



zenika  
<animés par la passion>

# Reveal my AsciiDoc

*a deep dive into presentation-as-code*

2020-01-09 - Benoit COUETIL

# Table of Contents

Introduction .....	1
Diagram-as-code .....	2
Sequence diagram .....	2
Source .....	2
Use case diagram .....	2
Source .....	3
Class diagram .....	3
Source .....	3
Activity diagram .....	4
Source .....	4
Component diagram .....	5
Source .....	5
State diagram .....	6
Source .....	6
Work Breakdown Structure .....	7
Source .....	7
Timing diagram .....	7
Source .....	8
Mind Map .....	8
Source .....	9
Timeline-as-code .....	10
Source .....	10
Timeline-as-code fragmented .....	10
Pyramid-as-code .....	11
Source .....	11
Pyramid-as-code fragmented .....	11
Screenshot-as-code .....	12
Last slide .....	13
Positioning and sizing .....	14
Text wrapping near image .....	14
Source .....	15
Maxed-out image .....	16
Source .....	16
Big image .....	16

# Introduction

Standard AsciiDoctor capabilities are shown in [this presentation](#).

The current presentation shows advanced AsciiDoctor / Reveal.js capabilities, mostly helping going deeper in the everything-as-code devOps principle.

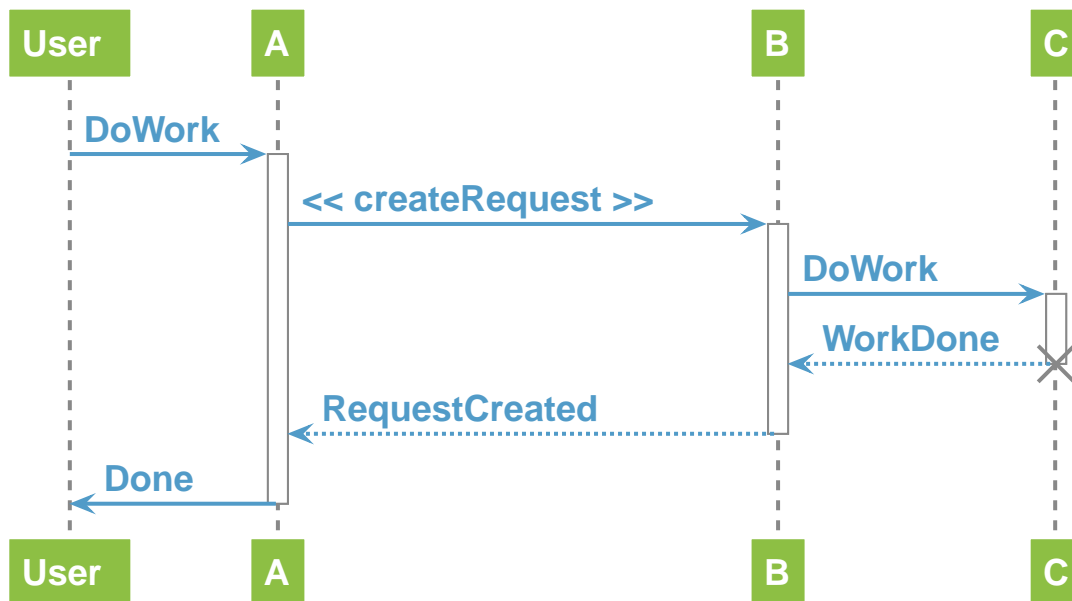
*Guidelines for enriching the toolbox*

- Fewest non-generated images
- Simplest source code
- Fewest reveal.js or AsciiDoctor fork (none so far)

# Diagram-as-code

All these architecture diagrams use PlantUML, with a custom skin to fit the theme.

