INSA

Automated process for software programming

PDLA : Processus de Développement Logiciel Automatisé

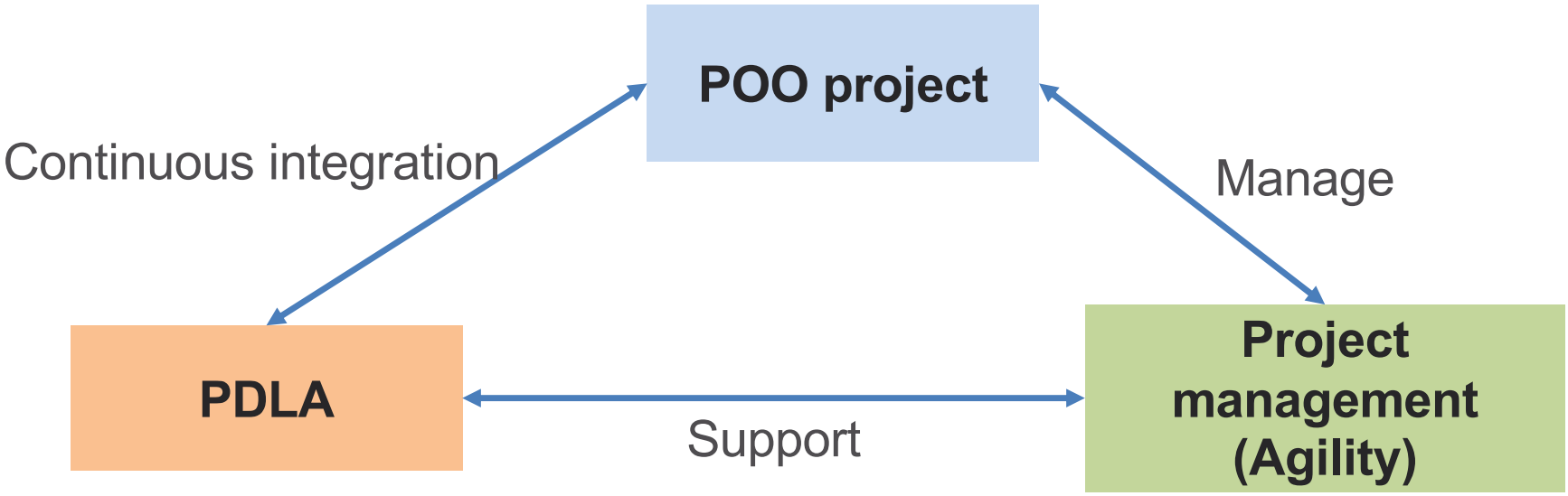
« **Dev**ops – **Continous integration & Deployment (CI/CD)** »

