Building on an unsound foundation: How release pipelines can impact our predictive models

Bram Adams

POLYTECHNIQUE MONTRÉAL

Shane McIntosh

McGill

shanemcintosh@acm.org

@shane_mcintosh

shanemcintosh.org

I’ve worn a lot of hats in the past

I’ve worn a lot of hats in the past

Researcher

McGill

Asst. Professor:

2015-Present

I’ve worn a lot of hats in the past

Researcher

McGill

Asst. Professor:
2015-Present

Queen's University

PhD:
2012-2015

I’ve worn a lot of hats in the past

Researcher

McGill

Asst. Professor:
2015-Present

Queen's UNIVERSITY

PhD:
2012-2015

Practitioner

EMC^2
where information lives

SW Eng:
2010-2012

I’ve worn a lot of hats in the past

Researcher

McGill

Asst. Professor:
2015-Present

Queen's University

PhD:
2012-2015

Practitioner

EMC² where information lives

SW Eng:
2010-2012

Rockstar

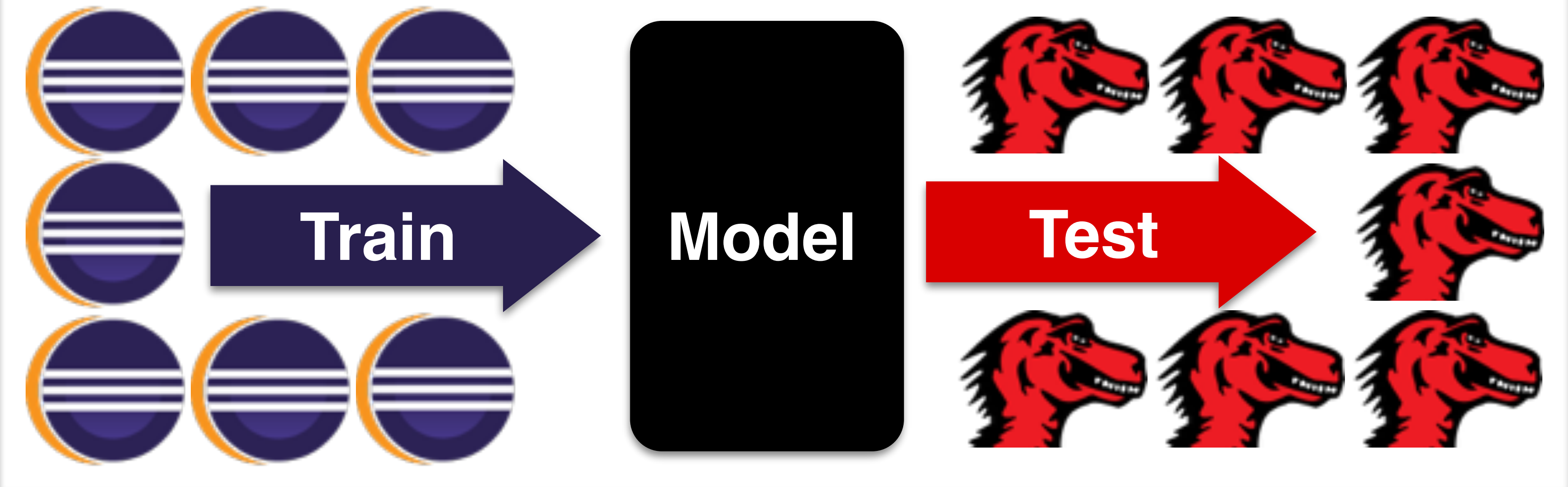
ROSESDEAD

Drummer:
2003-2008

Our past work on predictive modelling

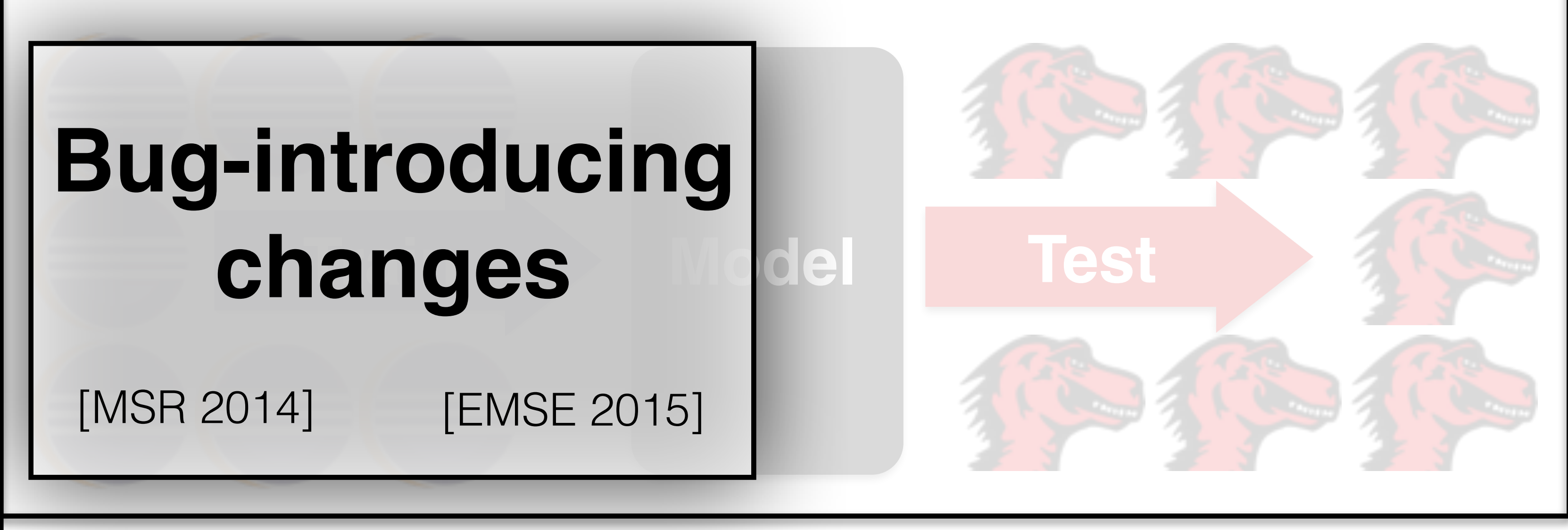
Our past work on predictive modelling

Cross-project modelling



Our past work on predictive modelling

Cross-project modelling



Bug-introducing changes

[MSR 2014] [EMSE 2015]

Our past work on predictive modelling

Cross-project modelling

Bug-introducing changes

[MSR 2014] [EMSE 2015]

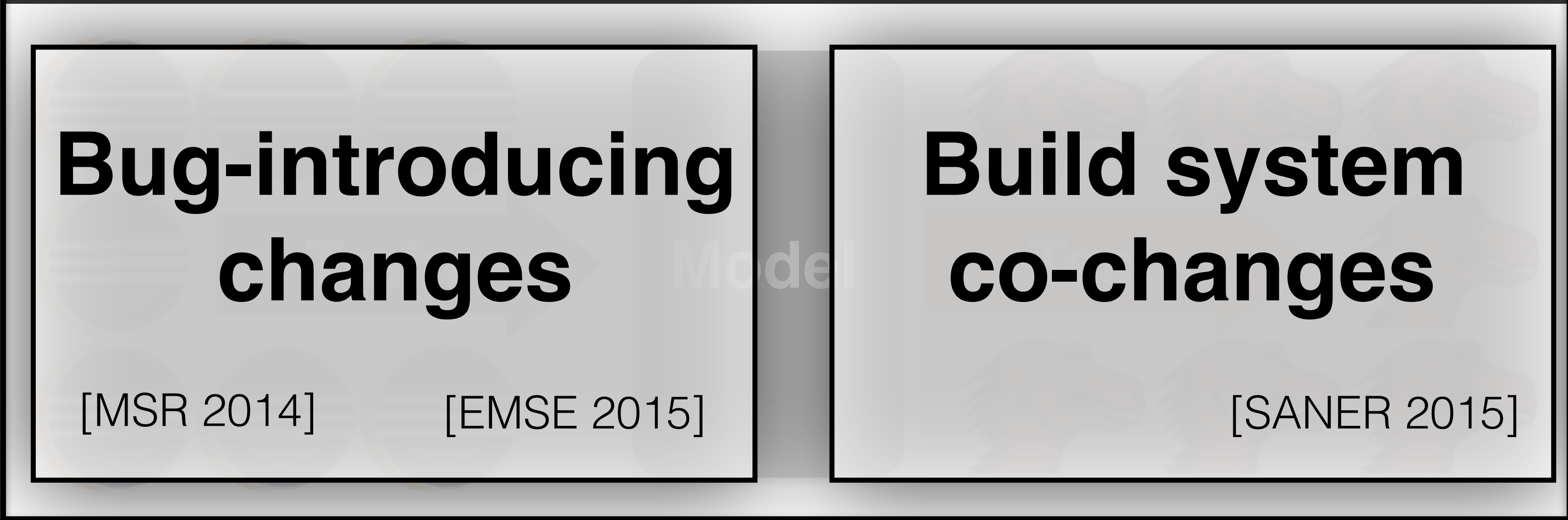
Model

Build system co-changes

[SANER 2015]

Our past work on predictive modelling

Cross-project modelling



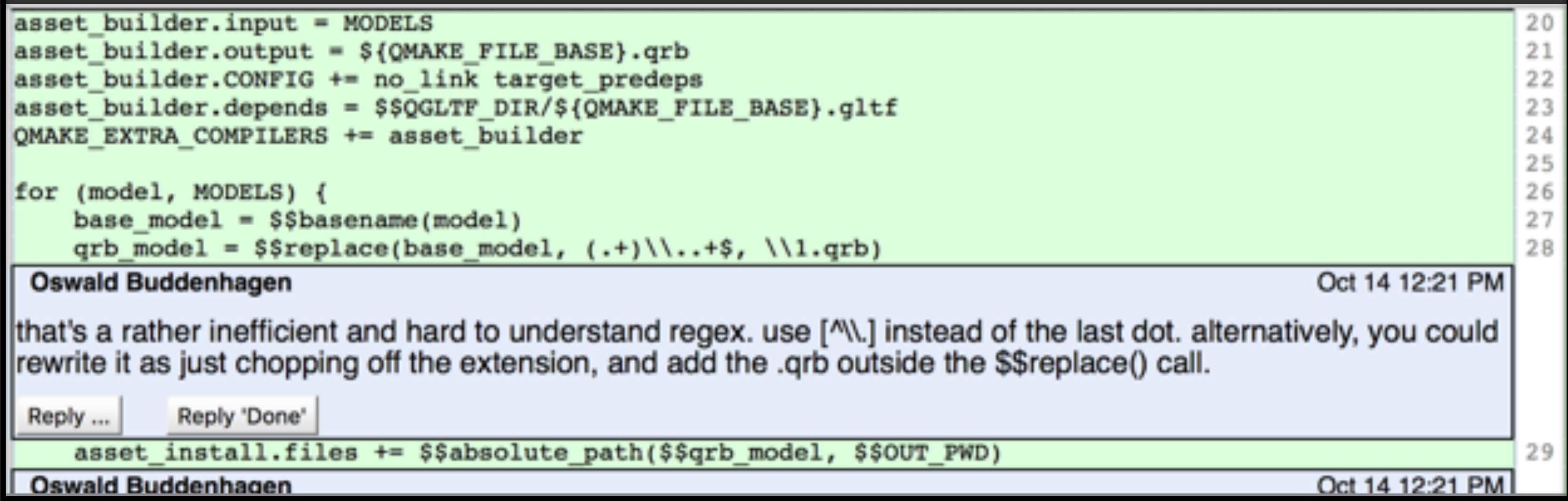
Bug-introducing changes

[MSR 2014] [EMSE 2015]

Build system co-changes

[SANER 2015]

Impact of modern code review



```
asset_builder.input = MODELS
asset_builder.output = ${QMAKE_FILE_BASE}.qrb
asset_builder.CONFIG += no_link target_predeps
asset_builder.depends = $$QGLTF_DIR/${QMAKE_FILE_BASE}.gltf
QMAKE_EXTRA_COMPILERS += asset_builder

for (model, MODELS) {
    base_model = $$basename(model)
    qrb_model = $$replace(base_model, (.+)\\..+$, \\1.qrb)
```

Oswald Buddenhagen Oct 14 12:21 PM

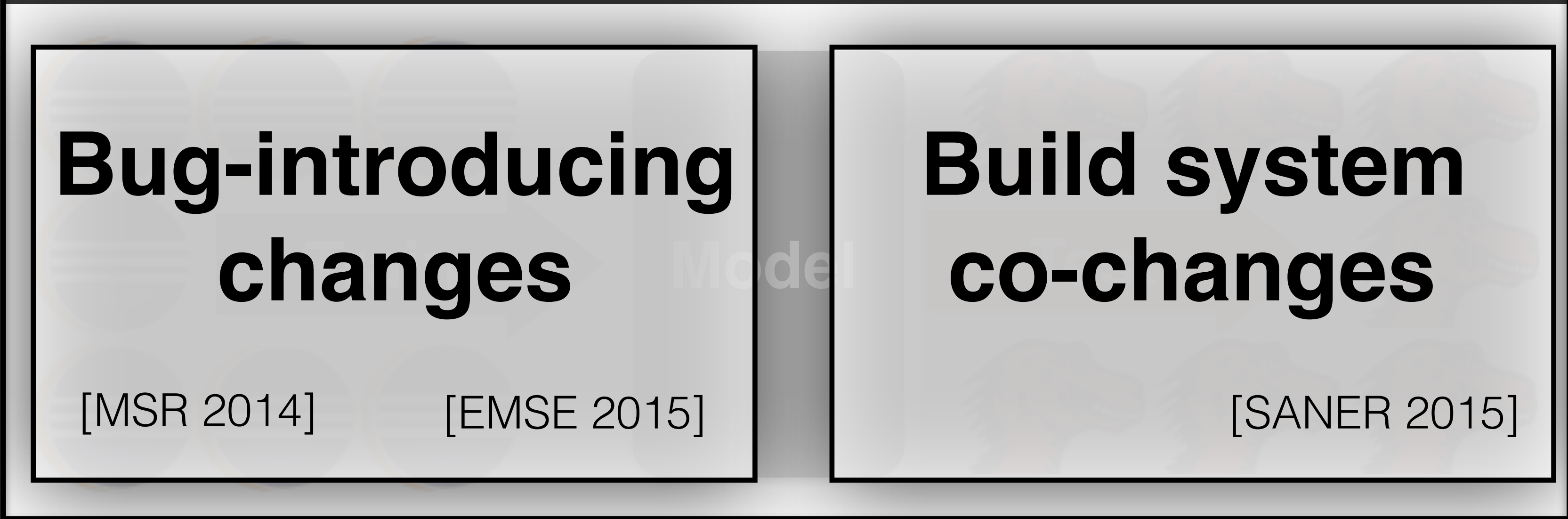
that's a rather inefficient and hard to understand regex. use [^\\.] instead of the last dot. alternatively, you could rewrite it as just chopping off the extension, and add the .qrb outside the $$replace() call.


```
    asset_install.files += $$absolute_path($$qrb_model, $$OUT_PWD)
```

Oswald Buddenhagen Oct 14 12:21 PM

Our past work on predictive modelling

Cross-project modelling



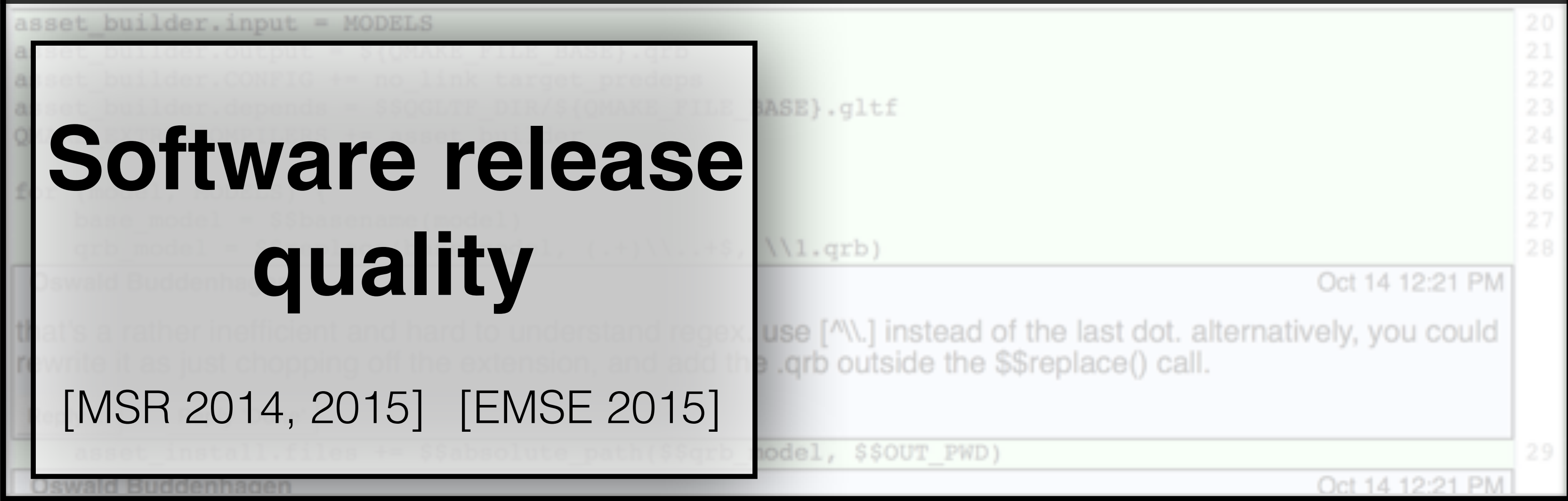
Bug-introducing changes

[MSR 2014] [EMSE 2015]

Build system co-changes

[SANER 2015]

Impact of modern code review



```
asset_builder.input = MODELS
asset_builder.output = ${QMAKE_FILE_BASE}.qrb
asset_builder.CONFIG += no_link target_predeps
asset_builder.depends = $$QGLTF_DIR/${QMAKE_FILE_BASE}.gltf
QMAKE_EXTRA_COMPILERS += asset_builder

for(model, MODELS) {
    base_model = $$basename(model)
    qrb_model = $$replace(base_model, (.*)\\..+$, \\1.qrb)
```

Oswald Buddenhagen Oct 14 12:21 PM

that's a rather inefficient and hard to understand regex. use [^\\.] instead of the last dot. alternatively, you could rewrite it as just chopping off the extension, and add the .qrb outside the $$replace() call.

```
    asset_install.files += $$absolute_path($$qrb_model, $$OUT_PWD)
```