## Sequence diagram



## Source

```
[plantuml, rma-sequence-diagram, svg]
....
participant User
User -> A: DoWork
activate A
A -> B: << createRequest >>
activate B
B -> C: DoWork
activate C
C --> B: WorkDone
destroy C
B --> A: RequestCreated
deactivate B
A -> User: Done
deactivate A
....
```

## Use case diagram



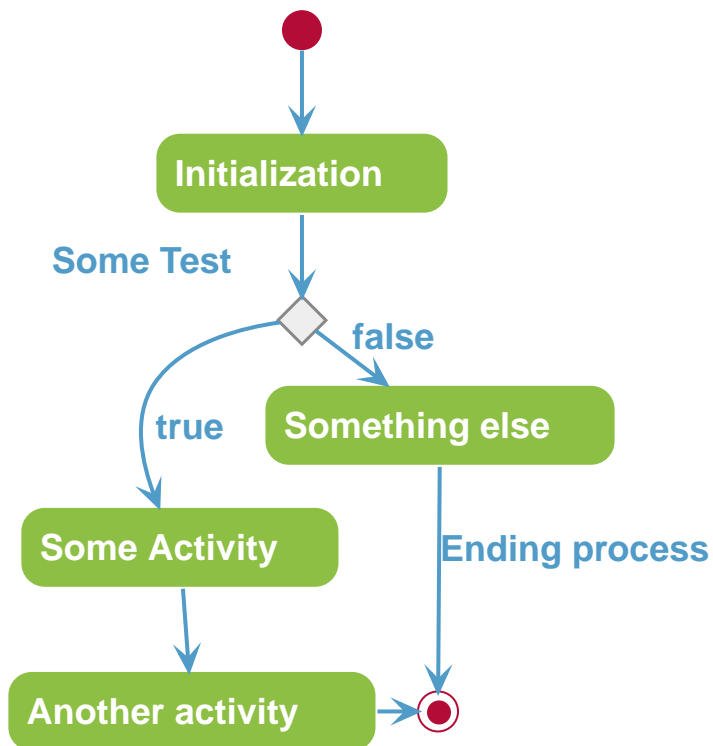
```
[plantuml, rma-class-diagram, svg]
....
abstract class ArrayList {
  Object[] elementData
  size()
}
enum TimeUnit {
  DAYS
  HOURS
  MINUTES
}
Interface BaseClass

namespace net.dummy {
  .BaseClass <|-- Person
  Meeting o-- Person

  .BaseClass <|-- Meeting
}
namespace net.foo {
  net.dummy.Person <|-- Person
  .BaseClass <|-- Person

  net.dummy.Meeting o-- Person
}
BaseClass <|-- net.unused.Person
....
```

## Activity diagram



## Source

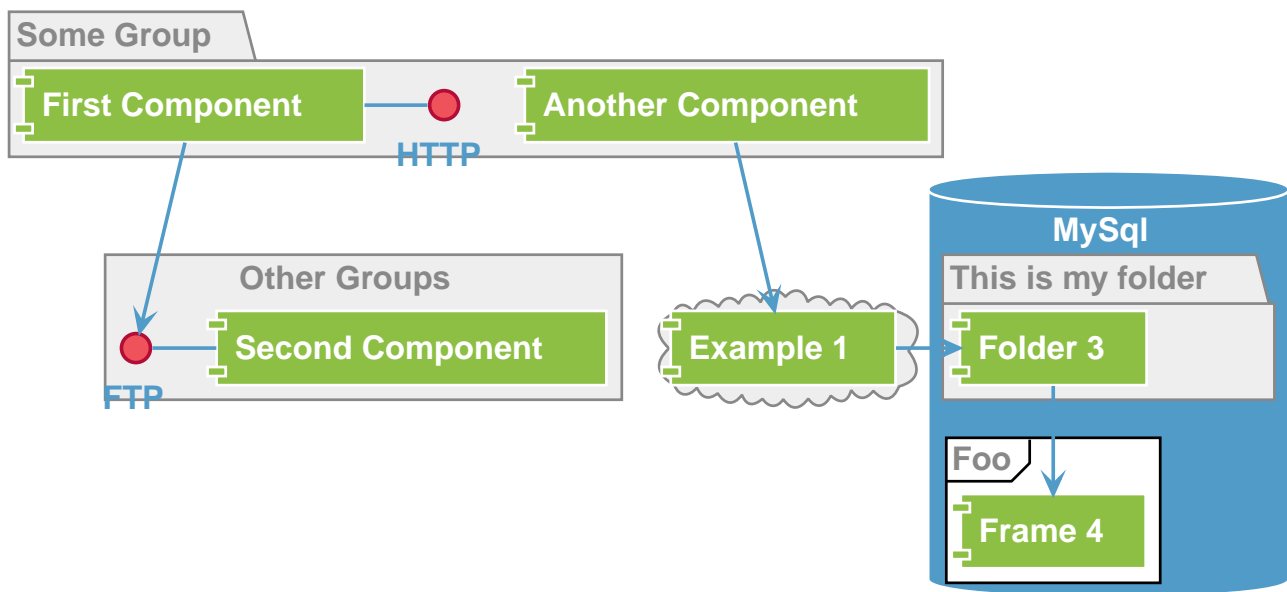
```

[plantuml, activity-diagram, svg]
....
(*) --> "Initialization"

if "Some Test" then
-->[true] "Some Activity"
--> "Another activity"
-right-> (*)
else
->[false] "Something else"
-->[Ending process] (*)
endif
....