Nawal Guermouche
guermouc@insa-toulouse.fr

Organization

INSA

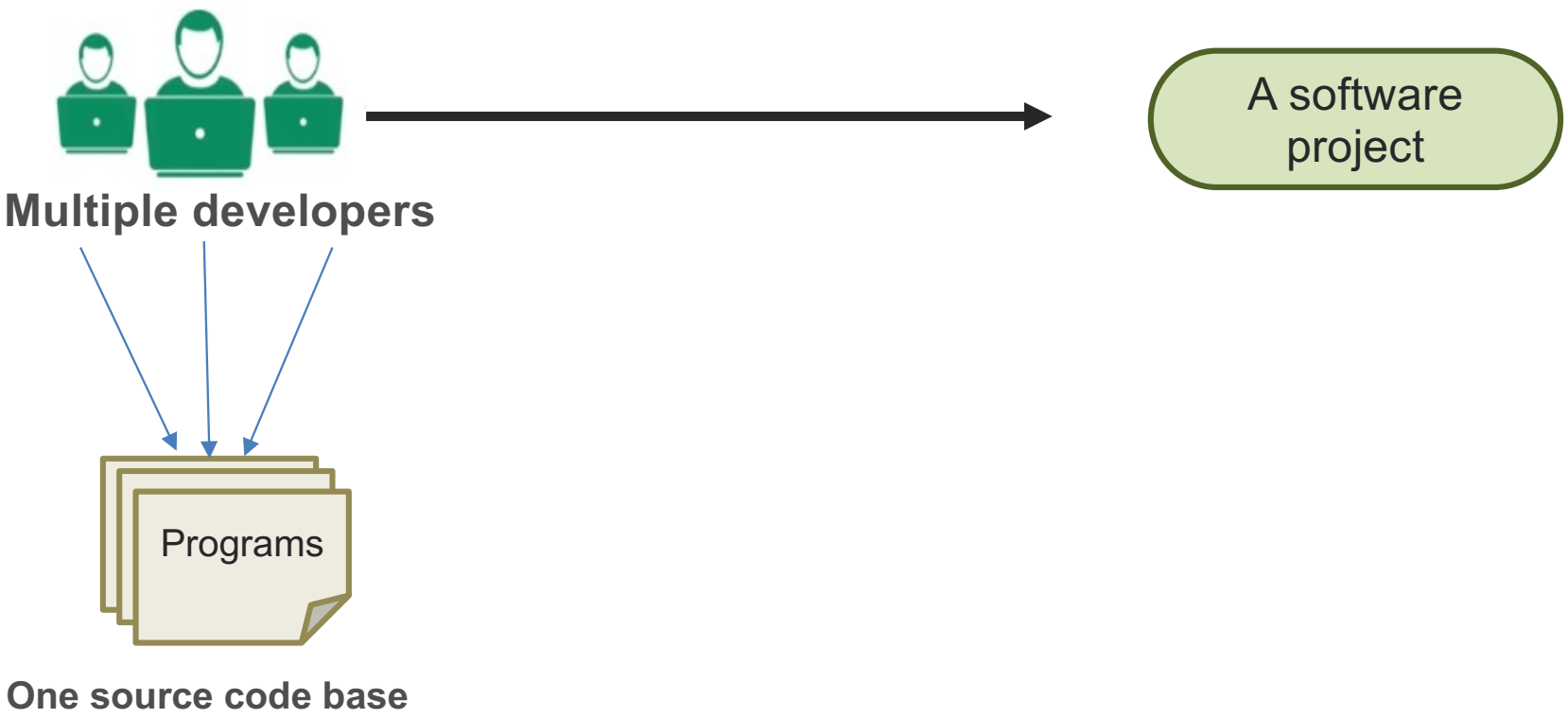
- 1 course to introduce related concepts
- 7 directed works
- Moodle :
  - Name : Processus de développement logiciel automatisé - PDLA
  - Key : 4APDLA


POO project
Continuous integration
Manage
PDLA
Support
Project management (Agility)