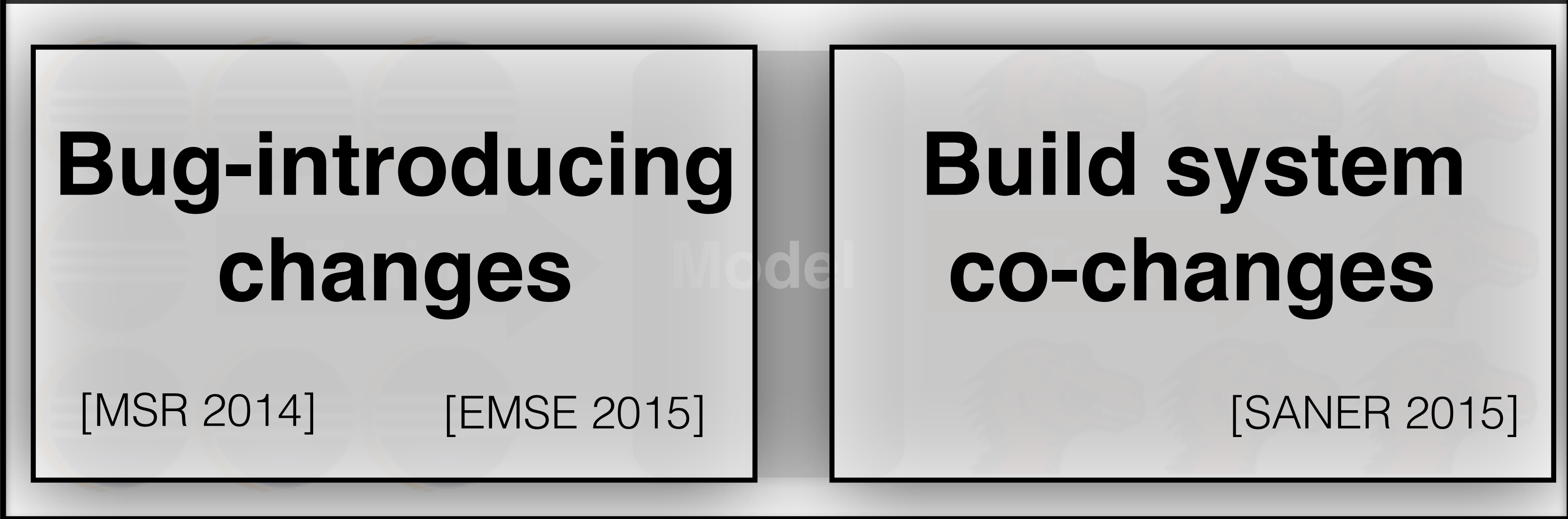
Oswald Buddenhagen Oct 14 12:21 PM

Software release quality

[MSR 2014, 2015] [EMSE 2015]

Our past work on predictive modelling

Cross-project modelling



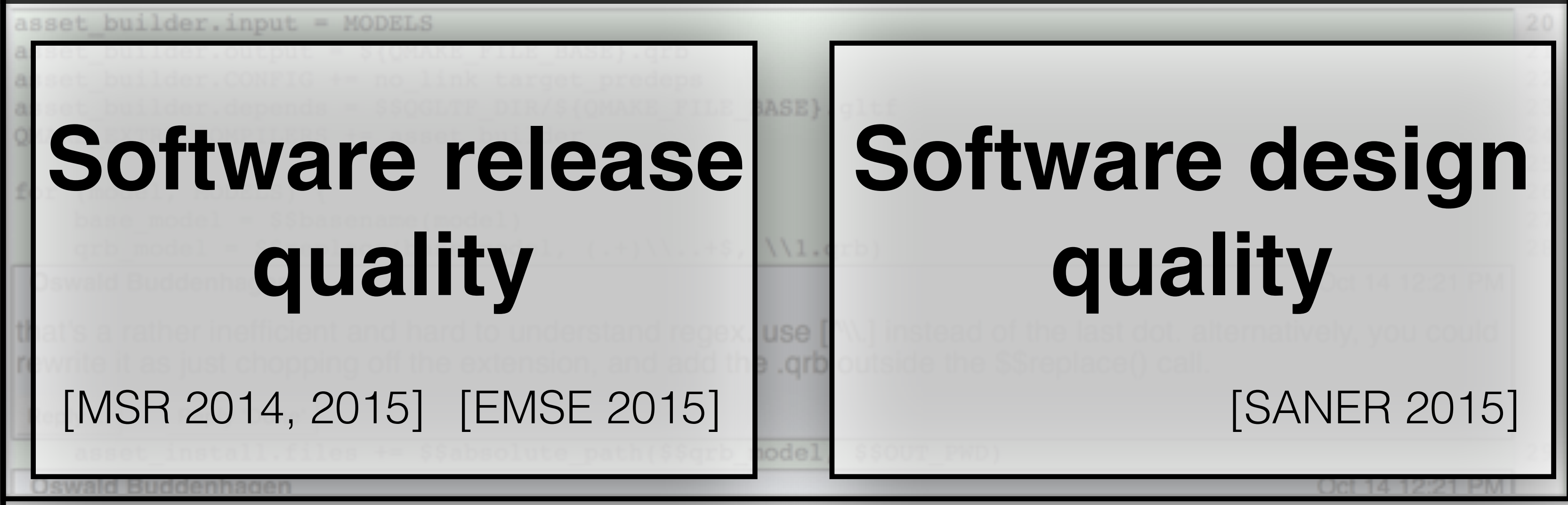
Bug-introducing changes

[MSR 2014] [EMSE 2015]

Build system co-changes

[SANER 2015]

Impact of modern code review



```
asset_builder.input = MODELS
```

Oswald Buddenhagen Oct 14 12:21 PM

Software release quality

[MSR 2014, 2015] [EMSE 2015]

Software design quality

[SANER 2015]

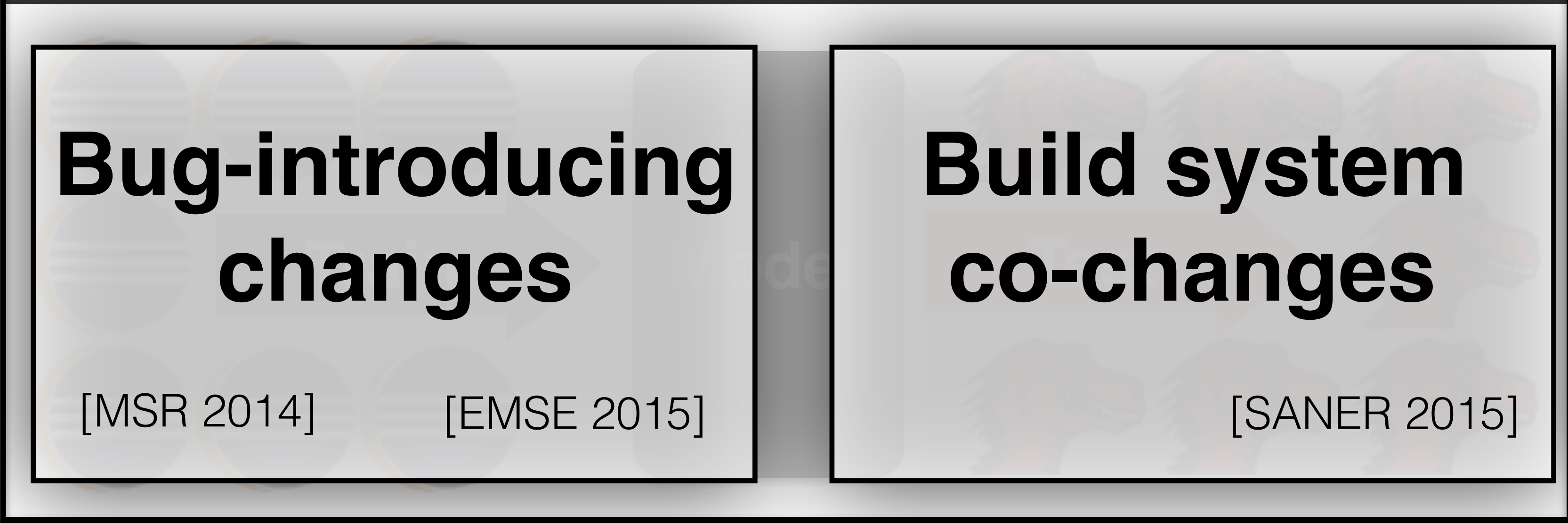
Our models may be trained on unsound data

Our models may be trained on unsound data

Modelling problems

Our past work on predictive modelling

Cross-project modelling



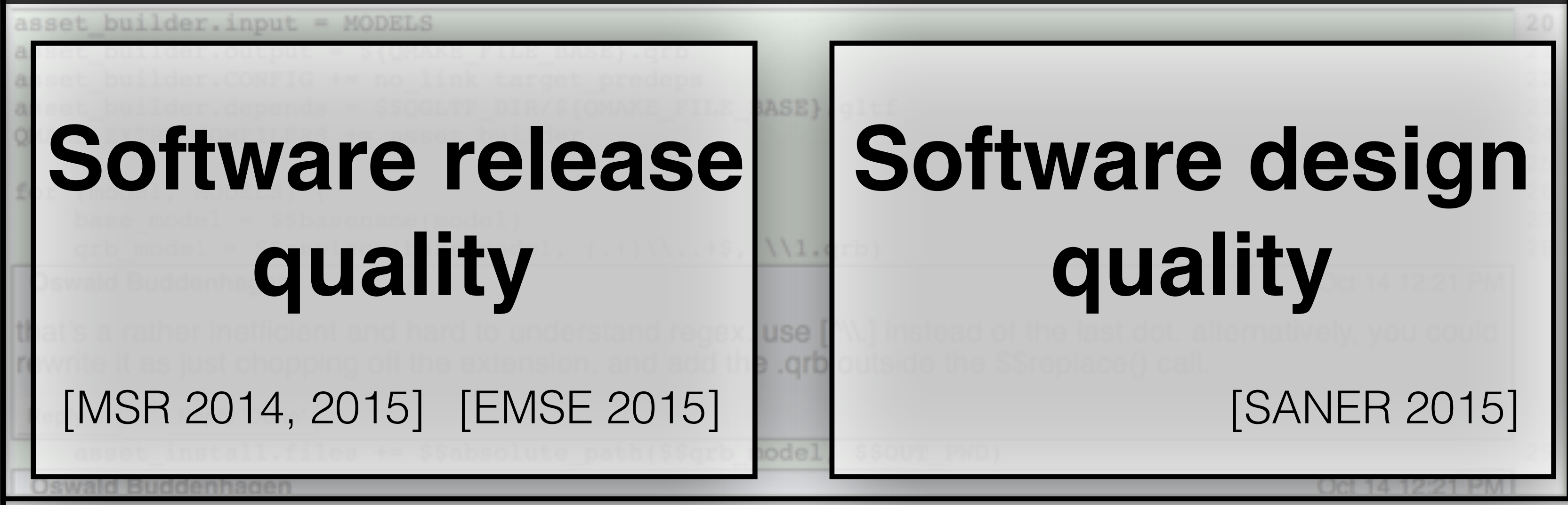
Bug-introducing changes

[MSR 2014] [EMSE 2015]

Build system co-changes

[SANER 2015]

Impact of modern code review



Software release quality

[MSR 2014, 2015] [EMSE 2015]

Software design quality

[SANER 2015]

Our past work on predictive modelling

Cross-project modelling

Bug-introducing changes

[MSR 2014] [EMSE 2015]

Build system co-changes

[SANER 2015]

Impact of modern code review

Software release quality

[MSR 2014, 2015] [EMSE 2015]

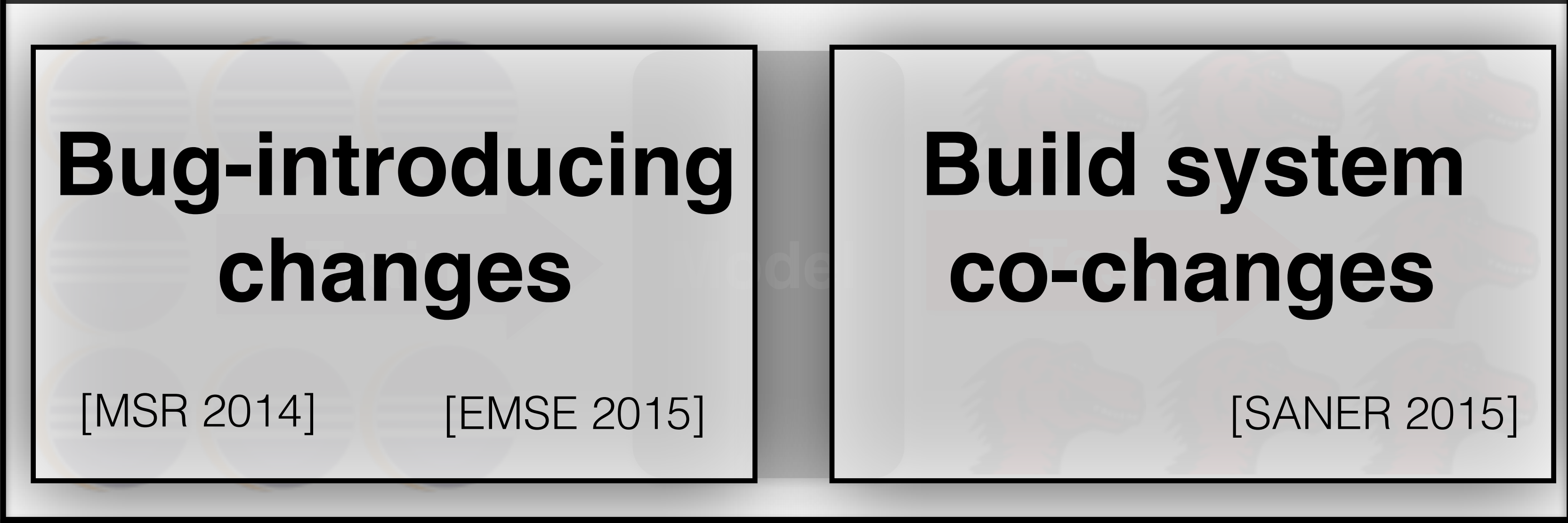
Software design quality

[SANER 2015]

Impact of experimental setup

Our past work on predictive modelling

Cross-project modelling



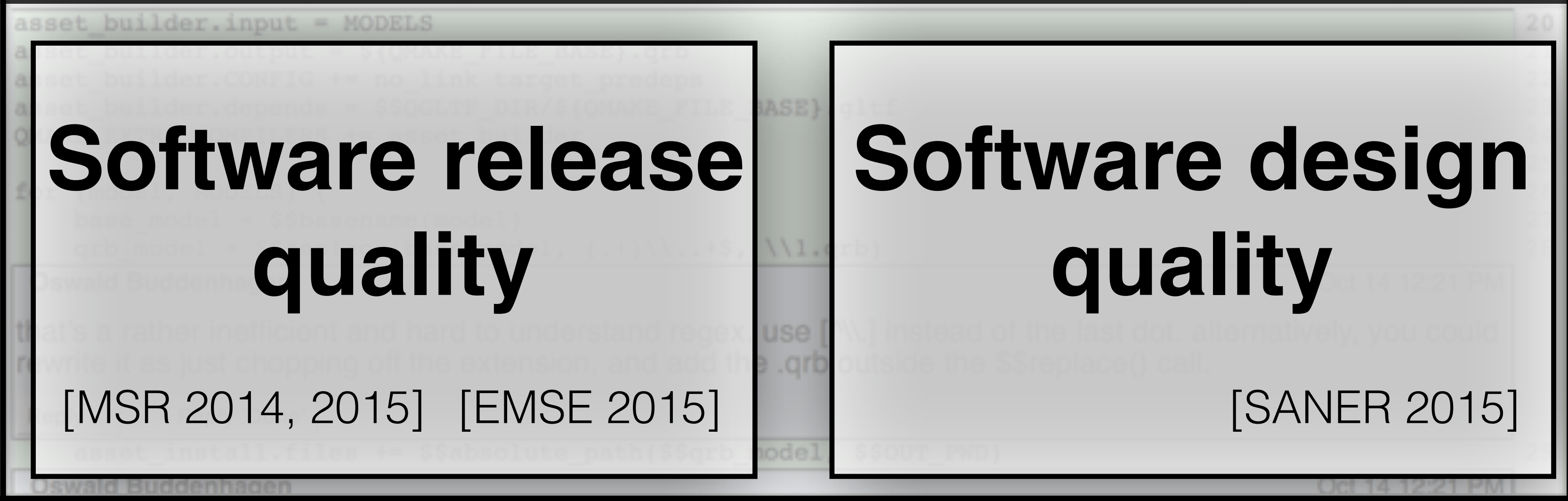
Bug-introducing changes

[MSR 2014] [EMSE 2015]

Build system co-changes

[SANER 2015]

Impact of modern code review



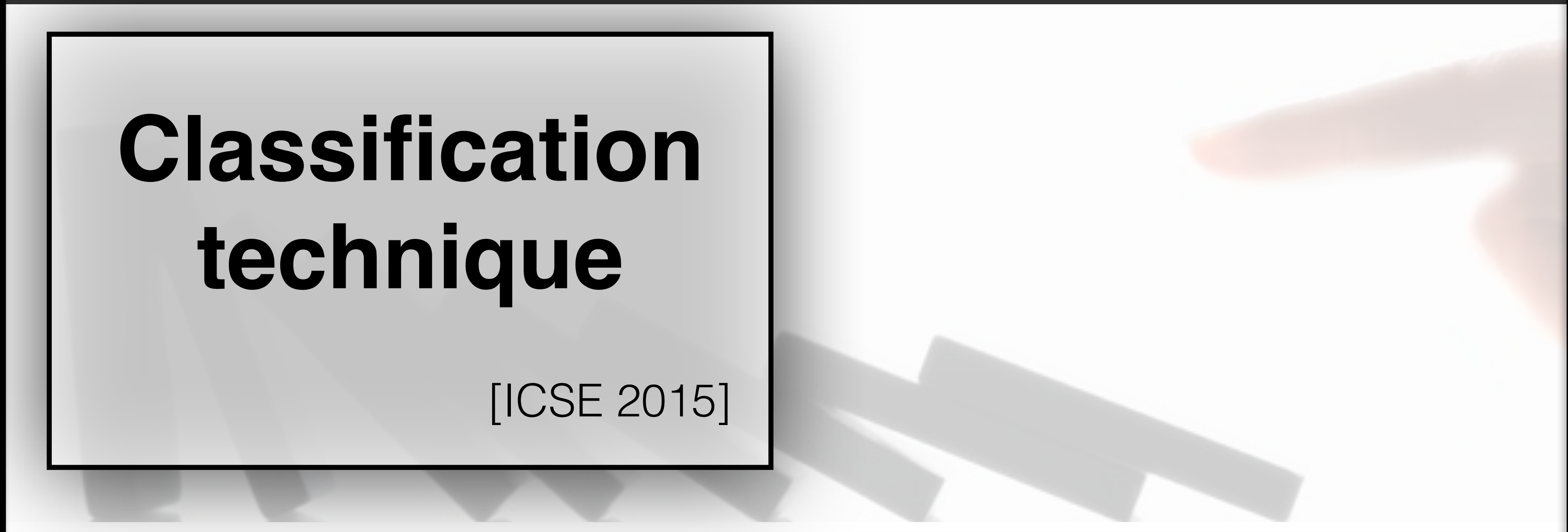
Software release quality

[MSR 2014, 2015] [EMSE 2015]