```

## Component diagram



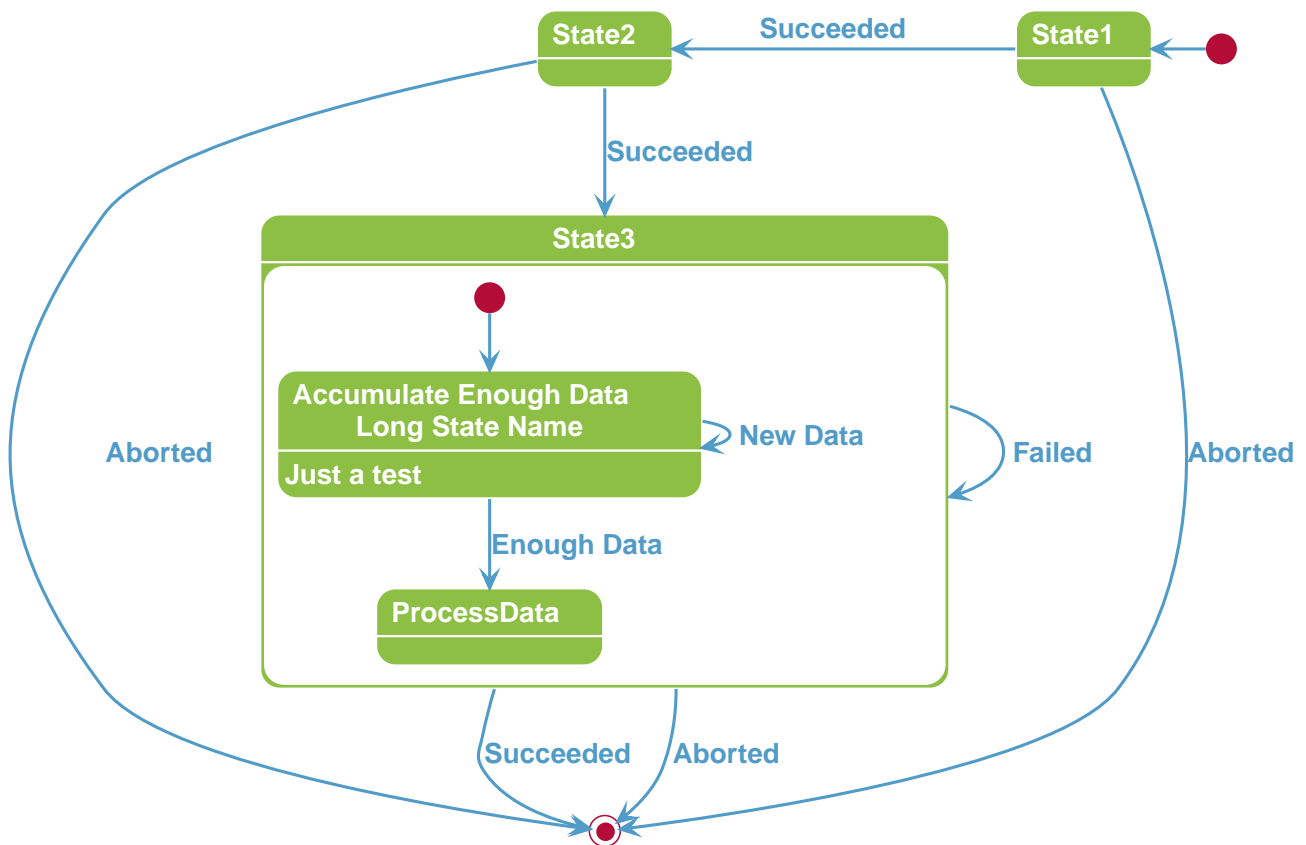
## Source

```

[plantuml, rma-component-diagram, svg]
....
package "Some Group" {
  HTTP - [First Component]
  [Another Component]
}
rectangle "Other Groups" {
  FTP - [Second Component]
  [First Component] --> FTP
}
cloud {
  [Example 1]
}
database "MySQL" {
  folder "This is my folder" {
    [Folder 3]
  }
  frame "Foo" {
    [Frame 4]
  }
}
[Another Component] --> [Example 1]
[Example 1] -right-> [Folder 3]
[Folder 3] --> [Frame 4]
....