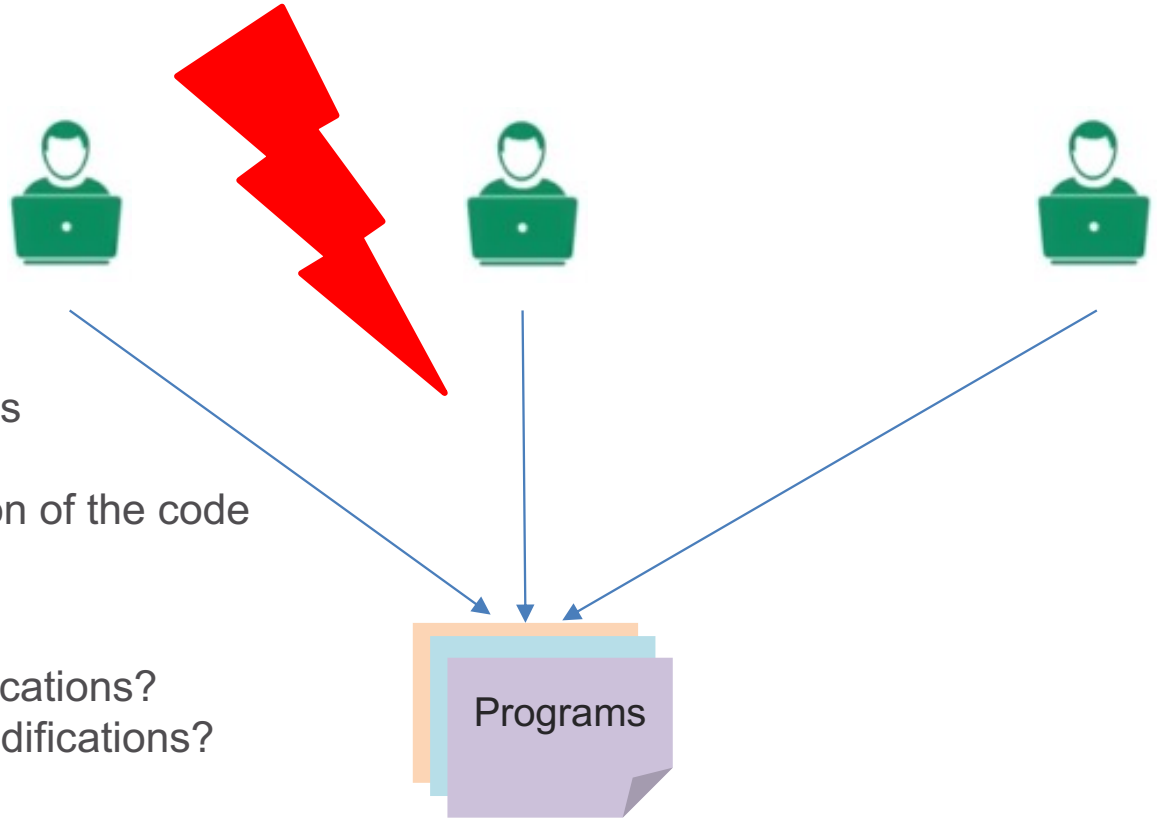
Multiple developers

A software project

- The source code



INSA



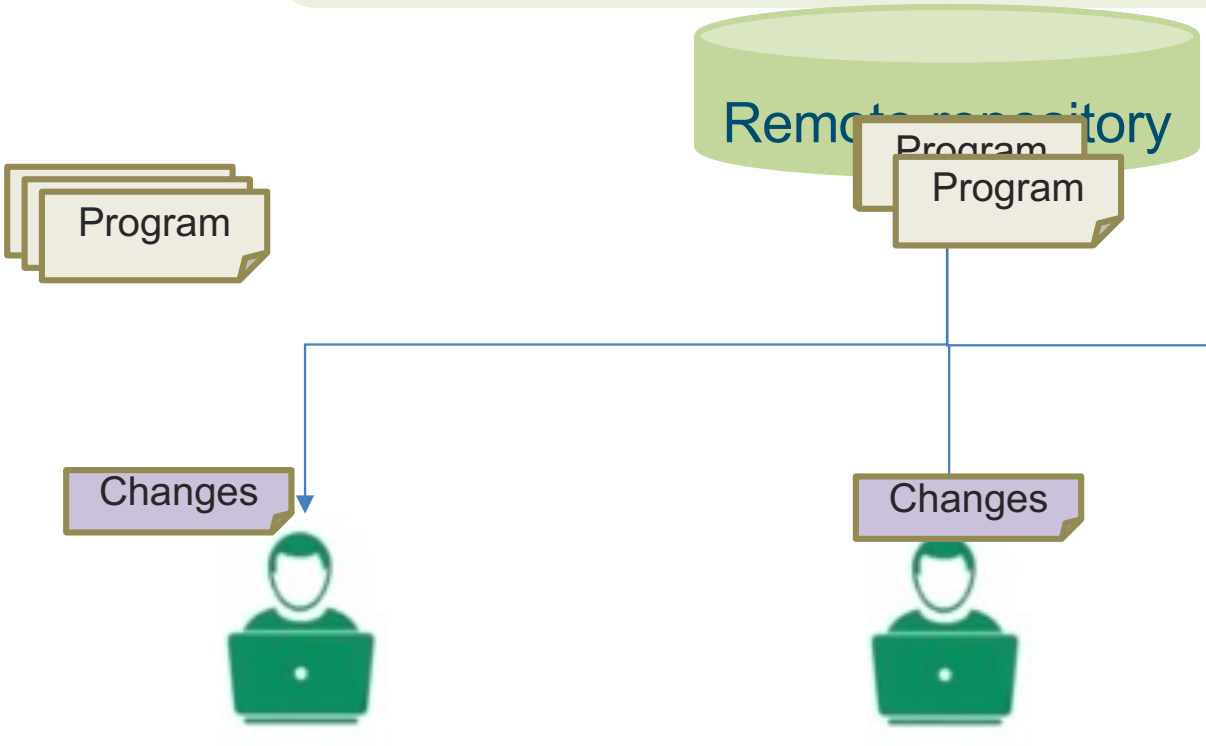
✗ Waiting the code of other colleagues
✗ Problems (e.g., lost code)
✗ Impossible to get a particular version of the code

How to avoid these problems?