Software design quality

[SANER 2015]

Impact of experimental setup



Classification technique

[ICSE 2015]

Our past work on predictive modelling

Cross-project modelling

Bug-introducing changes

[MSR 2014] [EMSE 2015]

Build system co-changes

[SANER 2015]

Impact of modern code review

Software release quality

[MSR 2014, 2015] [EMSE 2015]

Software design quality

[SANER 2015]

Impact of experimental setup

Classification technique

[ICSE 2015]

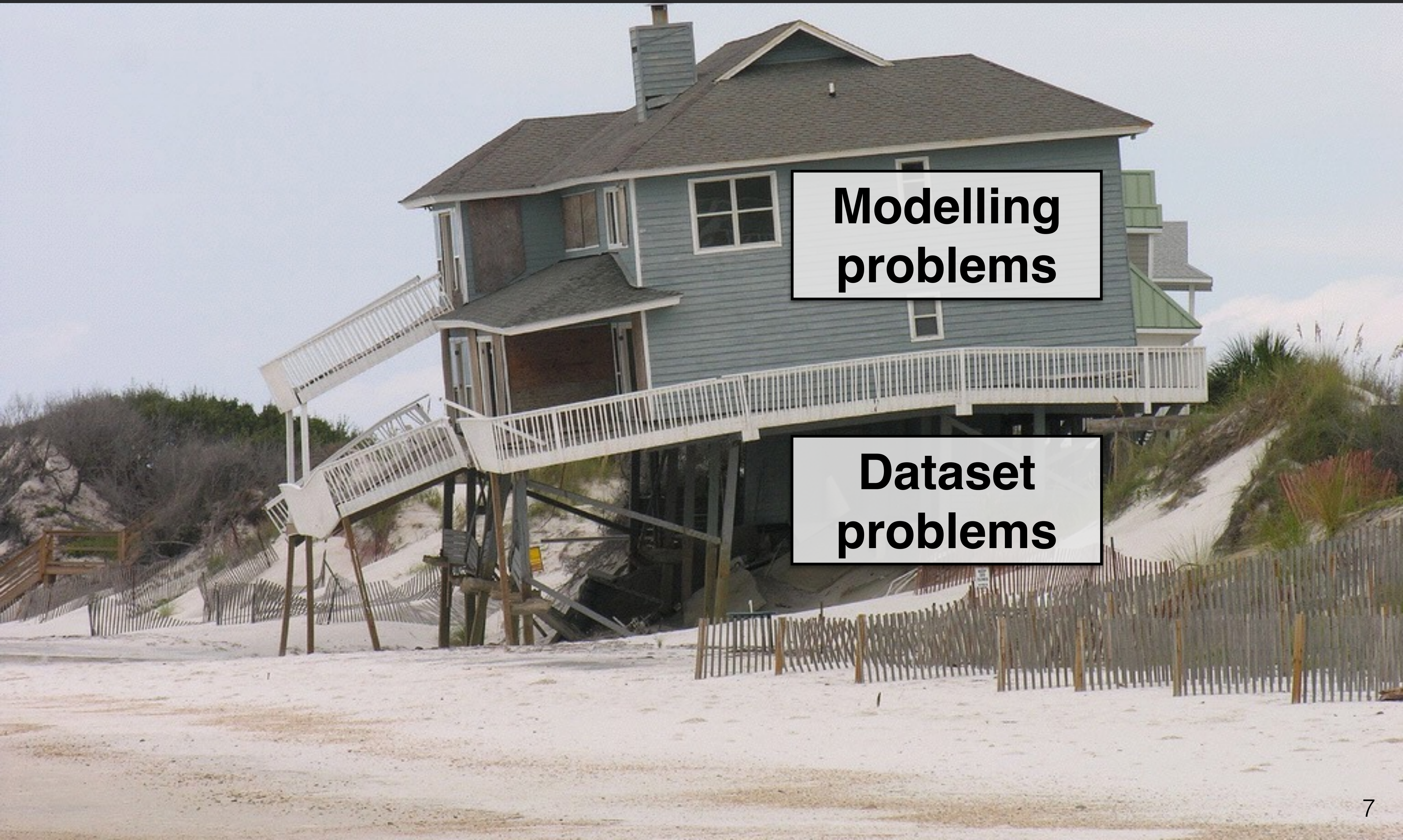
Classification technique parameters

[201?]

Our models may be trained on unsound data

Modelling problems

Our models may be trained on unsound data



Modelling problems

Dataset problems

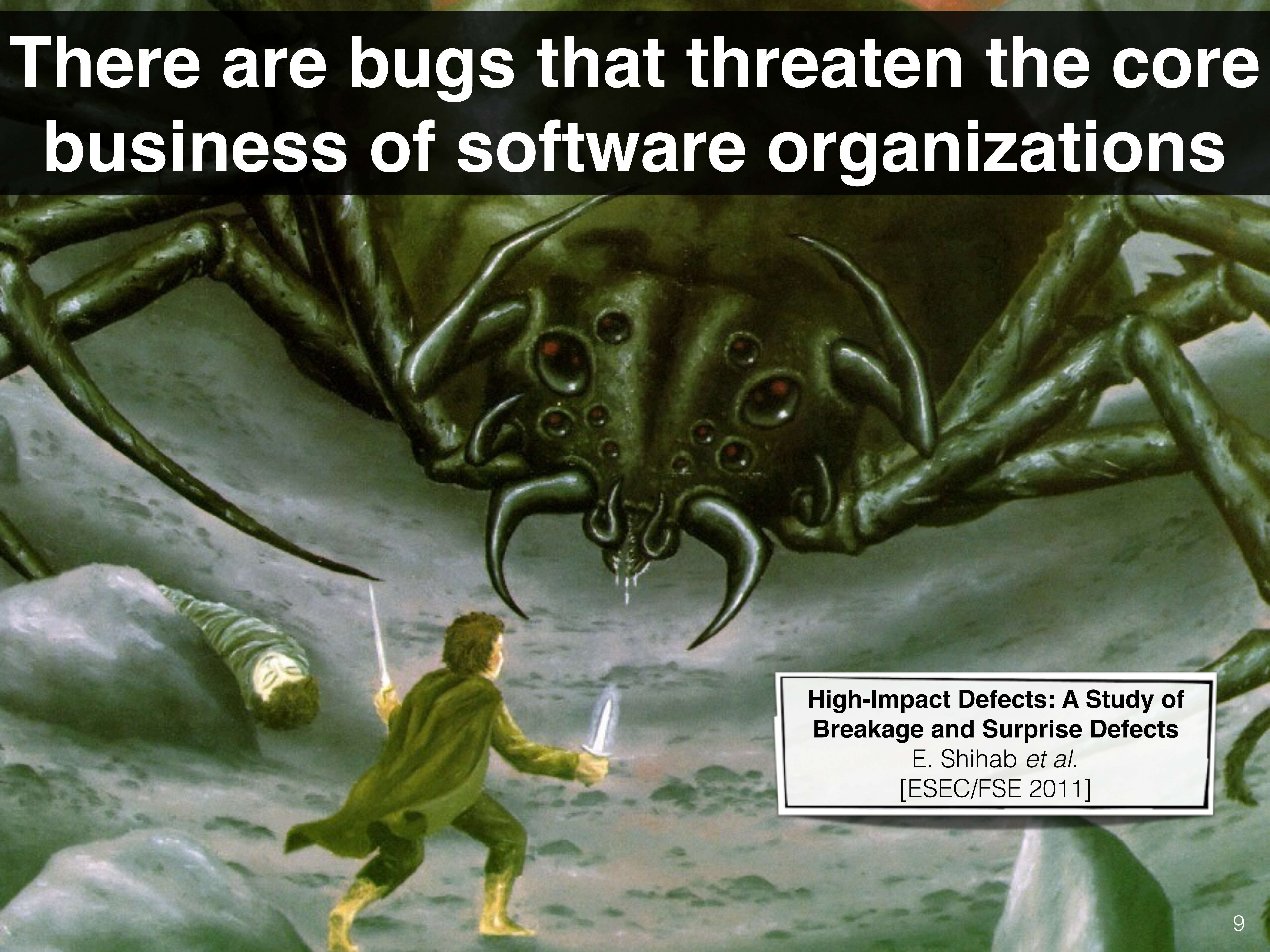
There are bugs that are relatively harmless

Non-Essential Changes in Version Histories
D. Kawrykow and M. P. Robillard
[ICSE 2011]

It’s Not a Bug, It’s a Feature: How Misclassification Impacts Bug Prediction
K. Herzig *et al.*
[ICSE 2013]

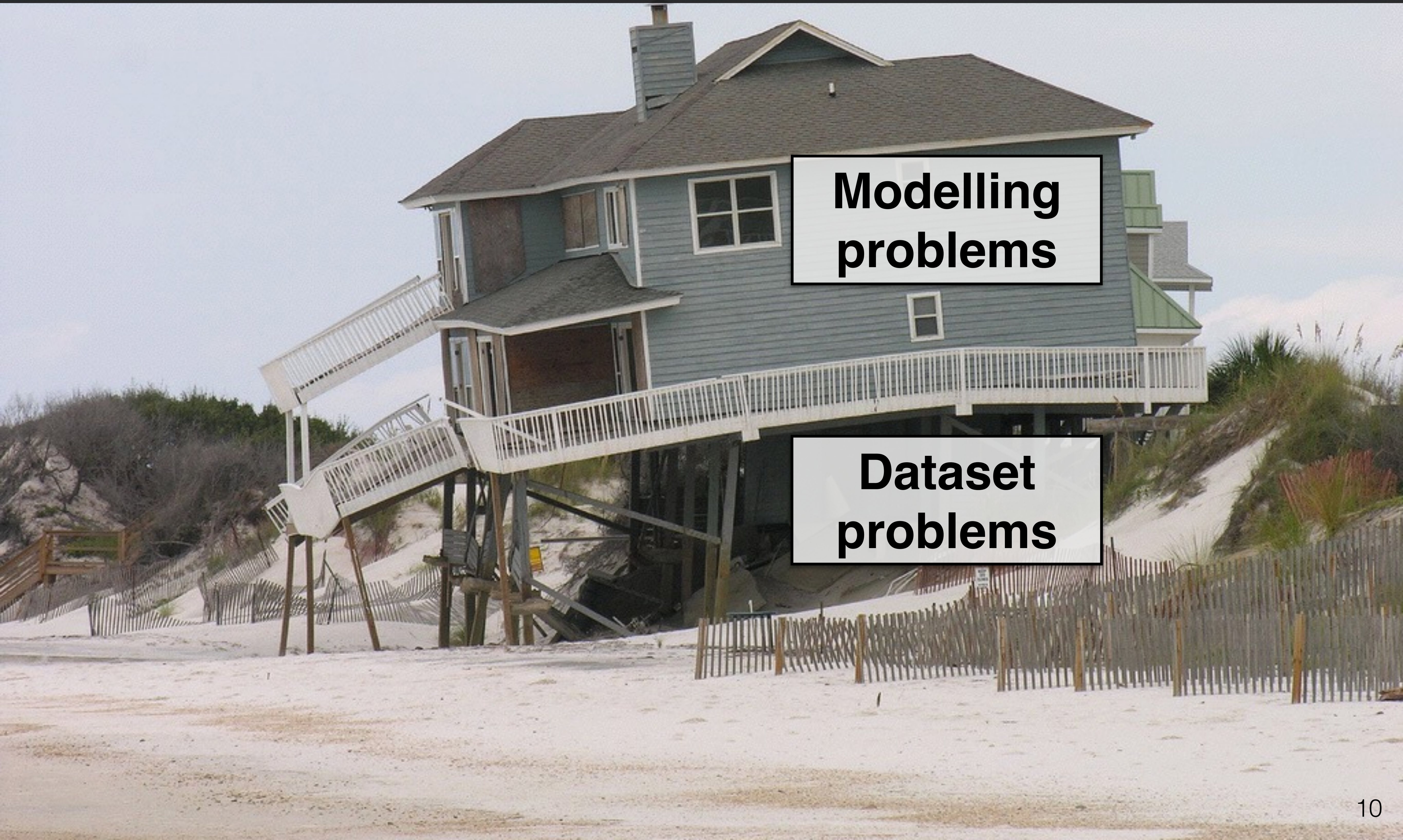
The Impact of Mislabelling on the Performance and Interpretation of Defect Prediction Models
C. Tantithamthavorn
[ICSE 2015]

There are bugs that threaten the core business of software organizations



High-Impact Defects: A Study of Breakage and Surprise Defects
E. Shihab *et al.*
[ESEC/FSE 2011]

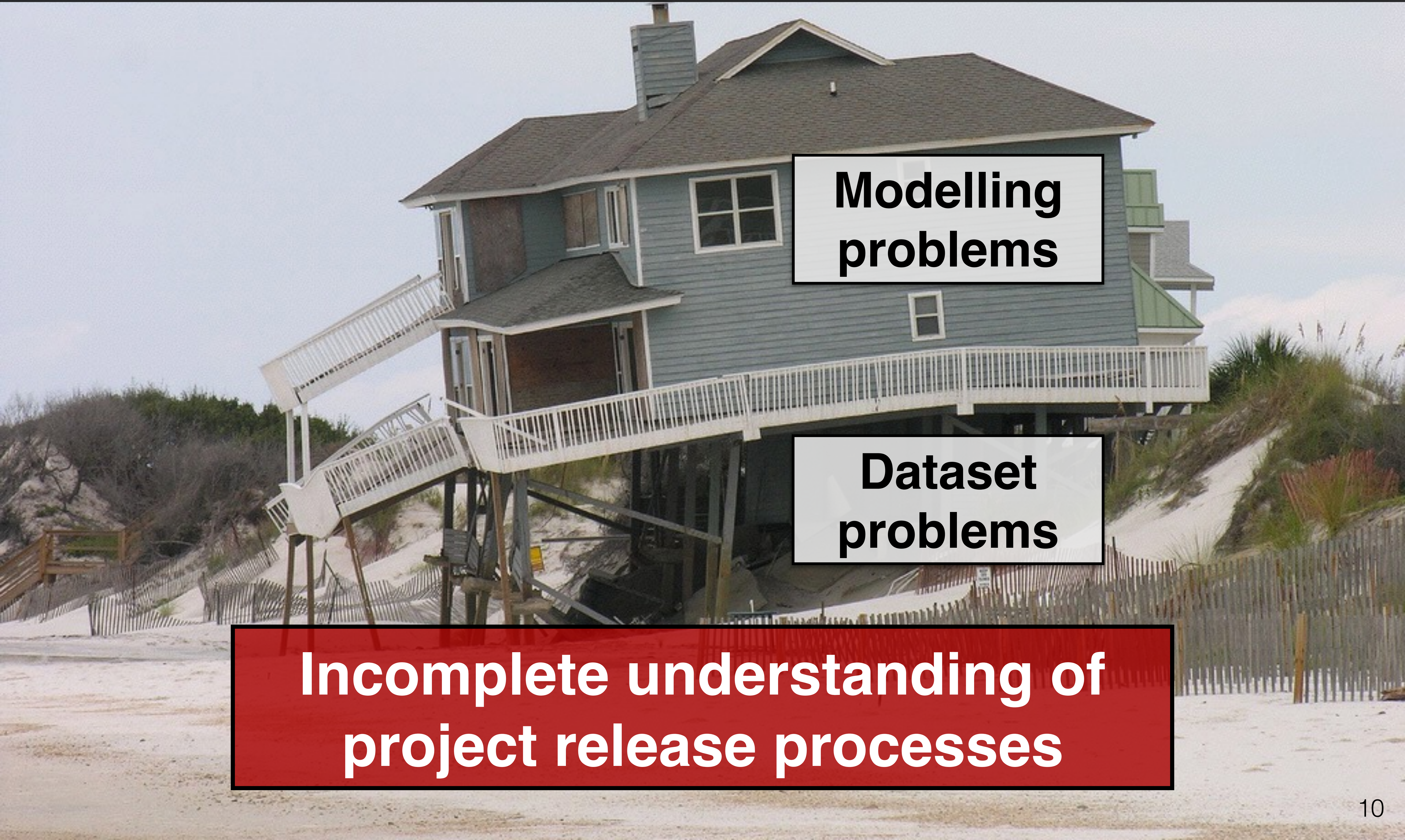
Our models may be trained on unsound data



Modelling problems

Dataset problems

Our models may be trained on unsound data



Modelling problems

Dataset problems

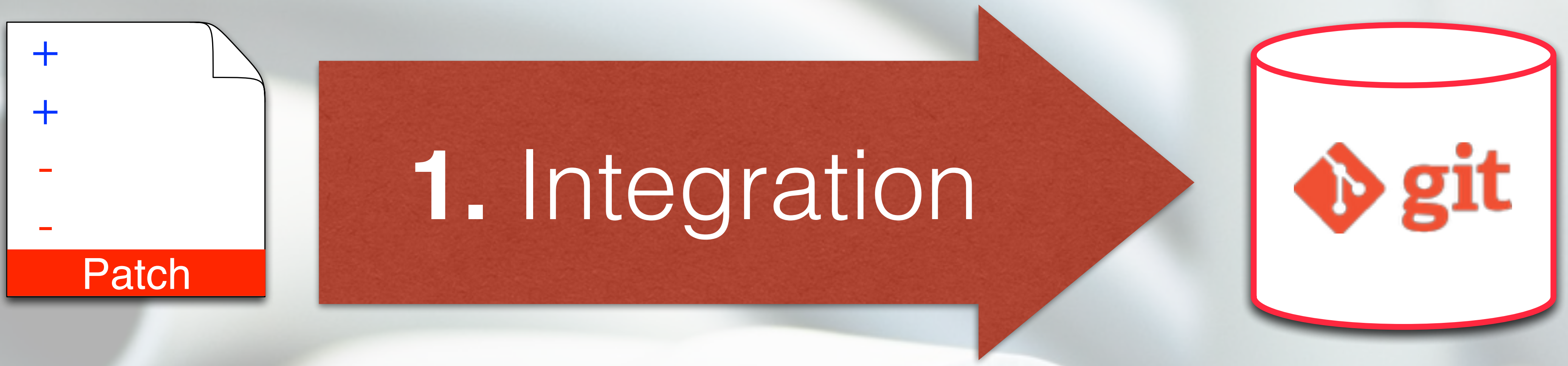
Incomplete understanding of project release processes