```

## State diagram

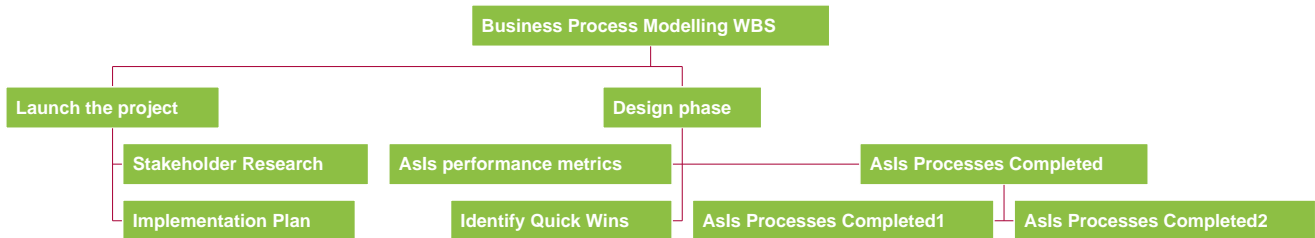


## Source



```
[plantuml, rma-state-diagram, svg]
....
[*] -left-> State1
State1 -left-> State2 : Succeeded
State1 --> [*] : Aborted
State2 --> State3 : Succeeded
State2 --> [*] : Aborted
state State3 {
  state "Accumulate Enough Data\nLong State Name" as long1
  long1 : Just a test
  [*] --> long1
  long1 --> long1 : New Data
  long1 --> ProcessData : Enough Data
}
State3 --> State3 : Failed
State3 --> [*] : Succeeded
State3 --> [*] : Aborted
....
```