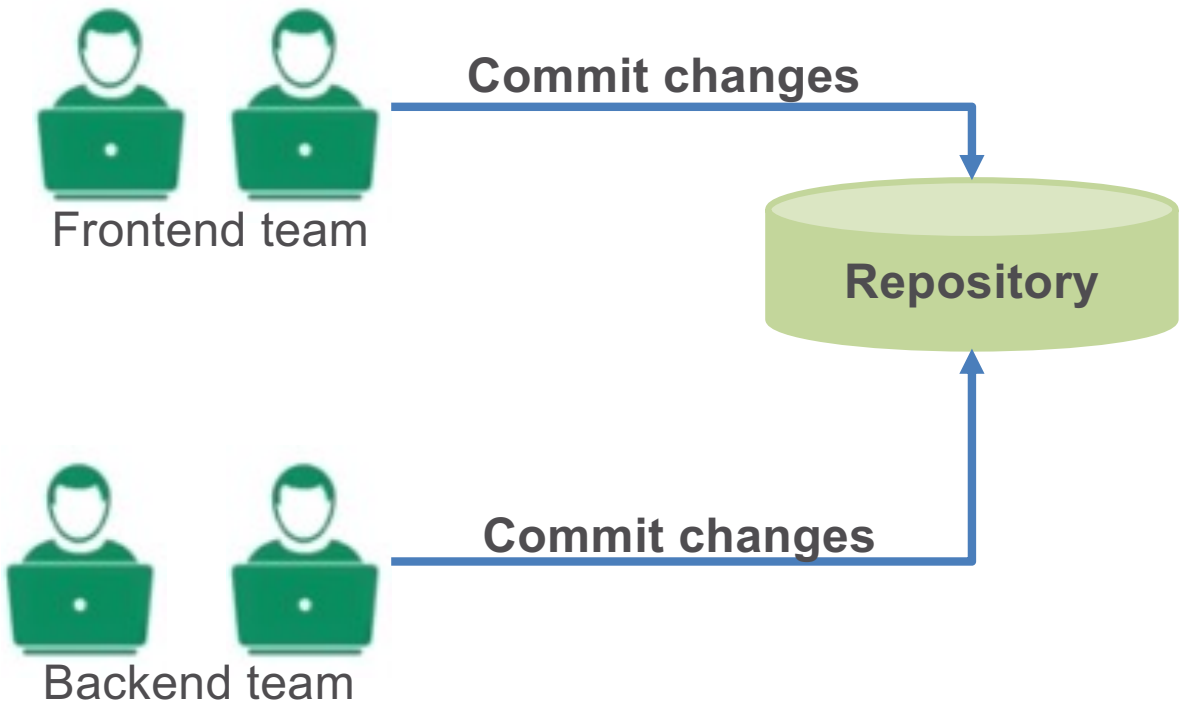
✓ how to integrate the different modifications?
✓ how to be aware of the different modifications?
✓ how to manage conflicts?
✓ how to come back to previous versions just before some modifications?

Use a Revision Control System

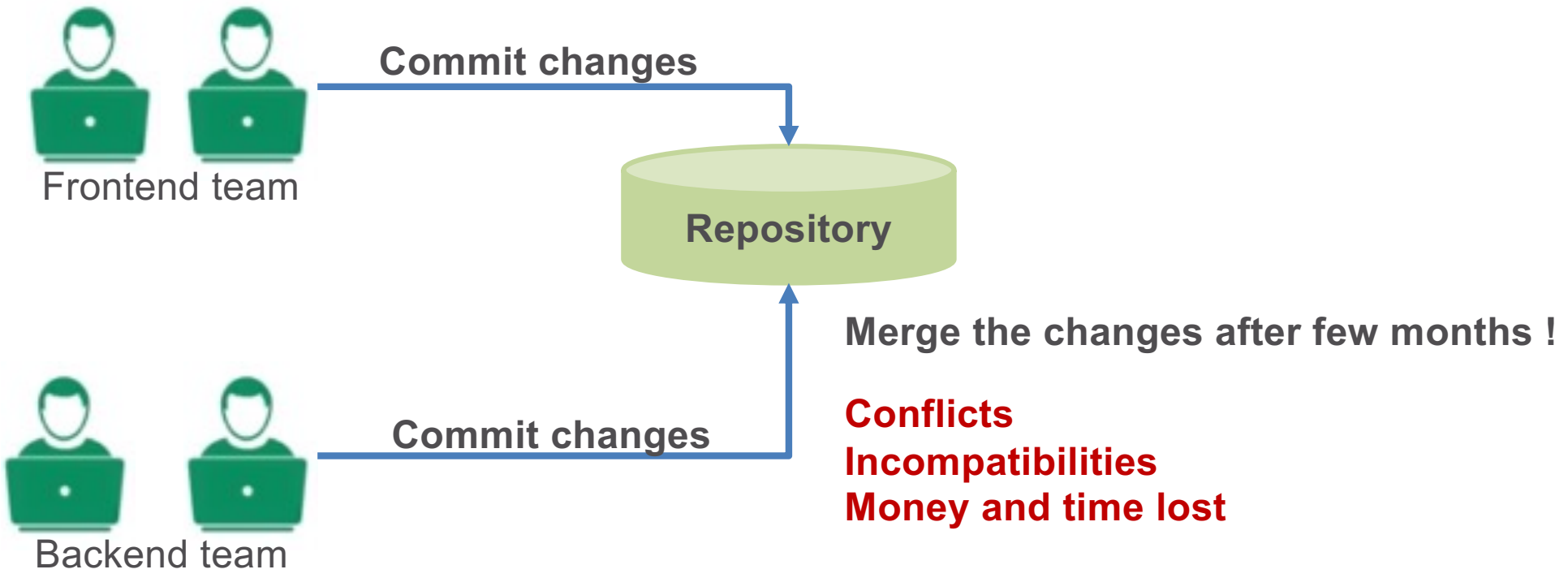
Revision Control System Tool
(e.g., Git)