Release pipelines:
How organizations deliver new content

Release pipelines: How organizations deliver new content

+
+
-
-
Patch

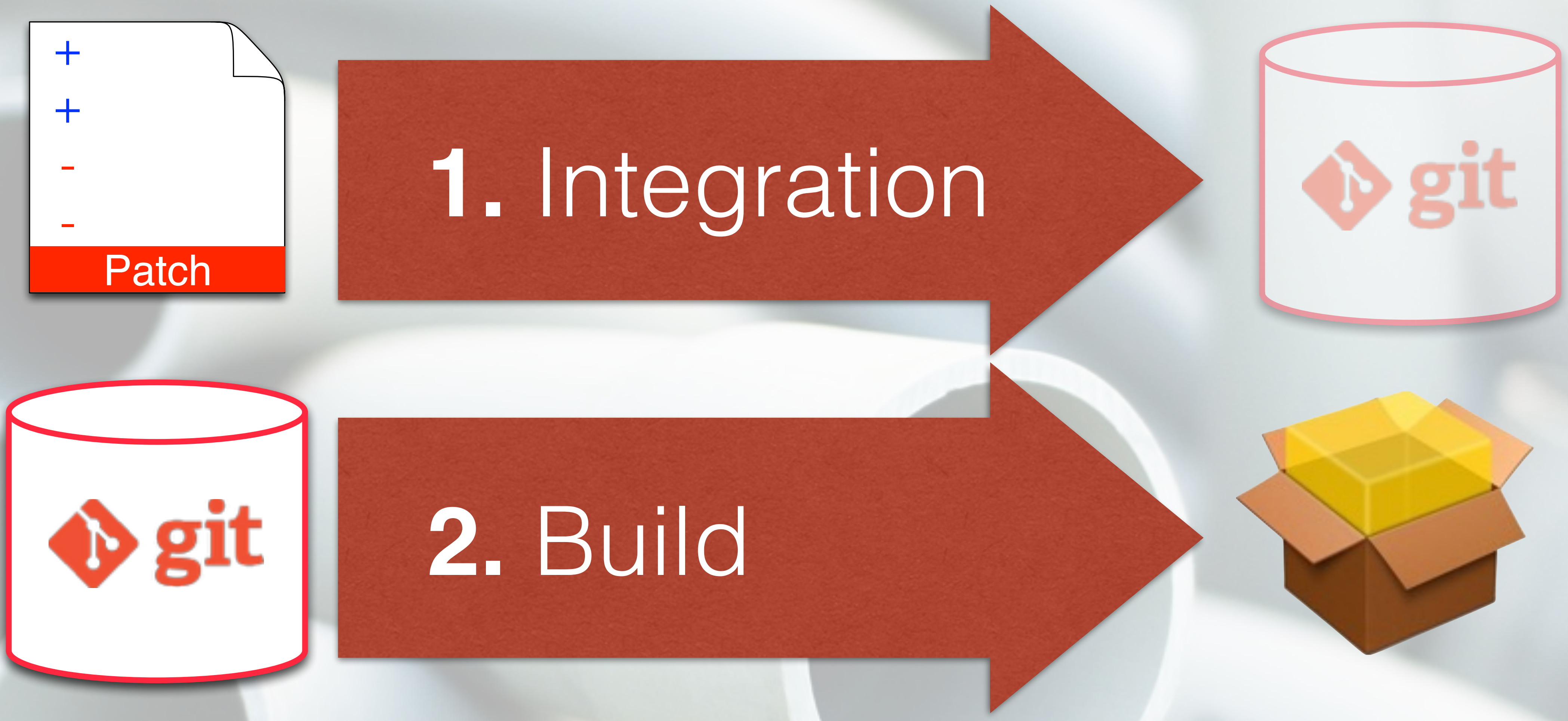
Release pipelines:
How organizations deliver new content



+
+
-
-
Patch

1. Integration

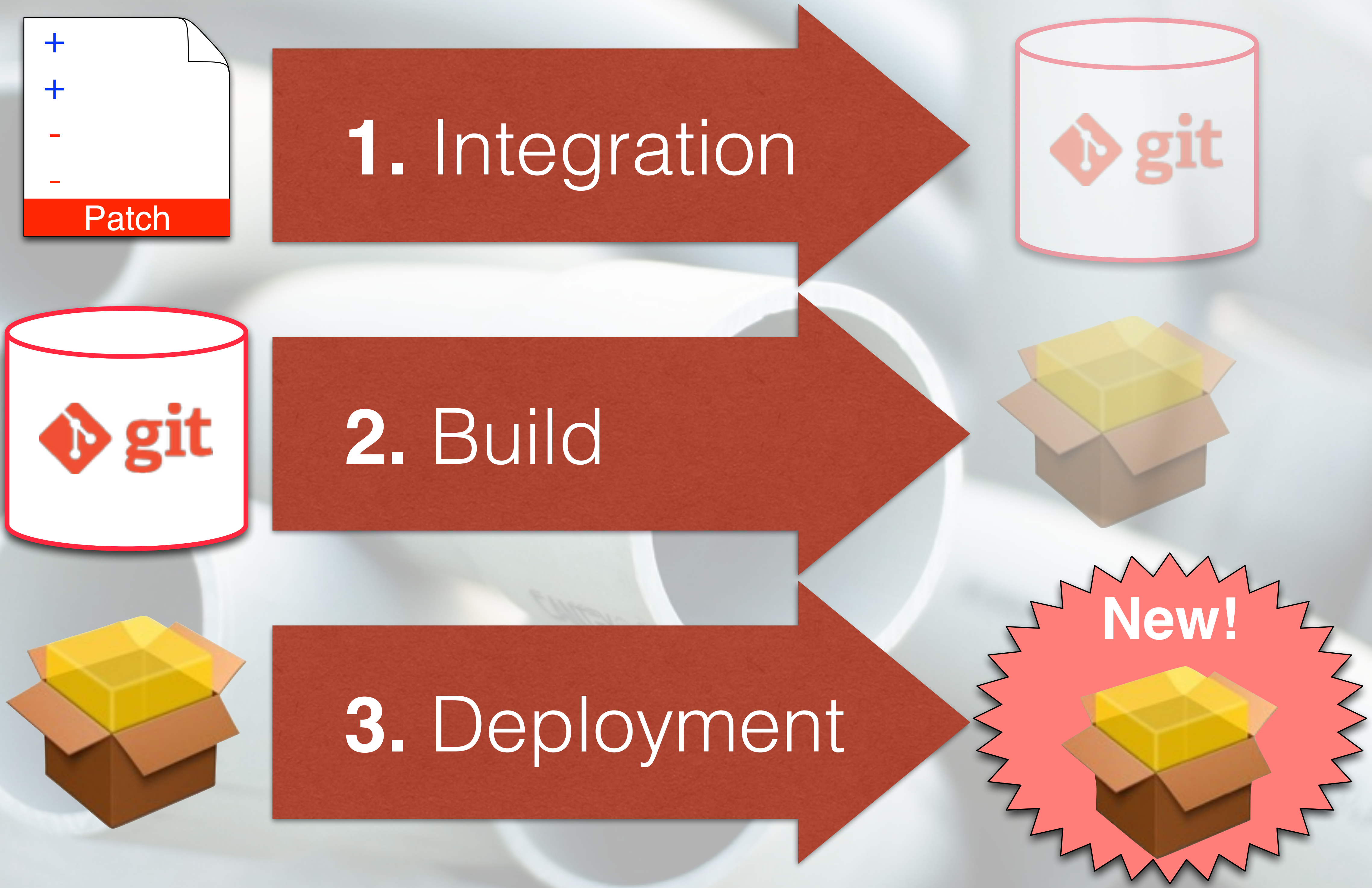
Release pipelines:
How organizations deliver new content



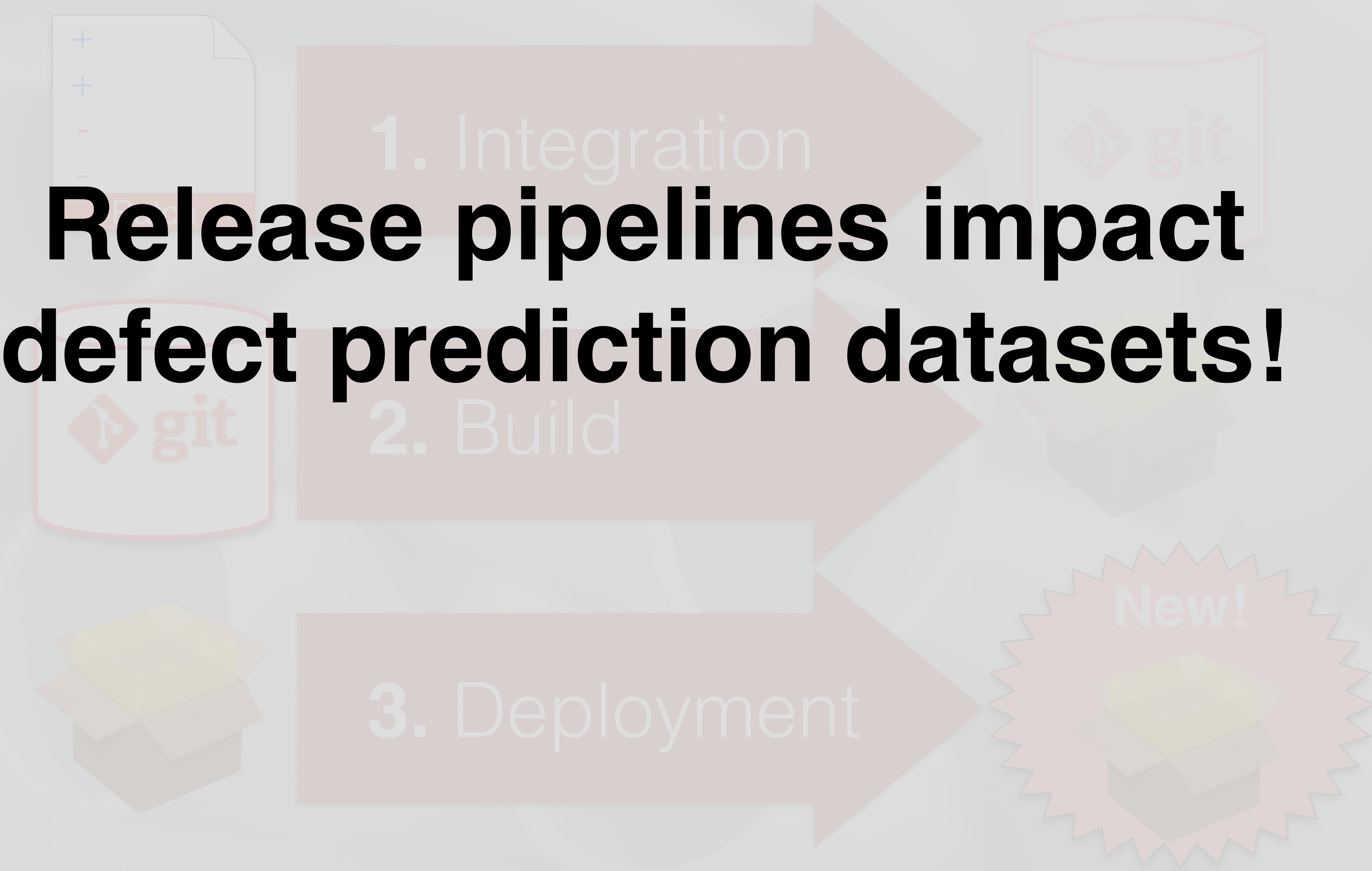
1. Integration

2. Build

Release pipelines: How organizations deliver new content



Release pipelines:
How organizations deliver new content



1. Integration

2. Build

3. Deployment

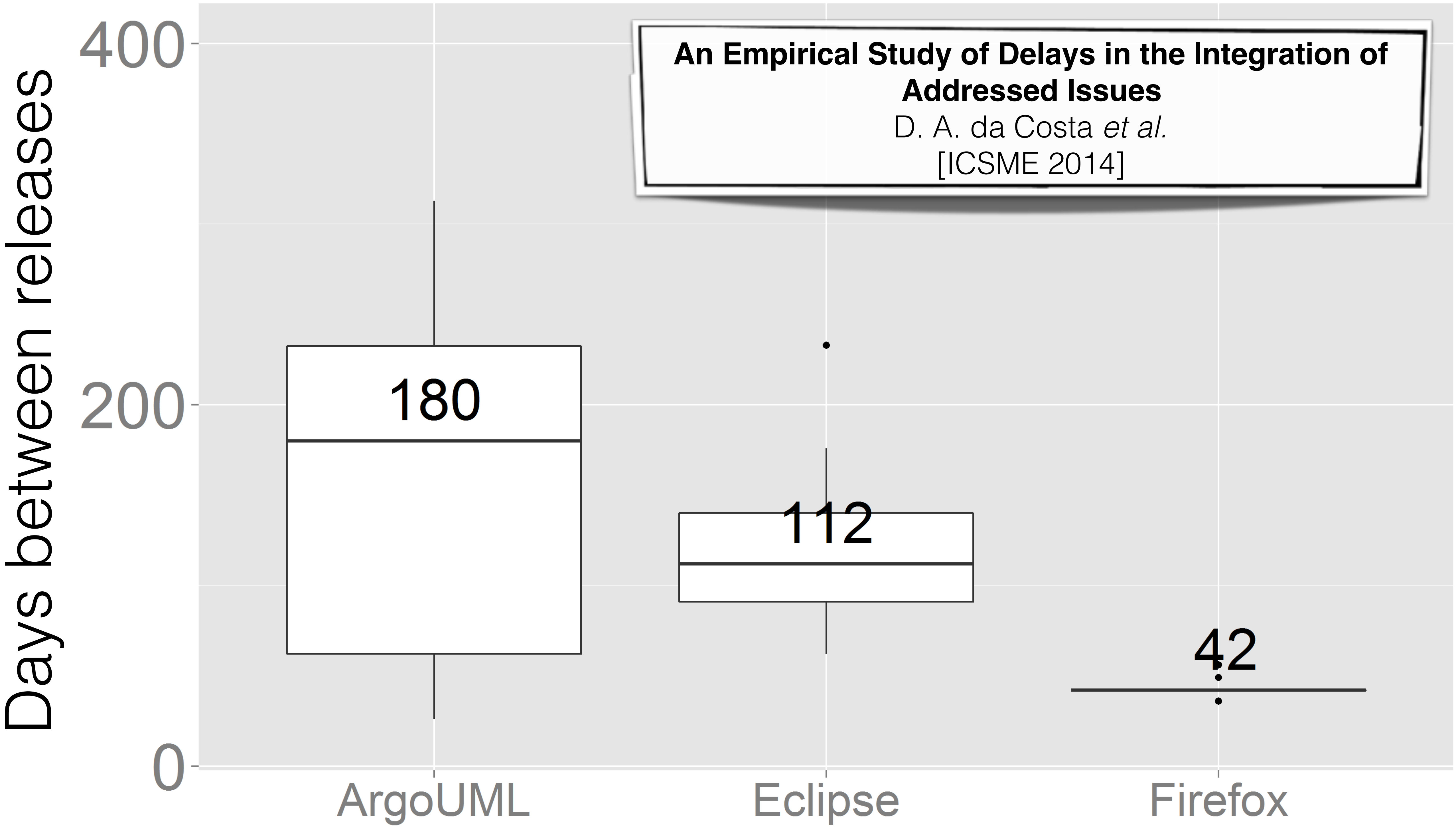
Release pipelines impact defect prediction datasets!

Harmful assumptions about release pipelines that can impact predictive modelling

Harmful assumptions about release pipelines that can impact predictive modelling

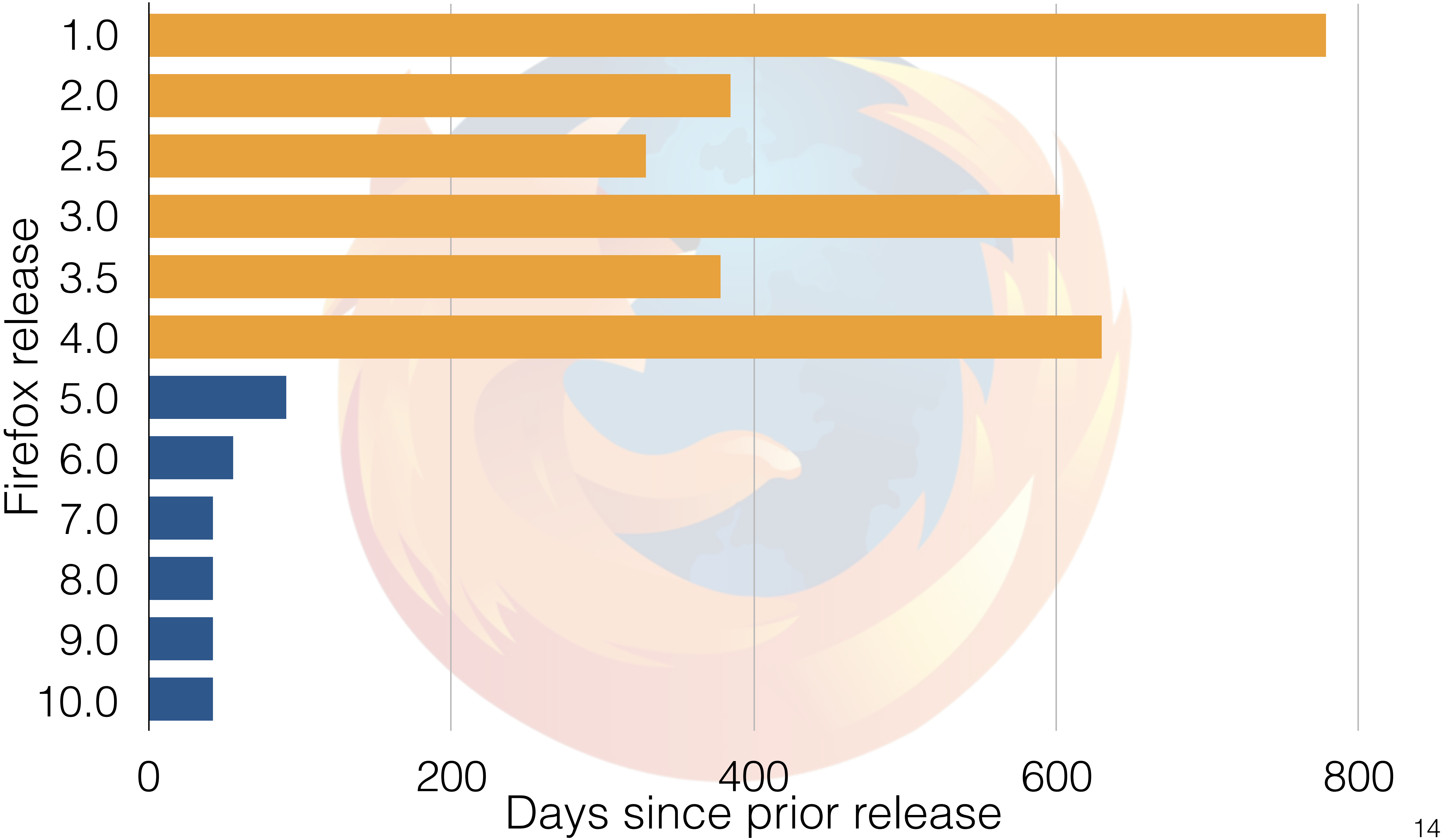
1. All releases are equal

Release cycles vary among popular studied systems



An Empirical Study of Delays in the Integration of Addressed Issues
D. A. da Costa *et al.*
[ICSME 2014]

Release cycles can even vary within systems!