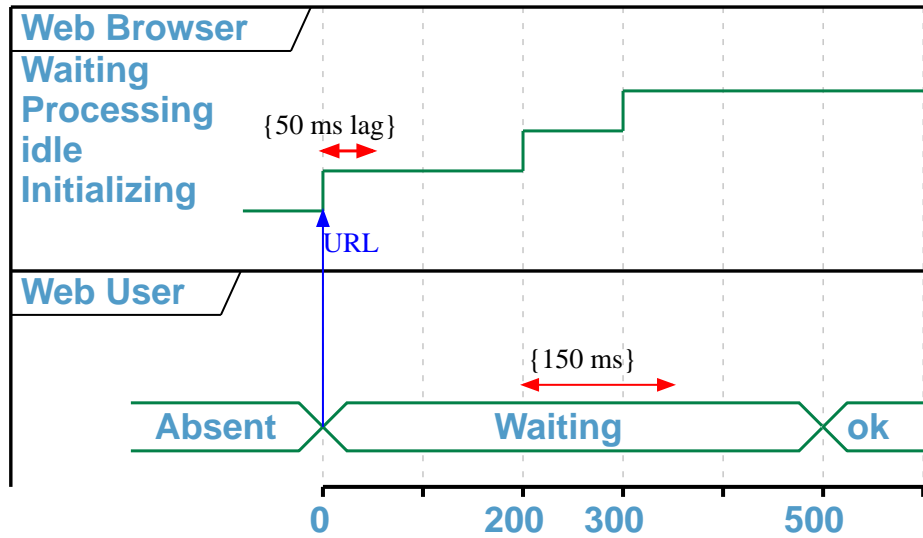
## Work Breakdown Structure



## Source

```
[plantuml, wbs, svg]
....
@startwbs
* Business Process Modelling WBS
** Launch the project
*** Stakeholder Research
*** Implementation Plan
** Design phase
*** AsIs Processes Completed
****< AsIs Processes Completed1
****> AsIs Processes Completed2
***< AsIs performance metrics
***< Identify Quick Wins
@endwbs
....
```

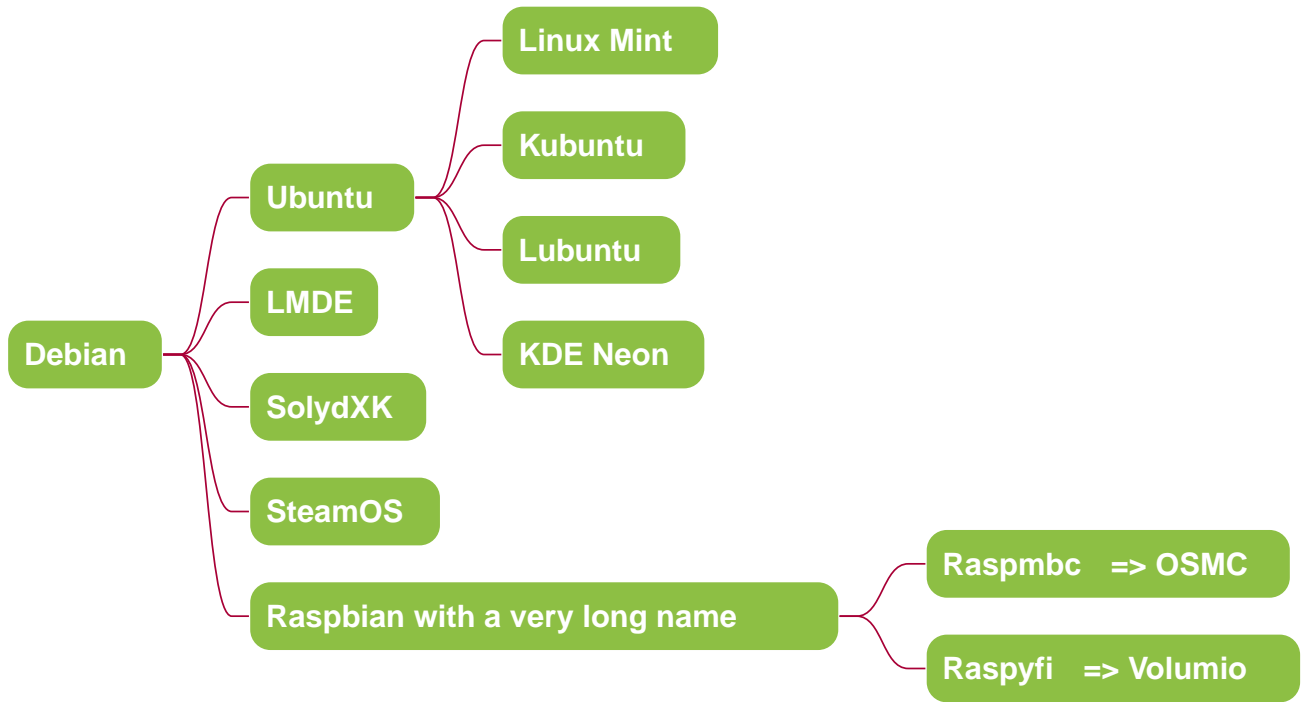
## Timing diagram



## Source

```
[plantuml, timing, svg]
....
'skinparam backgroundColor white # does not work
robust "Web Browser" as WB
concise "Web User" as WU
WB is Initializing
WU is Absent
@WB
0 is idle
+200 is Processing
+100 is Waiting
WB@0 <-> @50 : {50 ms lag}
@WU
WU -> WB : URL
0 is Waiting #white
+500 is ok #white
@200 <-> @+150 : {150 ms}
....
```

