "Jenkins test logs"
  
```

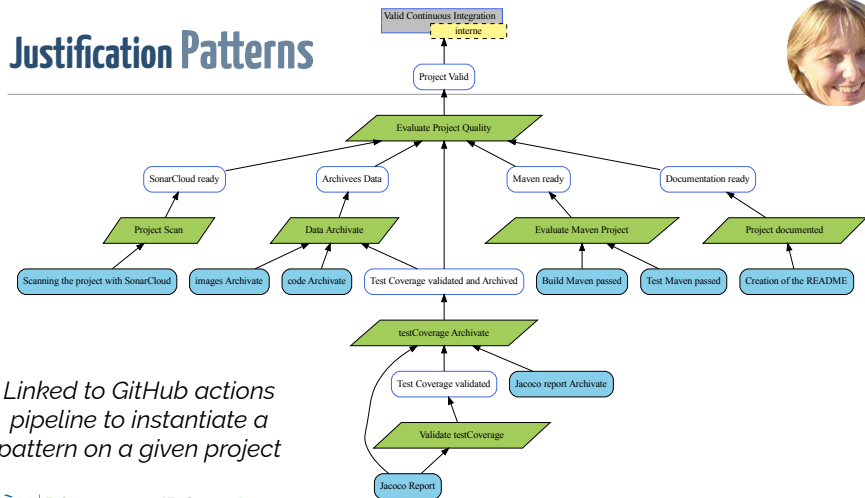
```

S --> C
D --> S
R --> S
A --> S
B --> S
  
```

@enduml

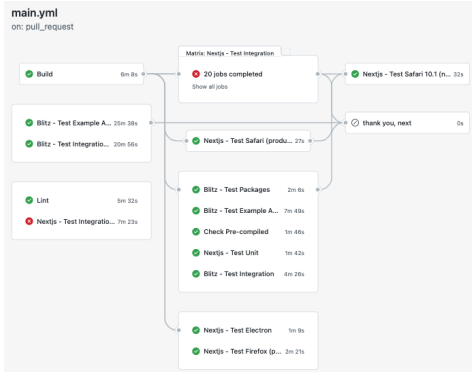


Justification Patterns



Linked to GitHub actions pipeline to instantiate a pattern on a given project

Extracting Justification diagrams from pipelines



the **Nit** Programming Language
a fun language for serious programming



Wrap-up

- **DevOps is not just about automation**
 - Strong relationship with MDE & RE
- Even if "**explainability**" is a research cash cow
 - we actually need to **justify!**
- **Justification Diagrams** can **model** such a process
 - **Preliminary results** applied to GitHub Actions
 - **Extracting DevOps requirements** from legacy pipelines
 - **Open-source DSL** available for Justification Diagrams

Blatant Advertisement

UQÀM
Université du Québec à Montréal



McMaster
University



Join us!



Open positions

- **8 prof positions:**
 - 6 *assistant/associate* @**McMaster** (open profile)
 - 2 *assistant/associate* @**UQAM** (HCI / System)
- **5+ postdocs** positions
- **6+ PhDs** positions
- **7+ M.Sc./M.Eng.** (+*internships*)



Open Postdoc positions

- Model-Based **Development of Assured Autonomous Vehicles**
- Certification and Assurance of **Security and Safety Properties**
- **Mixed-Criticality Systems and Dynamic Risk Assessment** via Machine Learning
- **Optimizing Performance and Architecture** of Large-Scale Cloud-Based Fare Collection Systems
- **Migration** from Decentralized to Centralized **Automotive E/E Architectures**
- Model-Driven Engineering (MDE) for **Mobile Health Applications**
- **Usability, Accessibility and Design Languages** for Wayfaring in Transport Systems

Credits

- **Francis Bordeleau** (ÉTS, Montréal, )
- **Juergen Dingel** (Queen's University, Kingston, )
- **Corinne Pulgar** (ÉTS, Montréal, )
- **Jean Privat** (UQAM, Montréal, )
- **Alexandre Lapointe-Boisvert** (UQAM, Montréal, )
- **Marie-Jean Meurs** (UQAM, Montréal, )
- **Vladimir Reinharz** (UQAM, Montréal, )
- **Richard Paige** (McMaster, Hamilton, )
- **Vera Pantelic** (McMaster, Hamilton, )
- **Mireille Blay-Fornarino** (UCA, Nice, )
- **Jean-Michel Bruel** (Univ. Of Toulouse, Toulouse, )