The rapid release cycle of modern software systems

The rapid release cycle of modern software systems

Often release several times in one day!

Harmful assumptions about release pipelines that can impact predictive modelling

1. All releases are equal

Harmful assumptions about release pipelines that can impact predictive modelling

1. All releases are equal
2. All branches are equal

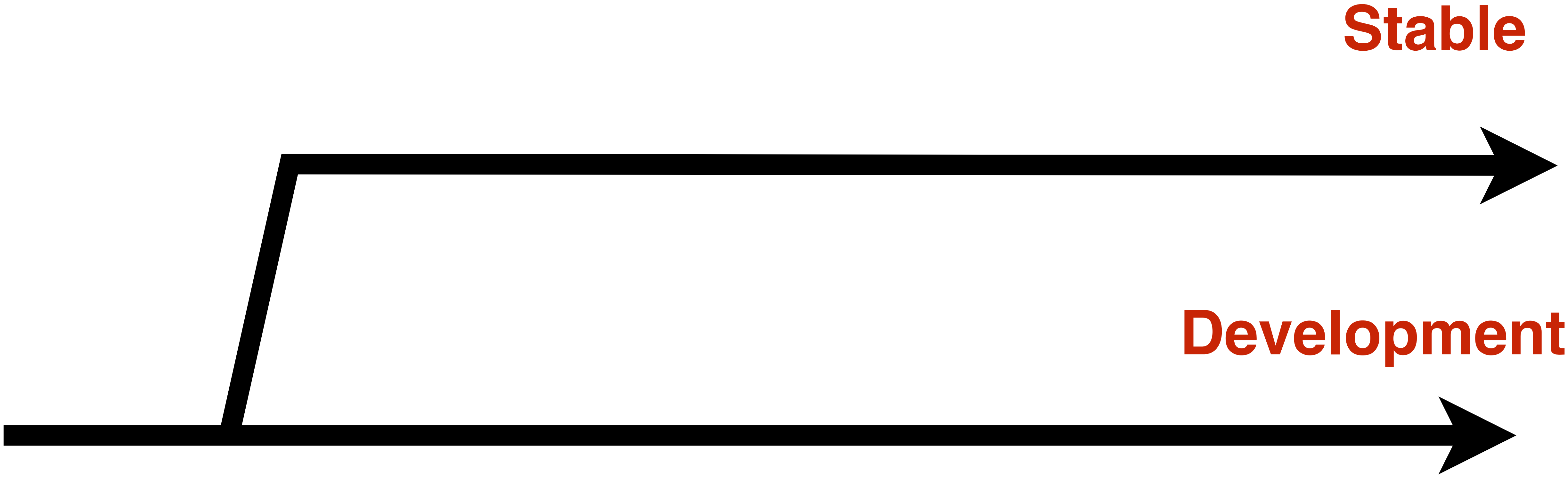
Handling the intricacies of a multi-branch release pipeline

<u>Commit types</u>

★ Feature development

★ Defect repairing

★ Merge



Stable

Development

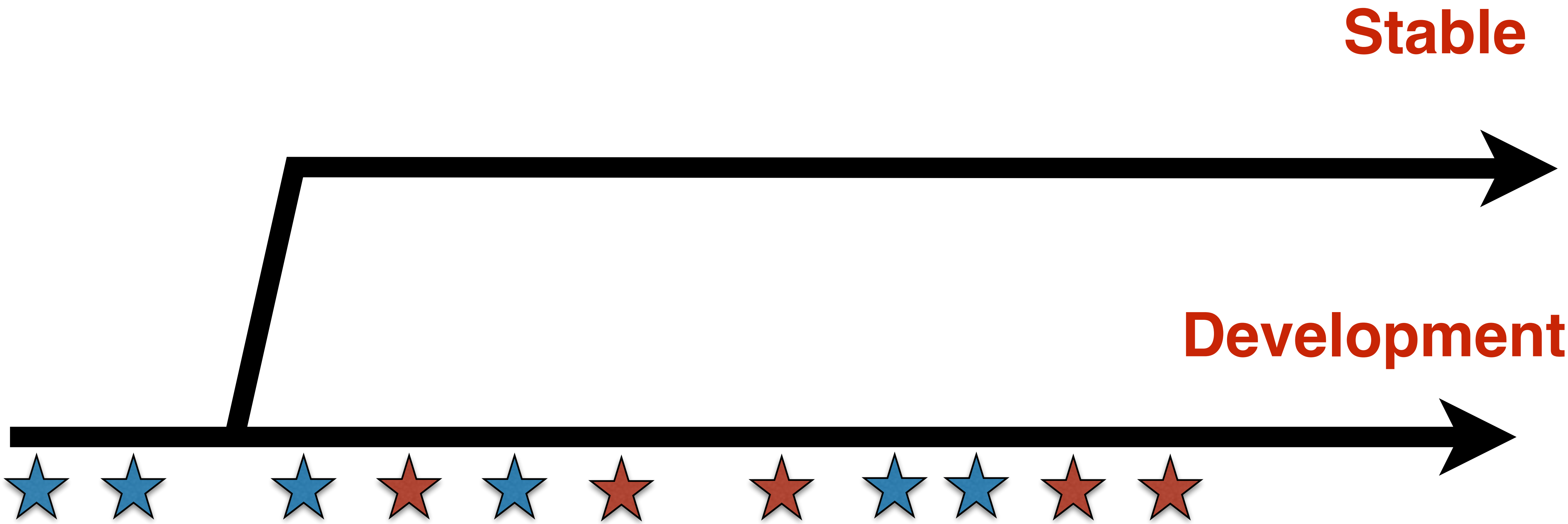
Handling the intricacies of a multi-branch release pipeline

Commit types

★ Feature development

★ Defect repairing

★ Merge



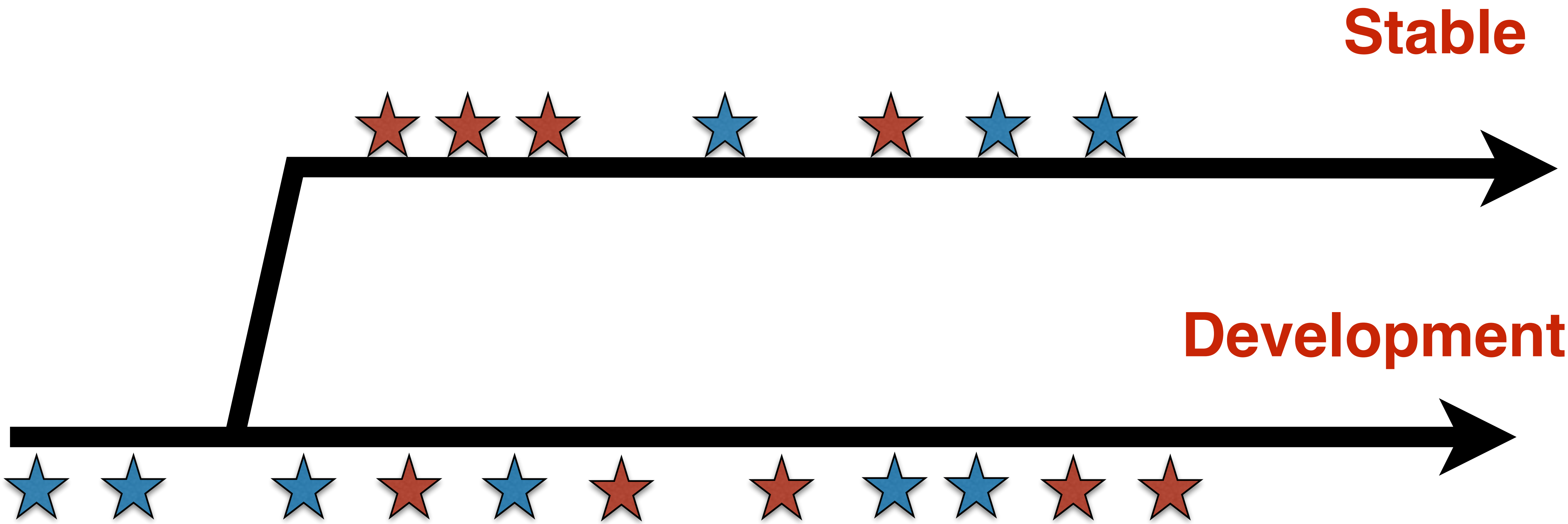
Handling the intricacies of a multi-branch release pipeline

Commit types

★ Feature development

★ Defect repairing

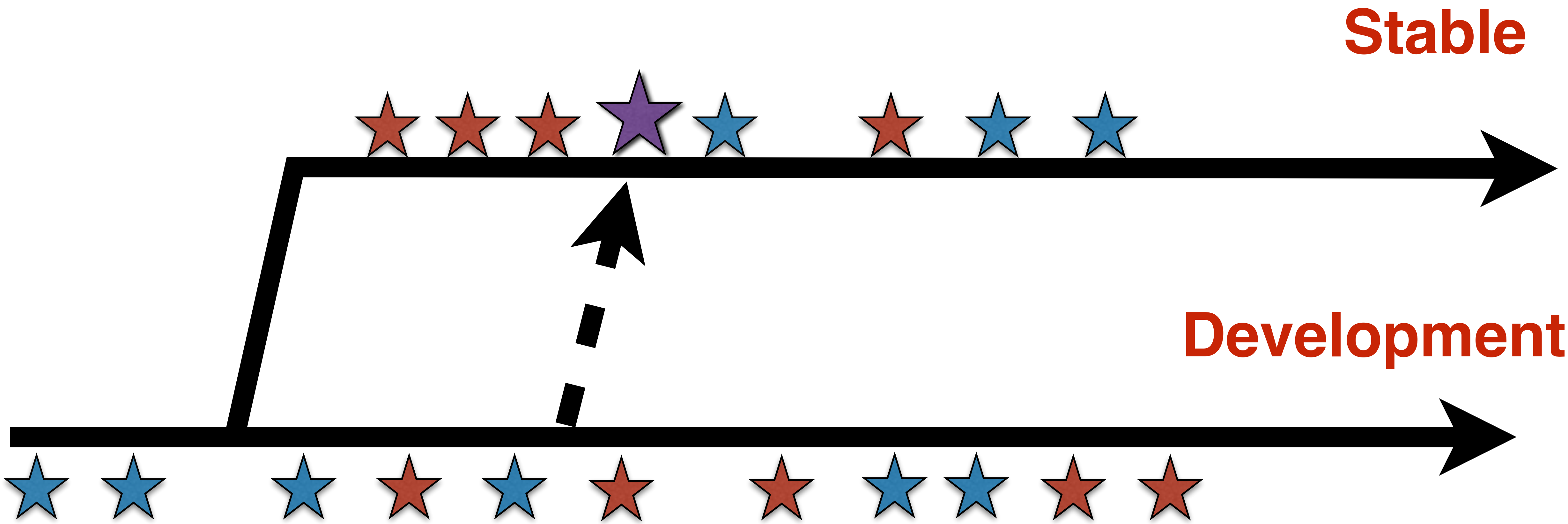
★ Merge



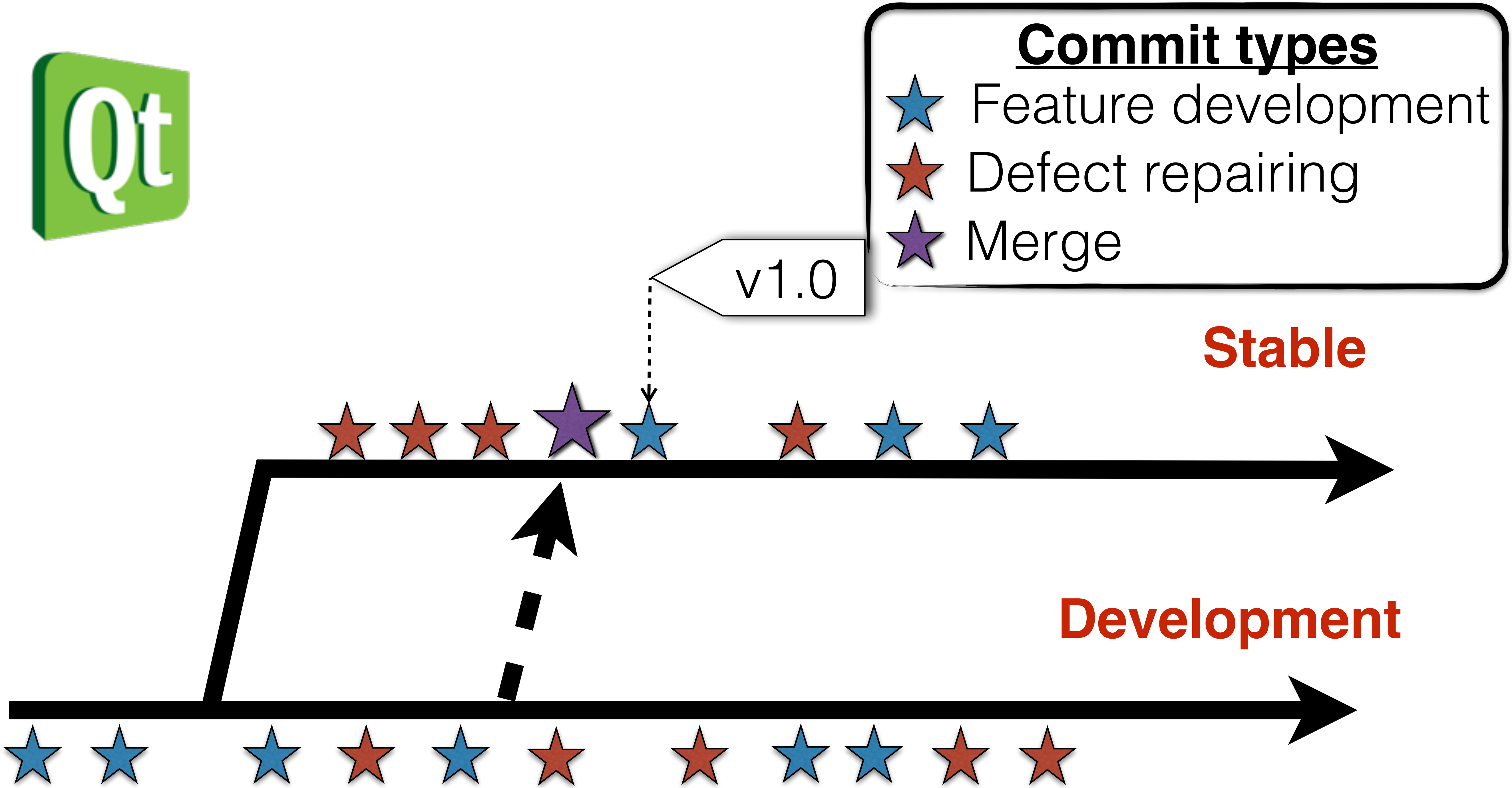
Handling the intricacies of a multi-branch release pipeline