## Mind Map



## Source

```

[plantuml, mindmap, svg]
....
@startmindmap
* Debian
** Ubuntu
*** Linux Mint
*** Kubuntu
*** Lubuntu
*** KDE Neon
** LMDE
** SolydXK
** SteamOS
** Raspbian with a very long name
*** <s>Raspmbc</s> => OSMC
*** <s>Raspyfi</s> => Volumio
@endmindmap
....

```

# Timeline-as-code

- 1997 - Mondrian (Google)
- 2002 - Codestriker (IBM)
- 2003 - Rietveld (Google)
- 2004 - Gerrit (Google, Android, Eclipse, OpenStack, GWT, IBM...)
- 2007 - Crucible (Atlassian)
- 2007 - Phabricator (Facebook, AngularJS, Quora, Uber)
- 2008 - Github
- 2011 - Gitlab
- 2012 - CodeFlow (Microsoft)

## Source

```
[.timeline]
* *1997* - Mondrian (Google)
* *2002* - Codestriker (IBM)
* *2003* - Rietveld (Google)
* *2004* - Gerrit (Google, Android, Eclipse, OpenStack, GWT, IBM...)
* *2007* - Crucible (Atlassian)
* *2007* - Phabricator (Facebook, AngularJS, Quora, Uber)
* *2008* - Github
* *2011* - Gitlab
* *2012* - CodeFlow (Microsoft)
```

To reveal lines as fragments, use [%step] and put the [.timeline] before the title. See the result on next slide.



Skip fragments with [PageDown](#) / [PageUp](#).

## Timeline-as-code fragmented

- 1997 - Mondrian (Google)
- 2002 - Codestriker (IBM)
- 2003 - Rietveld (Google)
- 2004 - Gerrit (Google, Android, Eclipse, OpenStack, GWT, IBM...)
- 2007 - Crucible (Atlassian)
- 2007 - Phabricator (Facebook, AngularJS, Quora, Uber)
- 2008 - Github
- 2011 - Gitlab
- 2012 - CodeFlow (Microsoft)

# Pyramid-as-code

- Continuous deployment
- Continuous reporting
- Pre-commit pipeline
- Continuous integration/testing
- Design & architecture
- Continuous improvement
- Quality of work life
- Organization and culture

## Source

```
[.pyramid]
* Continuous deployment
* Continuous reporting
* Pre-commit pipeline
* Continuous integration/testing
* Design & architecture
* Continuous improvement
* Quality of work life
* Organization and culture
```

To reveal lines as fragments, use [%step] and put the [.pyramid] before the title. See the result on next slide.