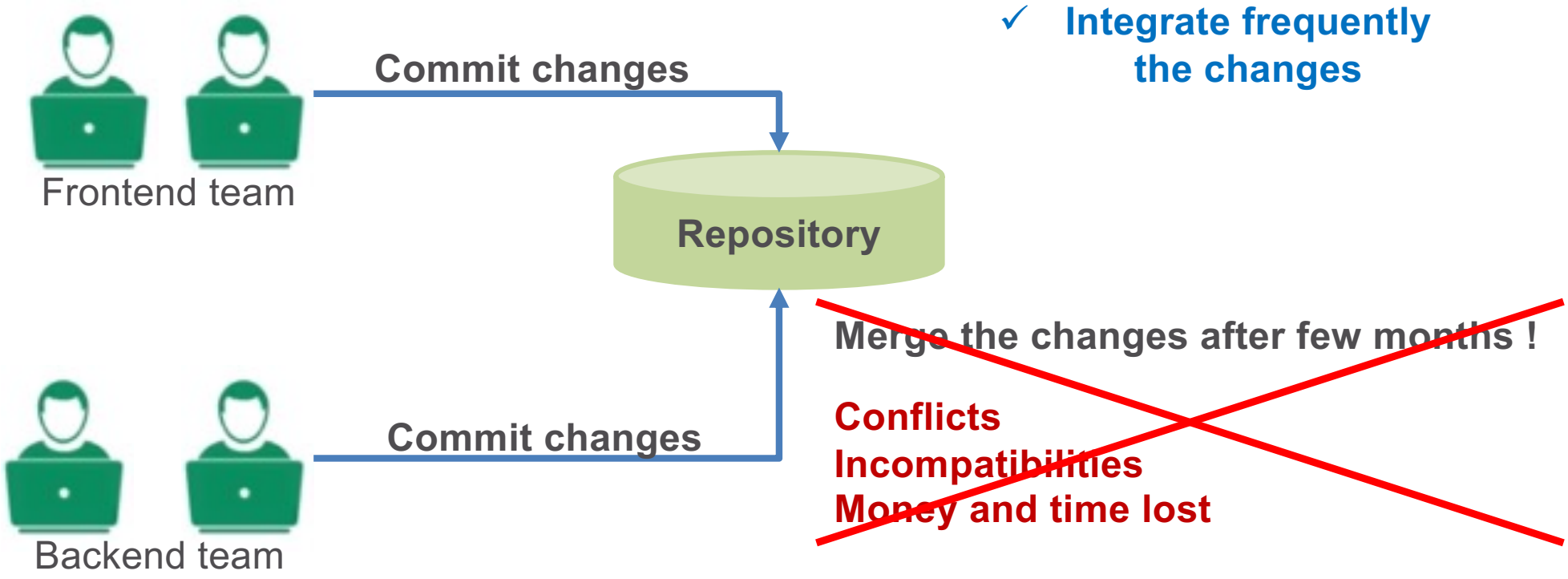
- ✓ Enables a collaborative work
- ✓ Enables the management of versions
- ✓ Enables managing conflicts