Commit types

- ★ Feature development
- ★ Defect repairing
- ★ Merge



Handling the intricacies of a multi-branch release pipeline



Commit types

★ Feature development

★ Defect repairing

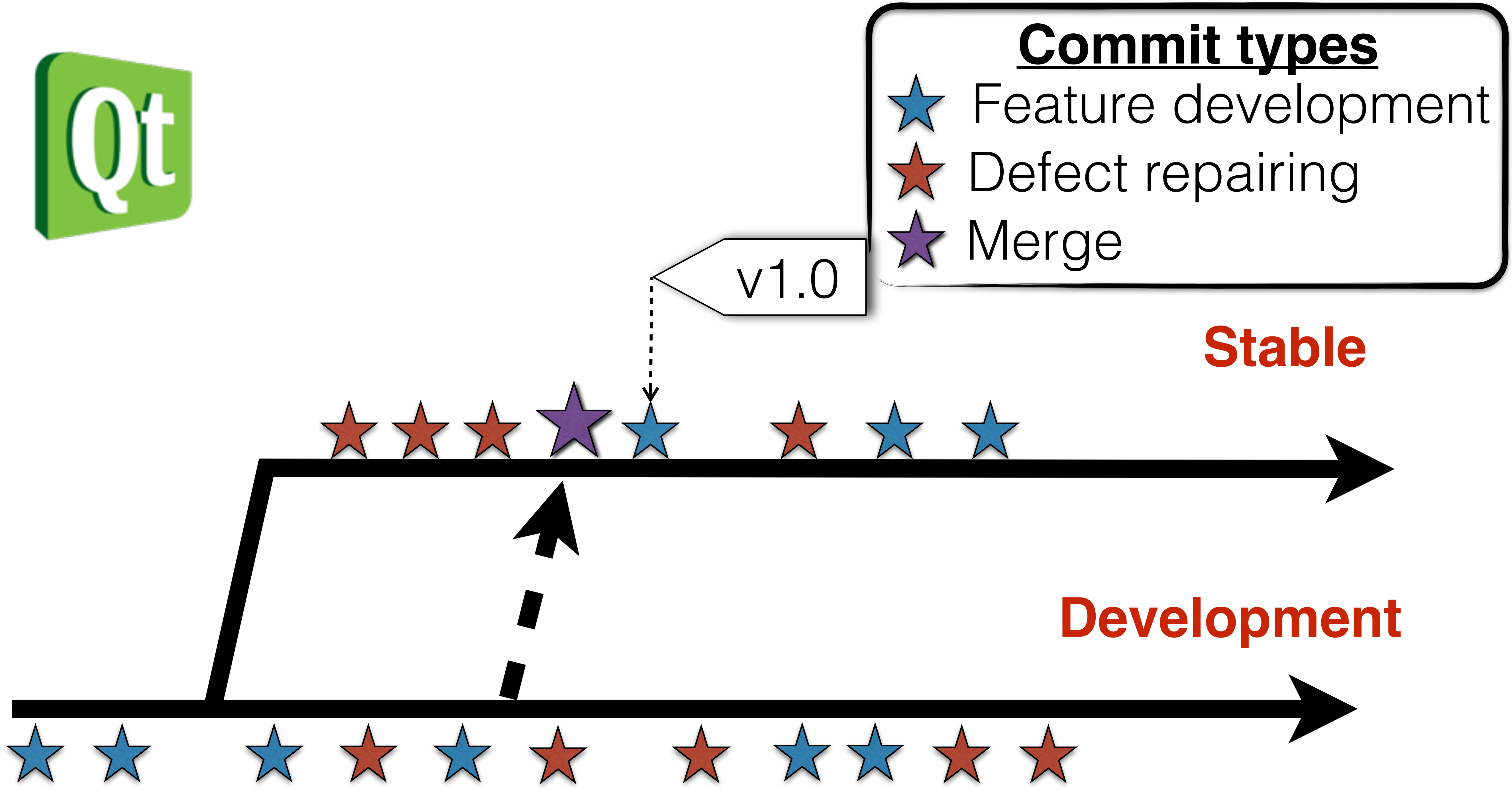
★ Merge

v1.0

Stable

Development

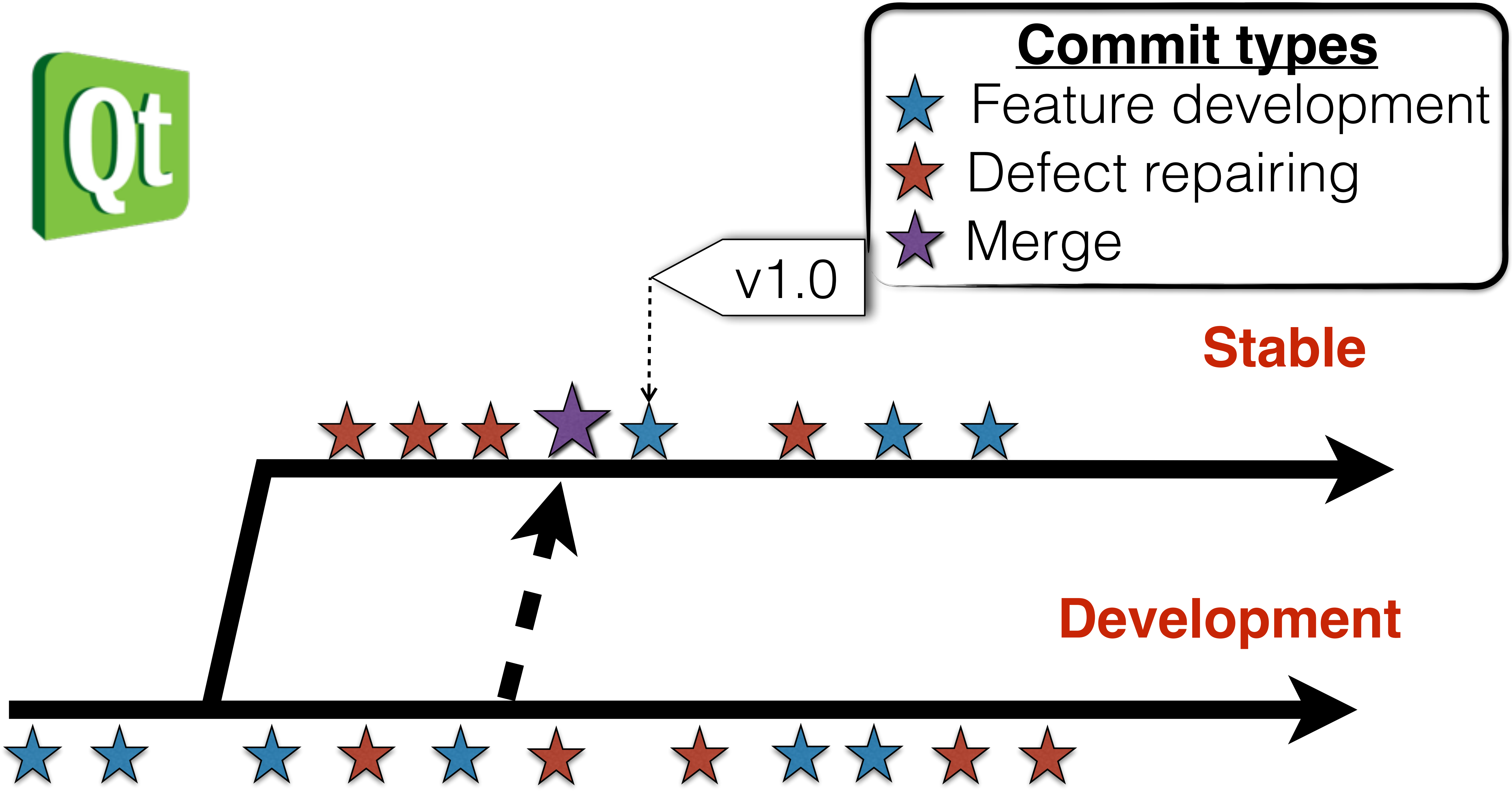
Handling the intricacies of a multi-branch release pipeline



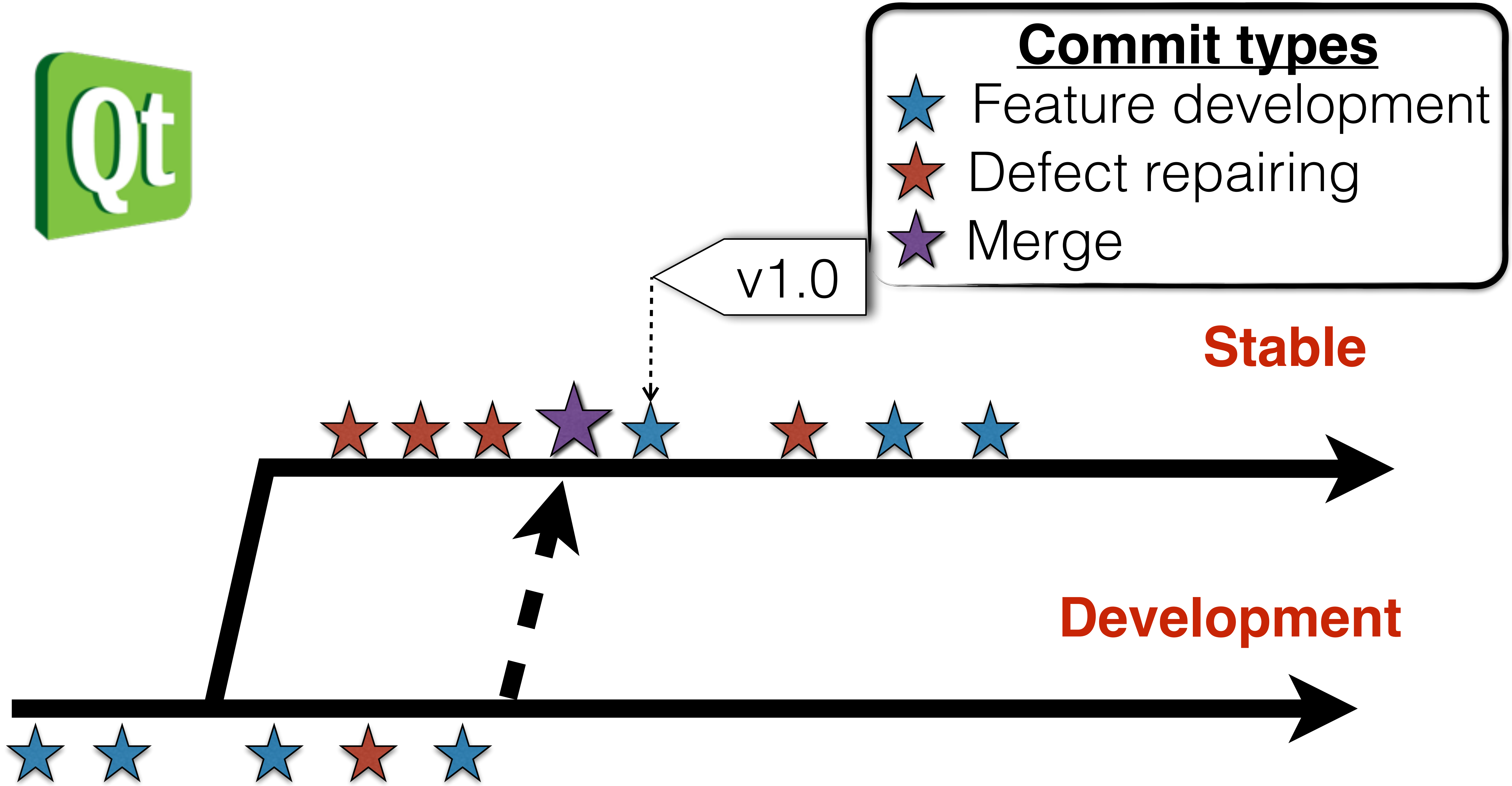
Commit types

- ★ Feature development
- ★ Defect repairing
- ★ Merge

Handling the intricacies of a multi-branch release pipeline



Handling the intricacies of a multi-branch release pipeline



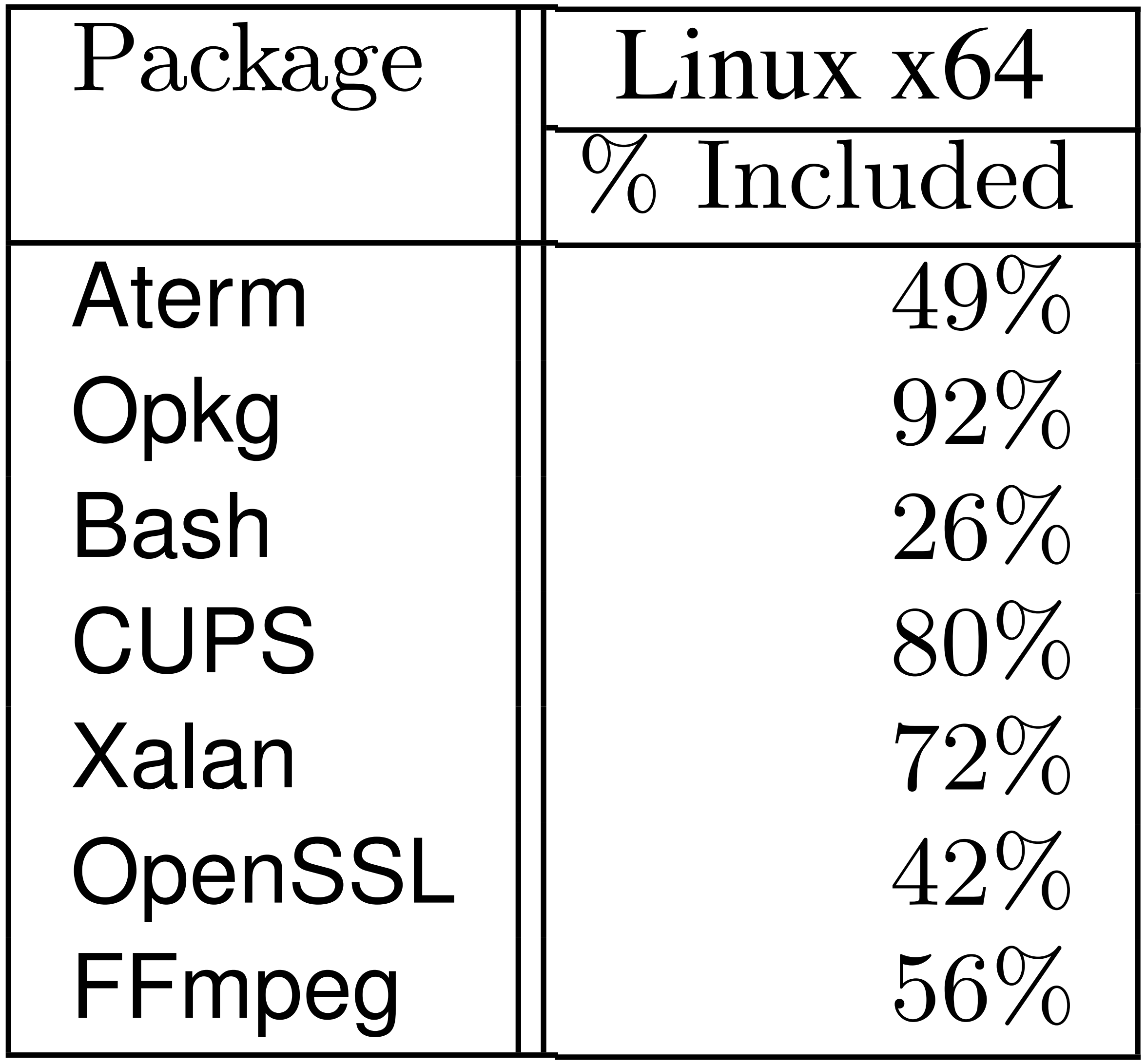
Harmful assumptions about release pipelines that can impact predictive modelling

1. All releases are equal
2. All branches are equal

Harmful assumptions about release pipelines that can impact predictive modelling

1. All releases are equal
2. All branches are equal
3. All files are equal

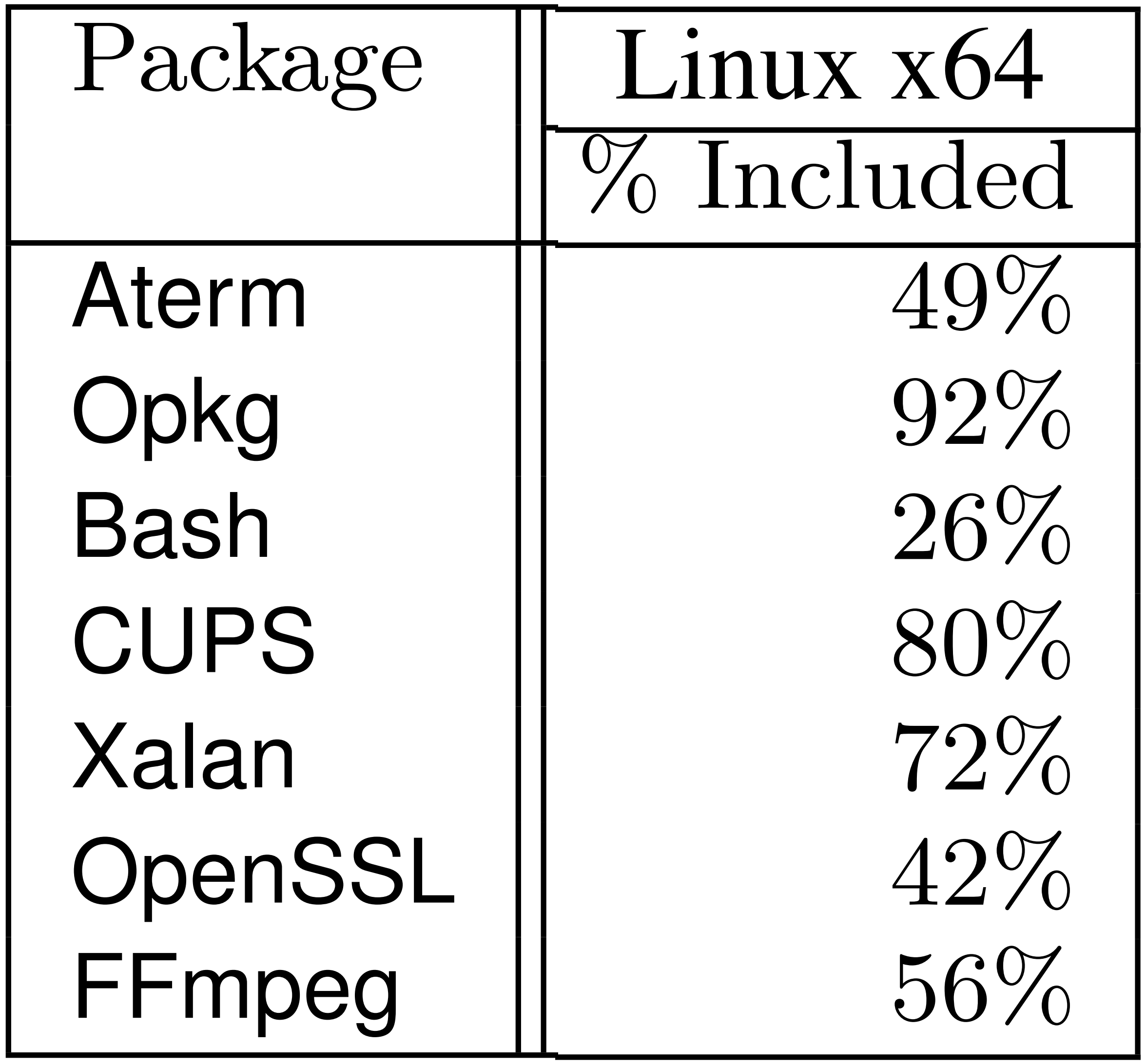
Many files are conditionally included in deliverables



Package	Linux x64
	% Included
Aterm	49%
Opkg	92%
Bash	26%
CUPS	80%
Xalan	72%
OpenSSL	42%
FFmpeg	56%

Tracing Software Build Processes to Uncover License Compliance Inconsistencies
S. van der Berg *et al.*
[ASE 2014]