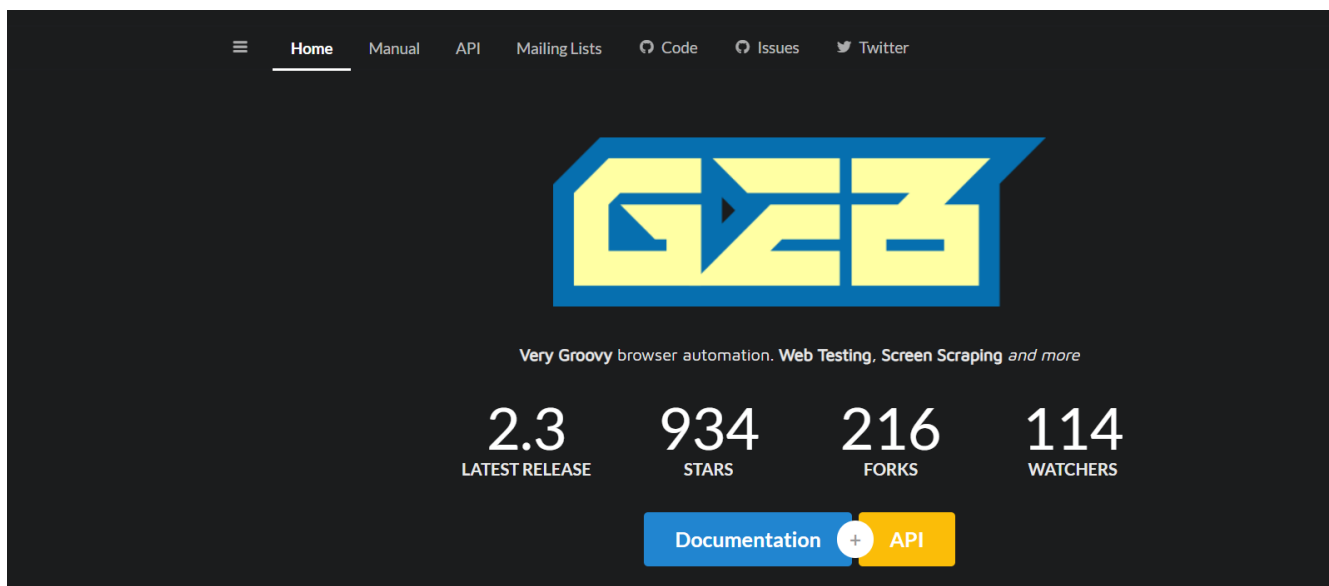


You can skip fragments with [PageDown](#) / [PageUp](#).

## Pyramid-as-code fragmented

- Continuous deployment
- Continuous reporting
- Pre-commit pipeline
- Continuous integration/testing
- Design & architecture
- Continuous improvement
- Quality of work life
- Organization and culture

# Screenshot-as-code



```
screenshot::http://www.gebish.org[asciidoctorj-screenshot,dimension=1600x800,width=600]
```

This uses the [screenshot asciidoc plugin](#) based on [Geb](#).

# Last slide

The last slide is there to open for questions.

We chose to have a slide with transparent background to reveal the background image.

```
[.questions]
```

```
=== !
```

```
[.bubbles]
```

```
=== !
```

```
[.hands]
```

```
=== !
```



They are appearing top to bottom here, this will be left to right on a standard presentation.



You don't have to use one, but you will then lose the last section in the ToC, because it is always hidden for this purpose.

# Positioning and sizing

Here are some positioning and sizing introduced specifically for Reveal.js slides.

```
[.halign-center]
```

You can center anything horizontally.

```
[.bottom]
```

You can put anything at the bottom.

You can center anything horizontally.

You can put anything at the bottom.

## Text wrapping near image




Three Rings for the Elven-kings under the sky,  
Seven for the Dwarf-lords in their halls of stone,  
Nine for Mortal Men doomed to die,  
One for the Dark Lord on his dark throne





In the Land of Mordor where the Shadows lie.

One Ring  to rule them all, One Ring  to find them,

One Ring  to bring them all and in the darkness bind them

In the Land of Mordor where the Shadows lie.

## Source

```
image::one-ring.jpg[width=300,float=left] ①
```

```
Three Rings for the Elven-kings under the sky,  
Seven for the Dwarf-lords in their halls of stone,  
Nine for Mortal Men doomed to die,  
One for the Dark Lord on his dark throne
```

```
image::one-ring.jpg[width=450,float=right] ②
```

```
In the Land of Mordor where the Shadows lie.
```

```
[.green] ③
```

```
*One Ring image:one-ring.jpg[width=40] to rule them all, One Ring image:one-ring.jpg[width=40] to find them,*
```

```
[.green] ③
```

```
*One Ring image:one-ring.jpg[width=40] to bring them all and in the darkness bind them*
```

```
In the Land of Mordor where the Shadows lie.
```

① :: block image float left

② :: block image float right

③ : inline images

## Maxed-out image



## Source

Whatever the image size is, it will cover space without changing the ratio.

```
[.maxed-image]  
image::sunset.jpg[]
```

## Big image



[.big-image] is the same as [.maxed-image] but gives some space for text.

[.big-image]  
image::sunset.jpg[]