Developing a Web application example



Developing a Web application example

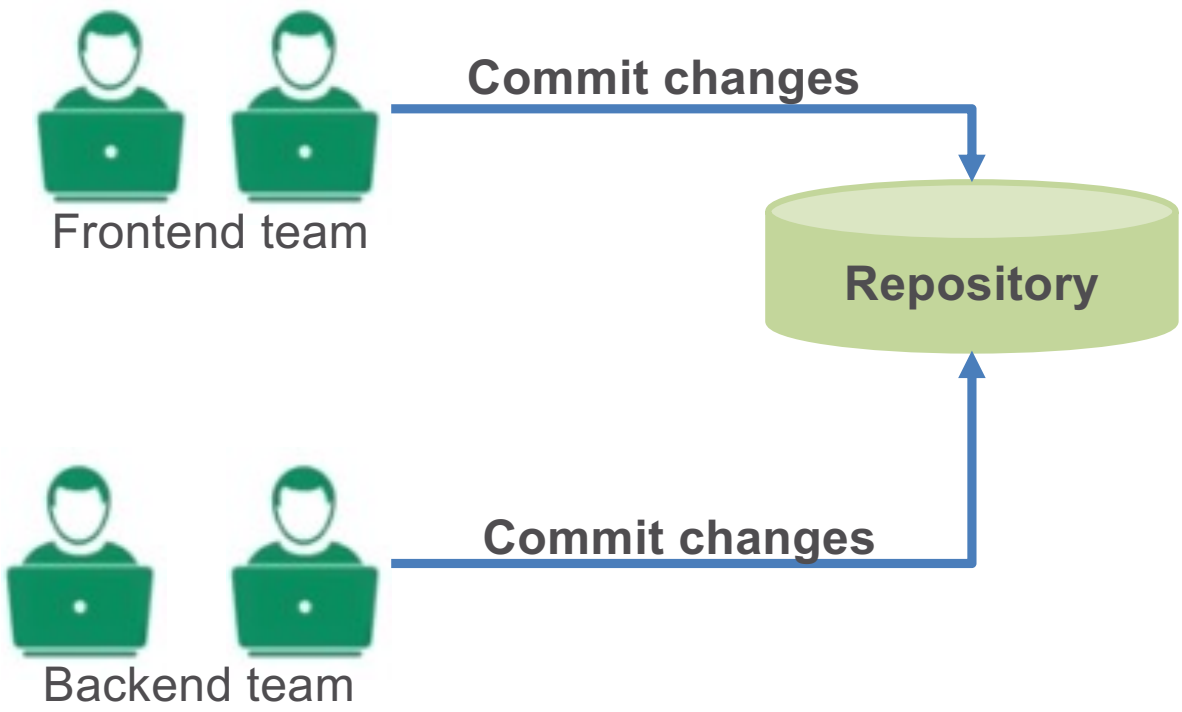


✓ Integrate frequently the changes

Merge the changes after few months !