Many files are conditionally included in deliverables

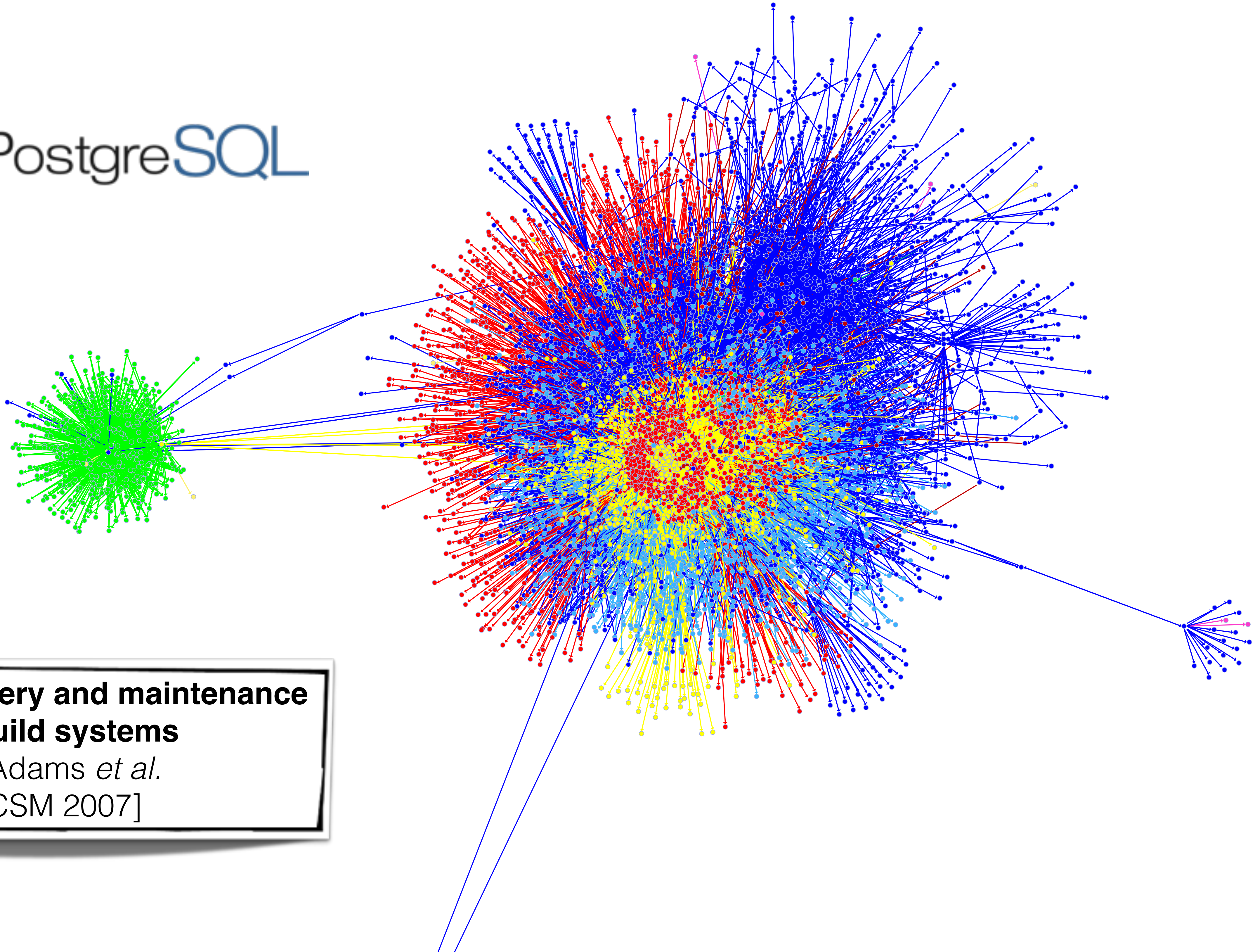


Package	Linux x64 % Included
Aterm	49%
Opkg	92%
Bash	26%
CUPS	80%
Xalan	72%
OpenSSL	42%
FFmpeg	56%

Tracing Software Build Processes to Uncover License Compliance Inconsistencies
S. van der Berg *et al.*
[ASE 2014]

Fixes in these files may have a smaller impact (if any) on customers

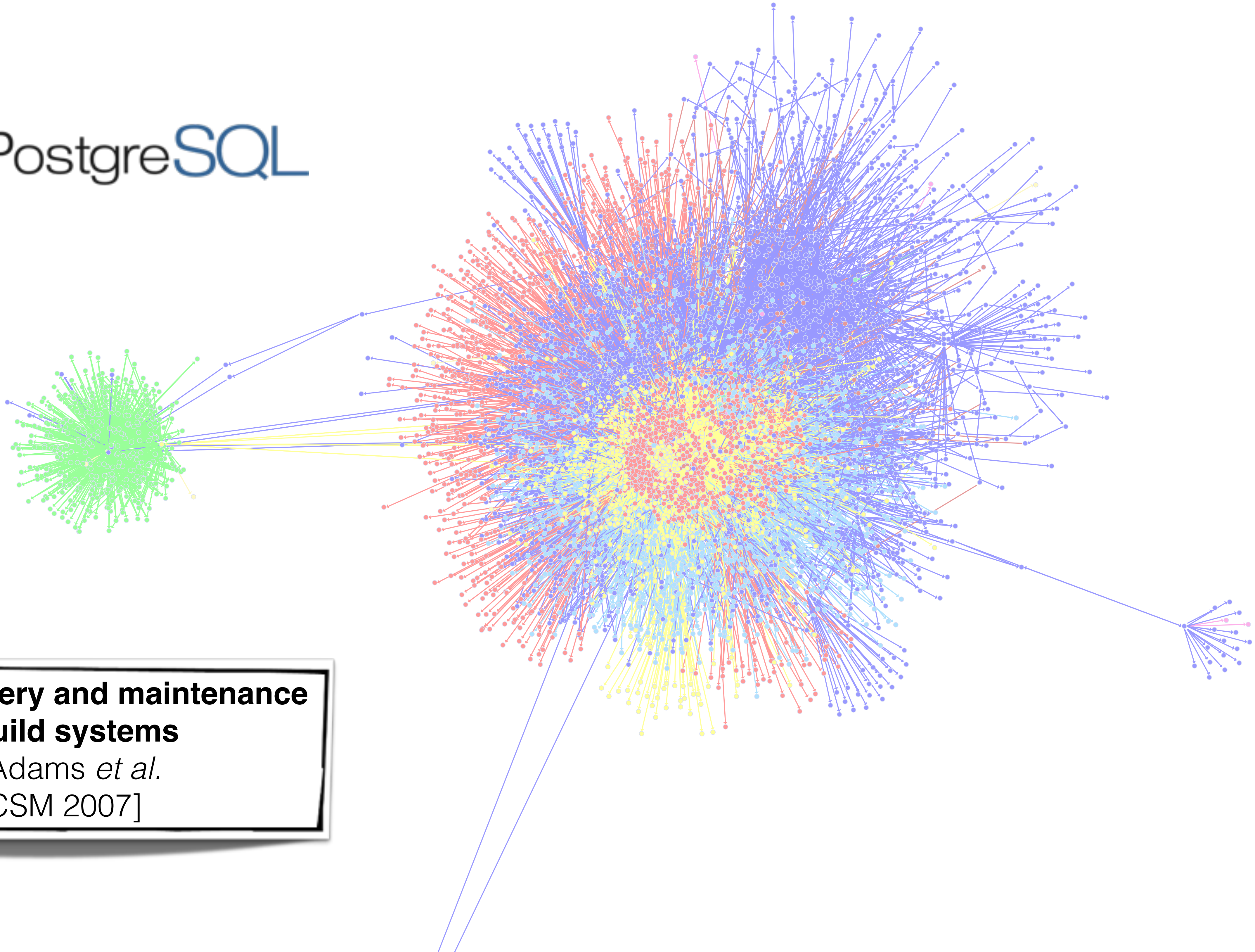
Understanding conditionally included files using the build system



Design recovery and maintenance of build systems
B. Adams *et al.*
[ICSM 2007]

Understanding conditionally included files using the build system

PostgreSQL



Design recovery and maintenance of build systems
B. Adams *et al.*
[ICSM 2007]

Harmful assumptions about release pipelines that can impact predictive modelling

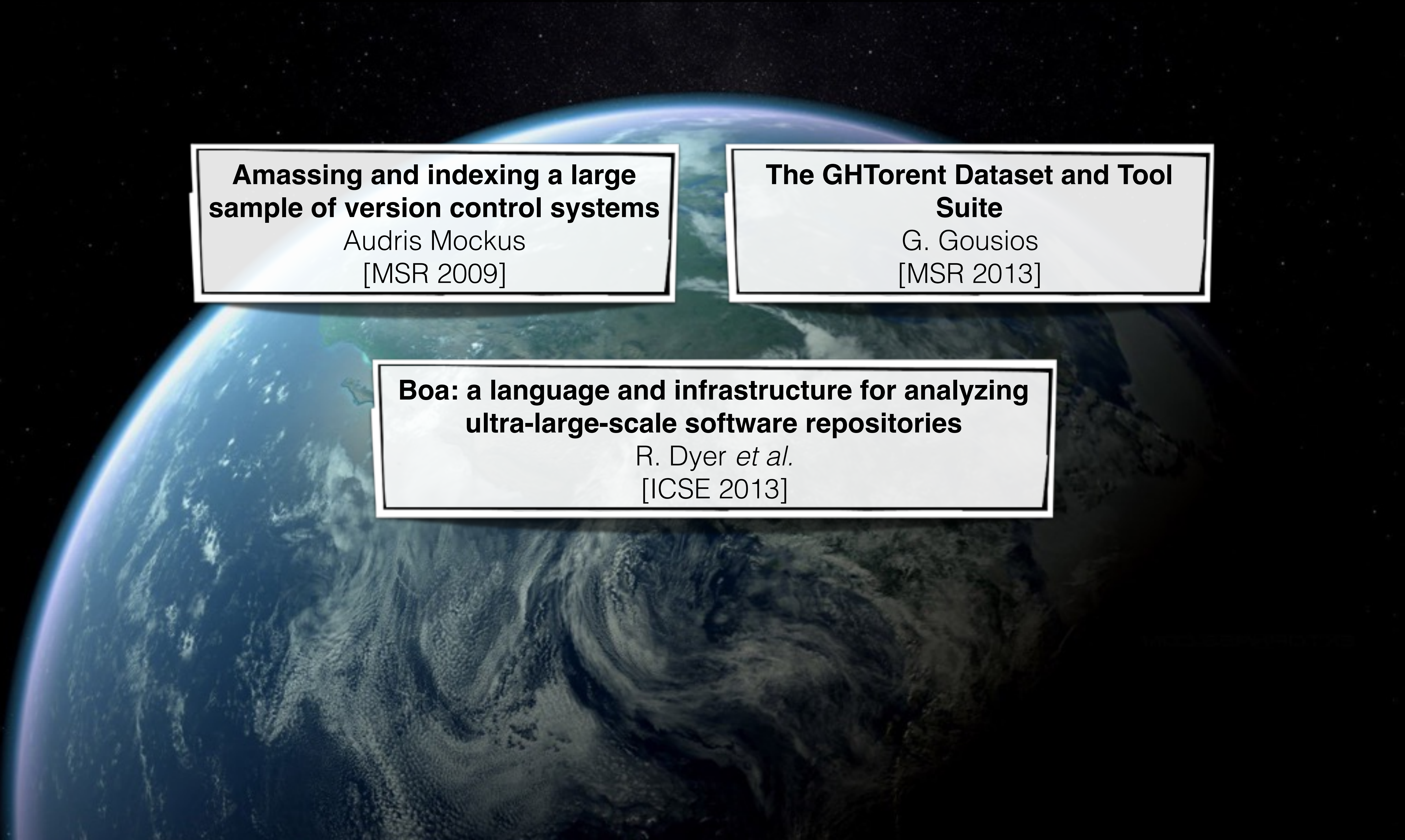
1. All releases are equal,
2. All branches are equal,
3. All files are equal

Harmful assumptions about release pipelines that can impact predictive modelling

1. All releases are equal,
2. All branches are equal,
3. All files are equal

but some are more equal than others

My nightmare

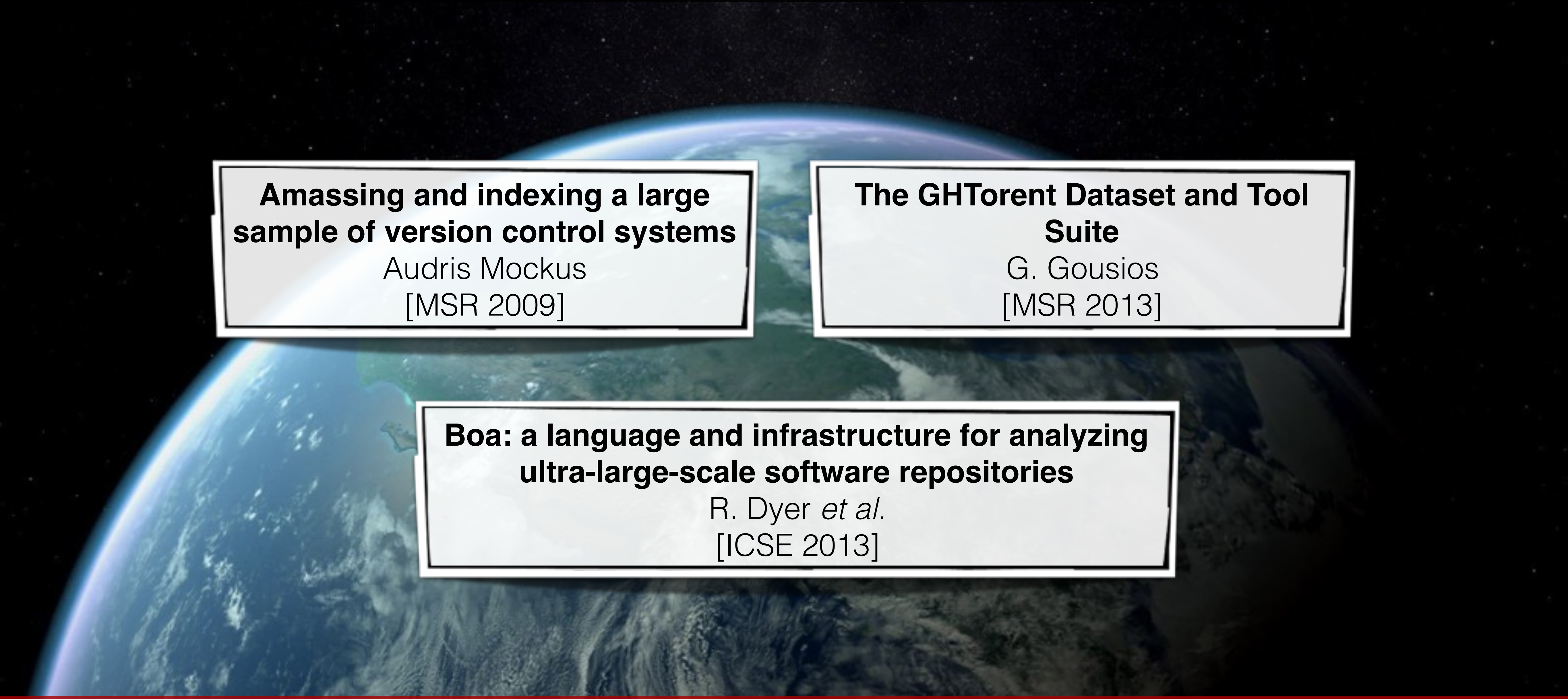


Amassing and indexing a large sample of version control systems
Audris Mockus
[MSR 2009]

The GHTorent Dataset and Tool Suite
G. Gousios
[MSR 2013]

Boa: a language and infrastructure for analyzing ultra-large-scale software repositories
R. Dyer *et al.*
[ICSE 2013]

My nightmare



Amassing and indexing a large sample of version control systems
Audris Mockus
[MSR 2009]

The GHTorent Dataset and Tool Suite
G. Gousios
[MSR 2013]

Boa: a language and infrastructure for analyzing ultra-large-scale software repositories
R. Dyer *et al.*
[ICSE 2013]

We collect all of the data in the world, but it’s meaningless without context!

Integration

Risk assessment

Integration

Risk assessment

Build

Build optimization

Integration

Risk assessment

Build

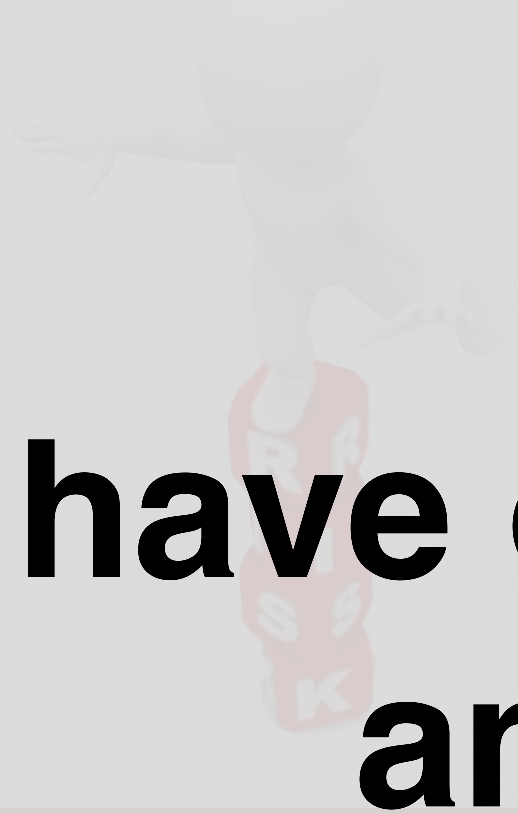
Build optimization

Deployment

Smarter deployment tooling

I have openings for Master's and PhD students!

Integration



Risk assessment

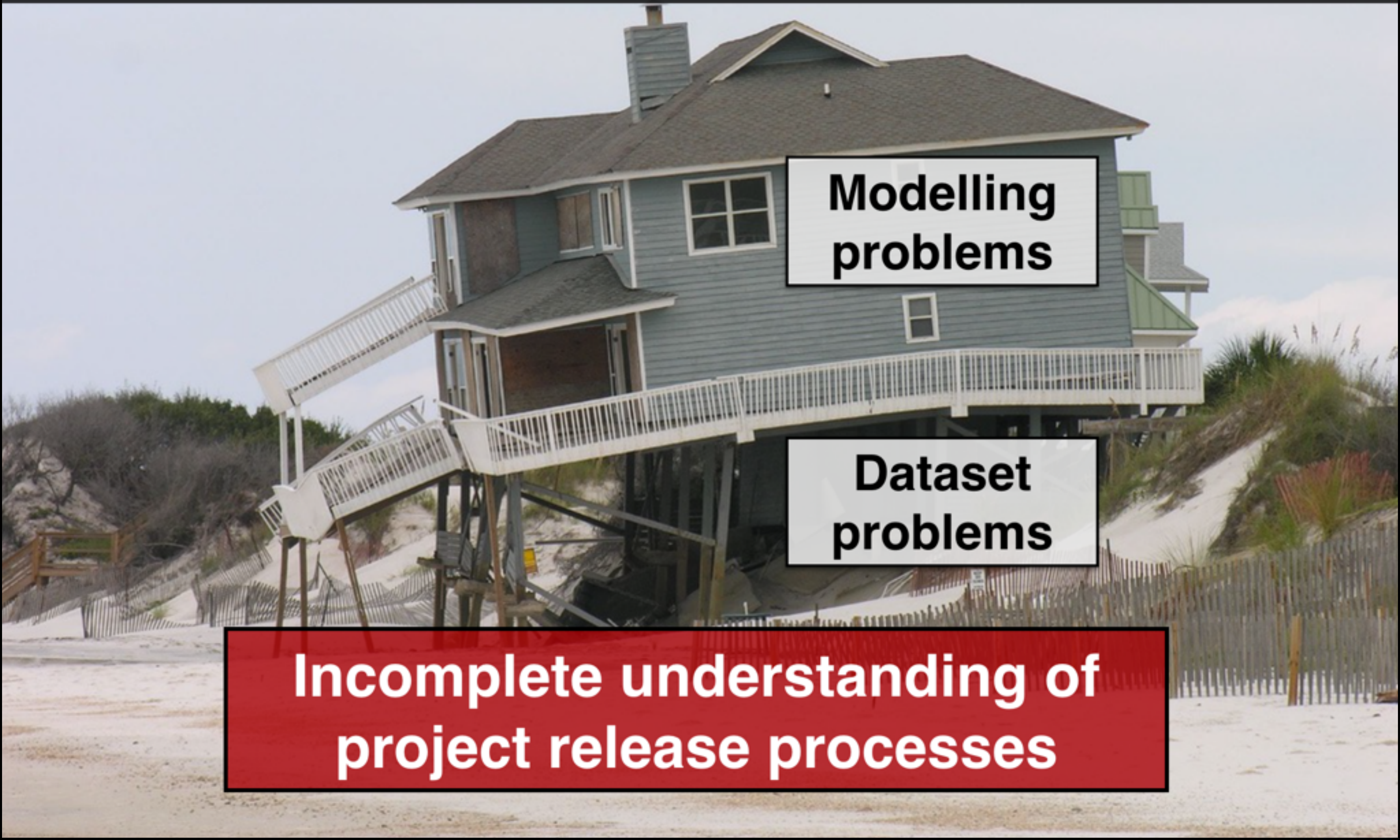
Build

Build optimization

Deployment

Smarter deployment tooling

Our models may be trained on unsound data



Modelling problems

Dataset problems

Incomplete understanding of project release processes

Harmful assumptions about release pipelines that can impact predictive modelling

1. All releases are equal,

2. All branches are equal,

3. All files are equal

but some are more equal than others

Integration

Risk assessment

Build

Build optimization

Deployment

Smarter deployment tooling

Building on an unsound foundation: How release pipelines can impact our predictive models

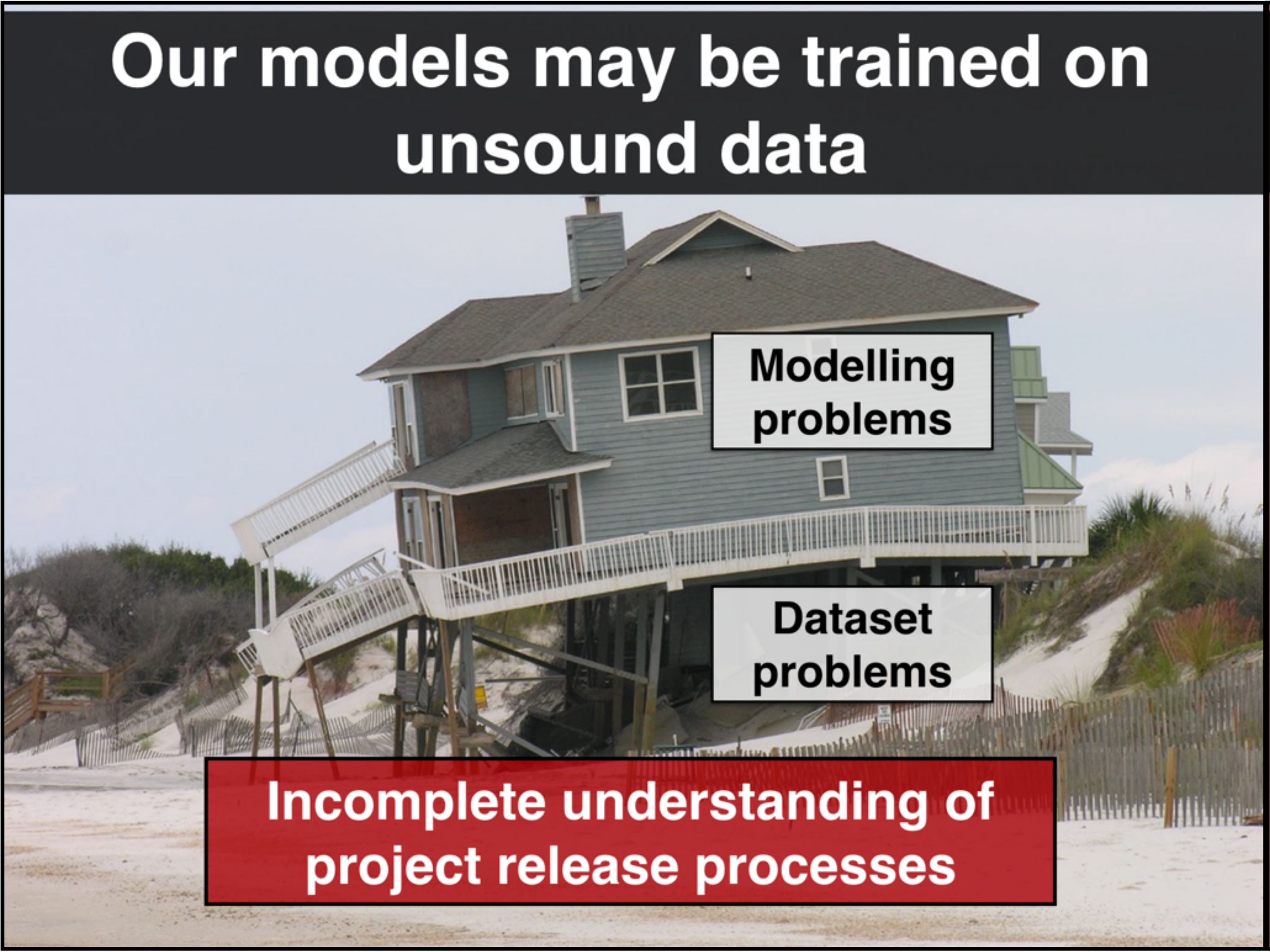
Bram
Adams

POLYTECHNIQUE
MONTRÉAL

Shane
McIntosh

McGill

shanemcintosh@acm.org
@shane_mcintosh
shanemcintosh.org

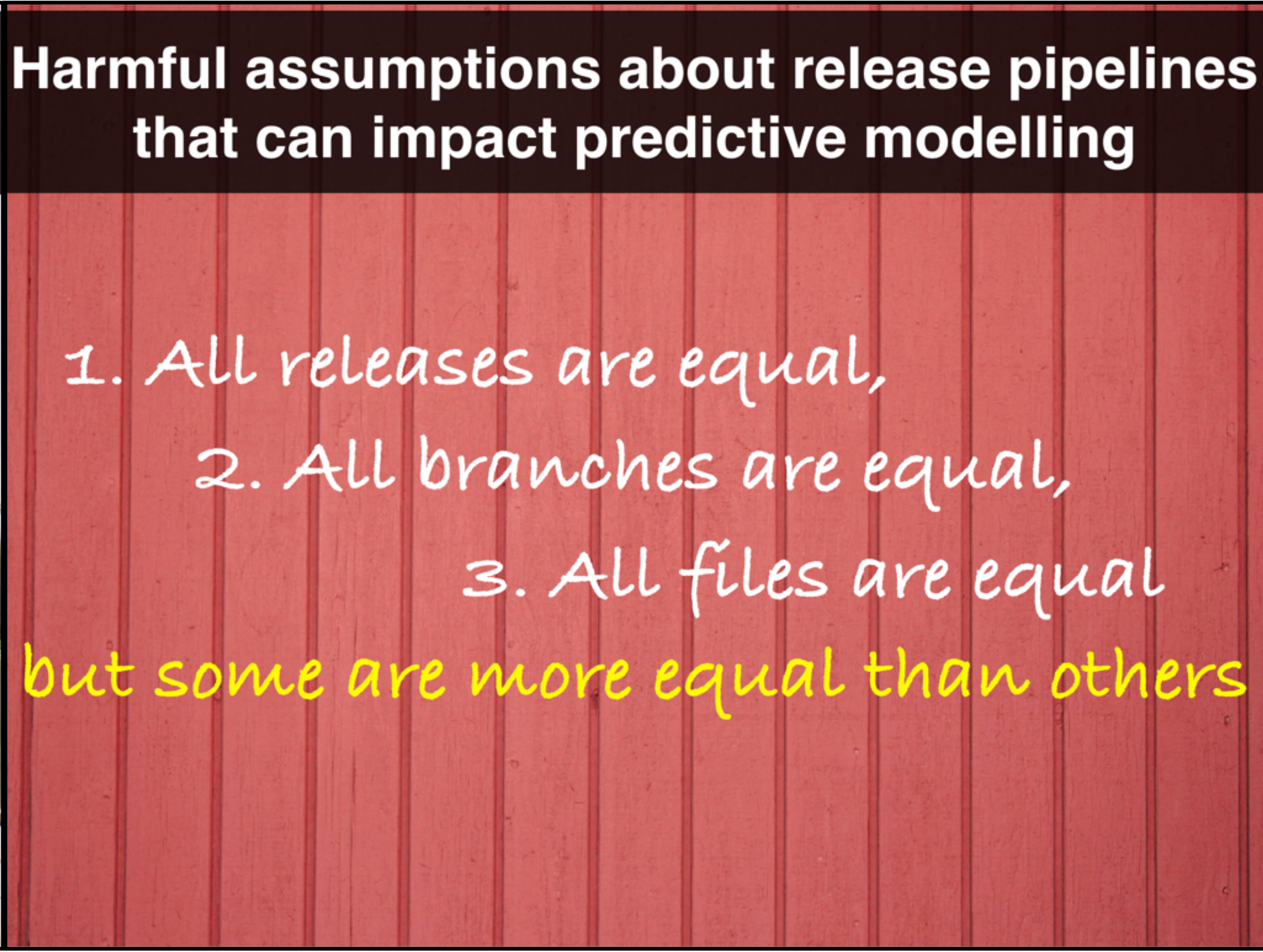


Our models may be trained on unsound data

Modelling problems

Dataset problems

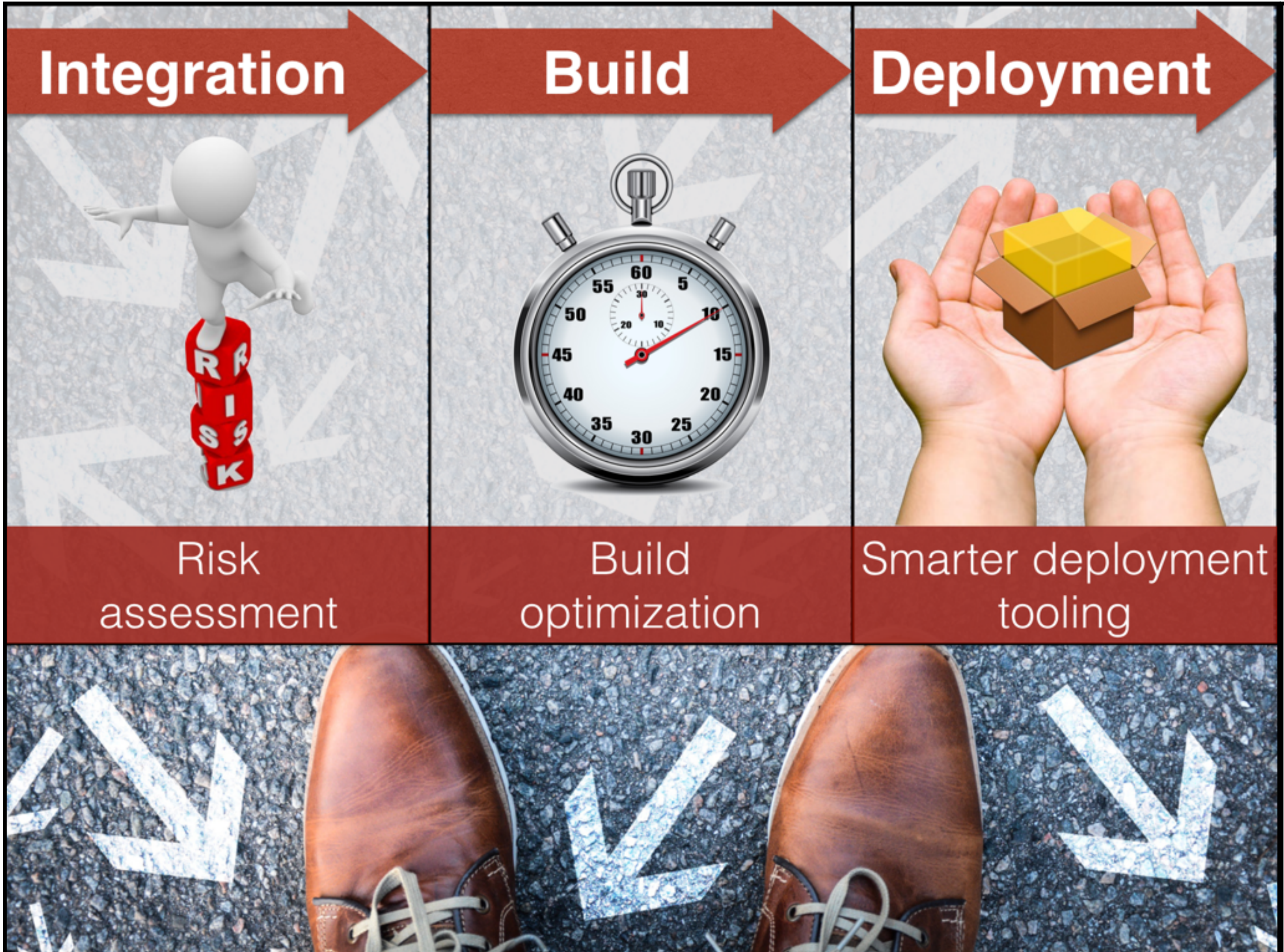
Incomplete understanding of project release processes



Harmful assumptions about release pipelines that can impact predictive modelling

1. All releases are equal,
2. All branches are equal,
3. All files are equal

but some are more equal than others

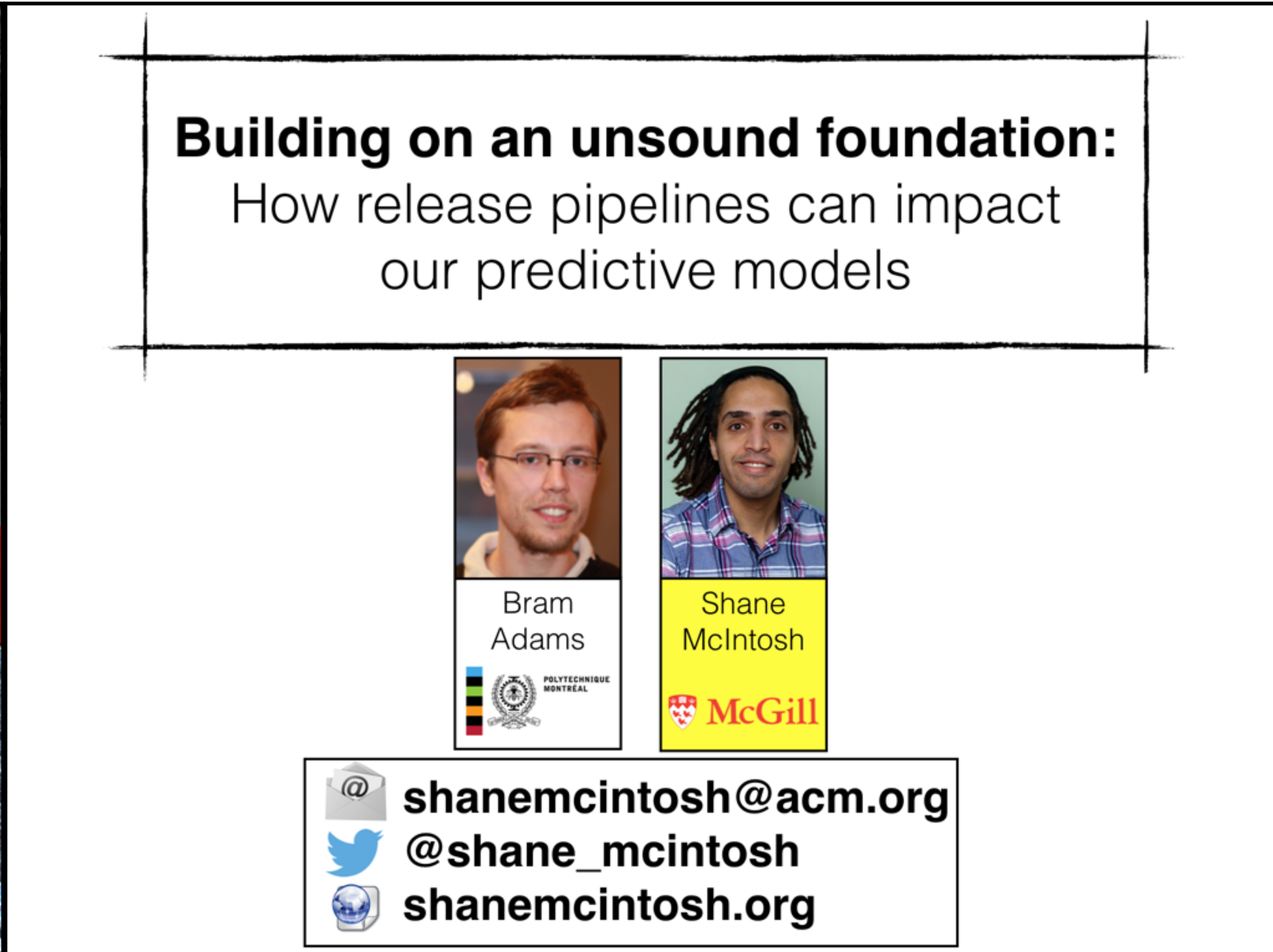


Integration → Build → Deployment

Risk assessment

Build optimization

Smarter deployment tooling



Building on an unsound foundation:

How release pipelines can impact our predictive models

Bram Adams
POLYTECHNIQUE MONTRÉAL

Shane McIntosh
McGill

shanemcintosh@acm.org
@shane_mcintosh
shanemcintosh.org