Conflicts
Incompatibilities
Money and time lost

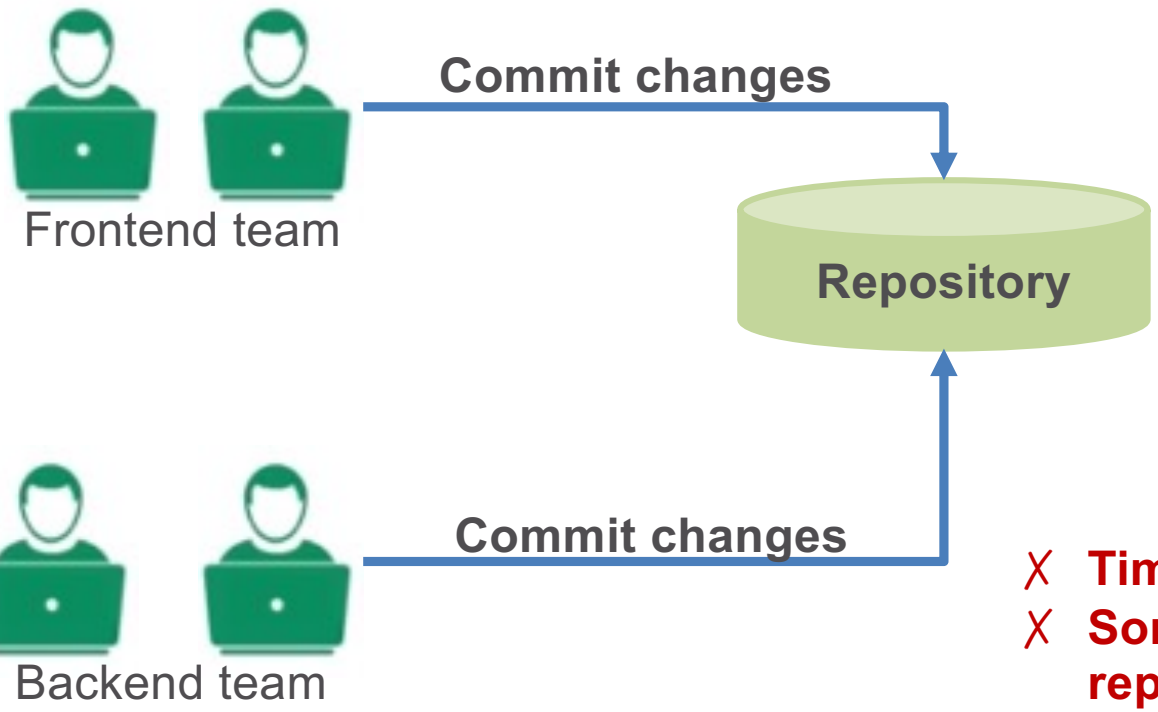
Developing a Web application example



✓ **Integrate frequently the changes**

- Build
- Test
- Deploy

Developing a Web application example



✓ **Integrate frequently the changes**

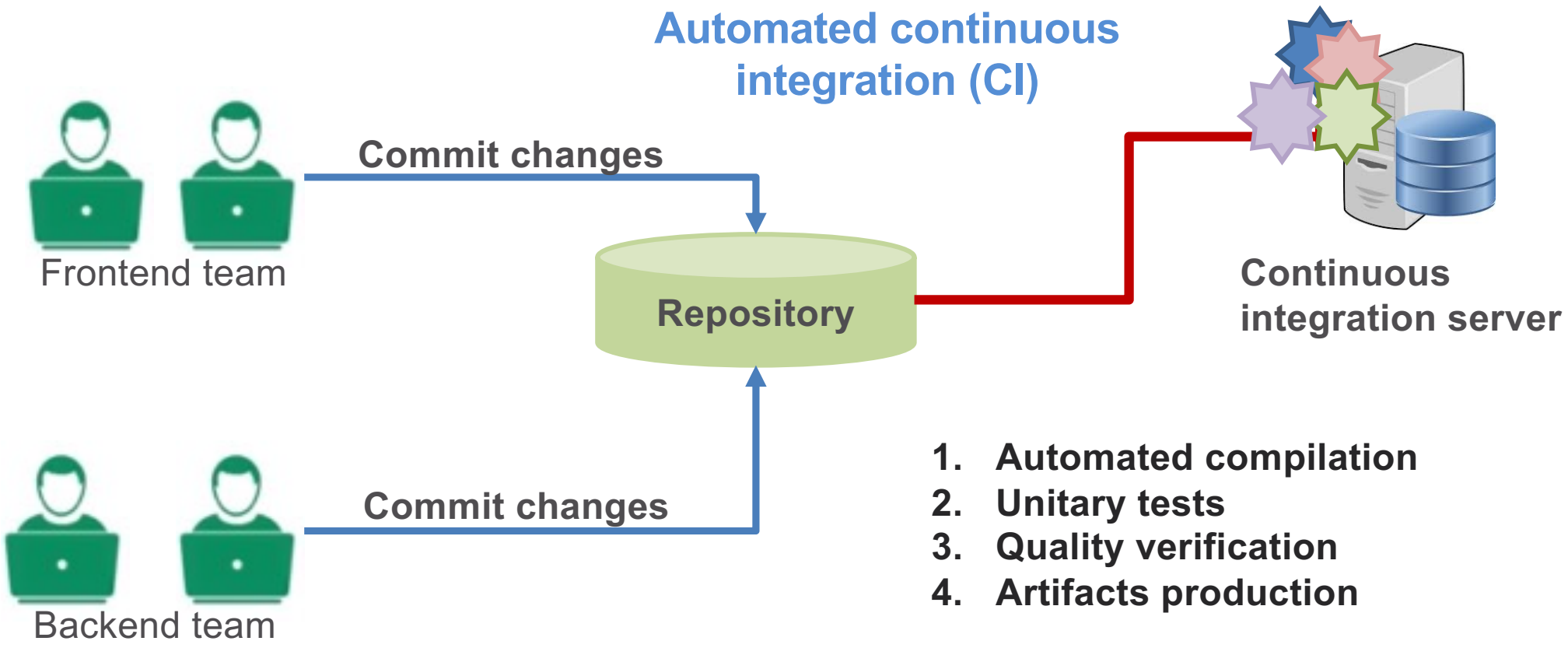
- Build
- Test
- Deploy

Manual integration

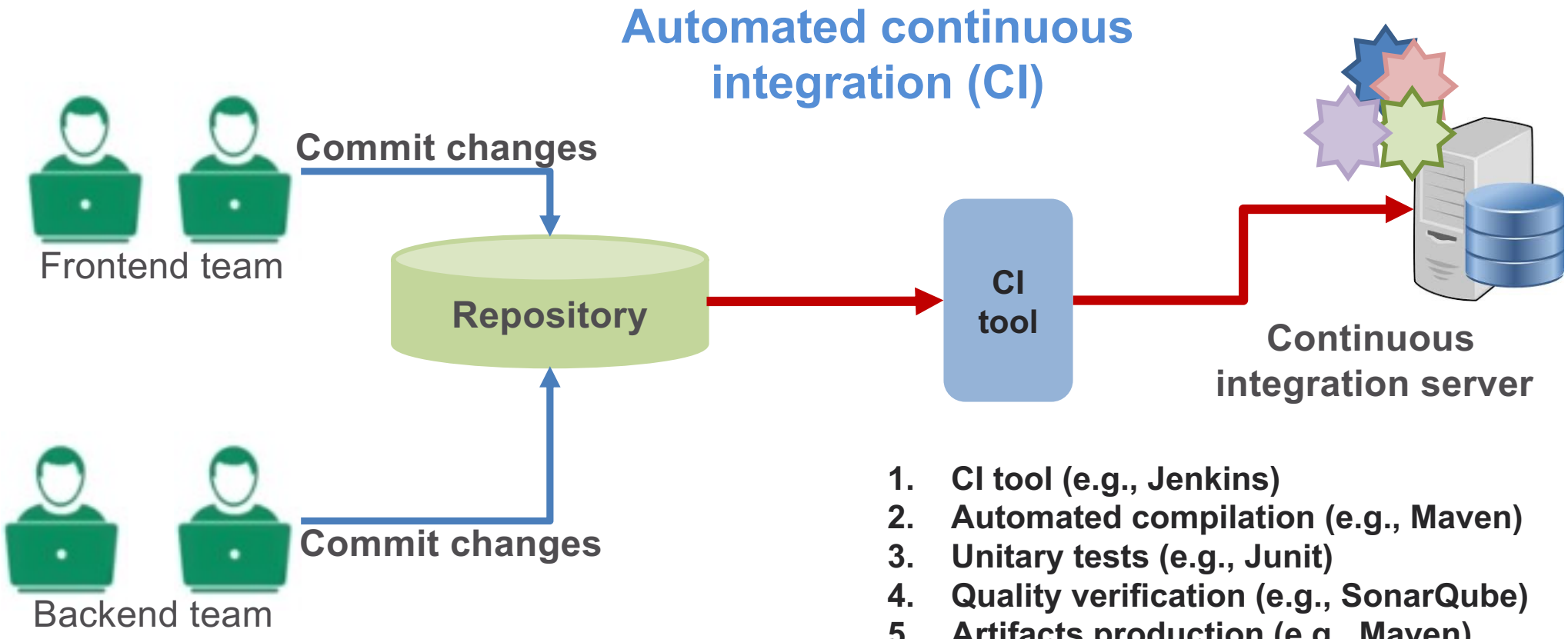
✗ **Time lost**

✗ **Some processes are not reproducible (like tests)**

Developing a Web application example

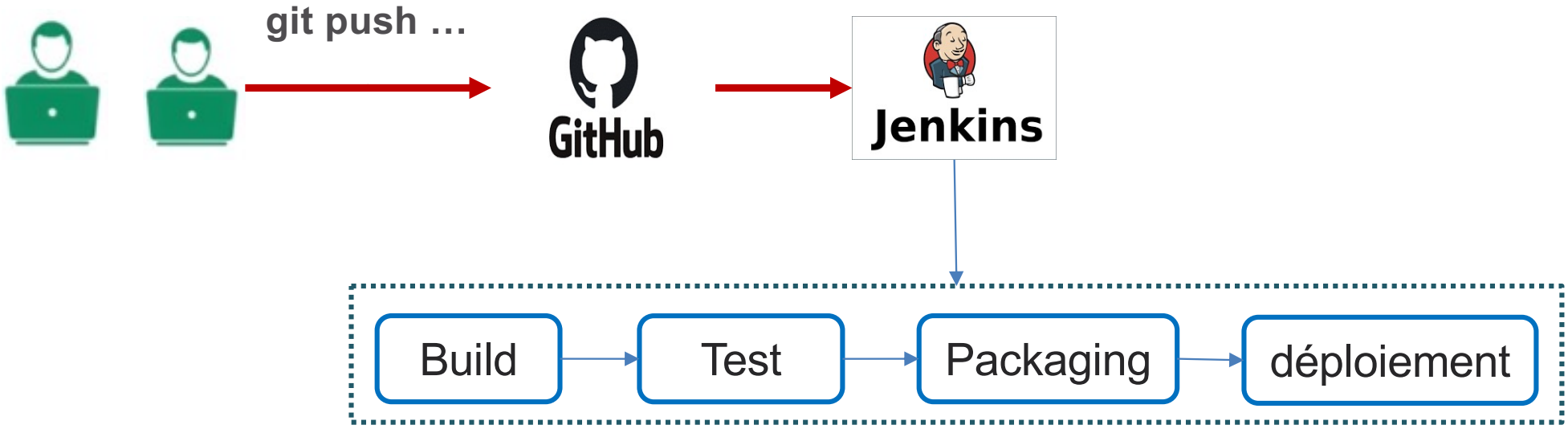


Developing a Web application example



Developing a Web application example

Example



Data format

- Tools use configuration files

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
https://maven.apache.org/xsd/maven-4.0.0.xsd">
    <modelVersion>4.0.0</modelVersion>

    <groupId>org.springframework</groupId>
    <artifactId>gs-maven</artifactId>
    <packaging>jar</packaging>
    <version>0.1.0</version>

    <properties>
        <maven.compiler.source>1.8</maven.compiler.source>
        <maven.compiler.target>1.8</maven.compiler.target>
    </properties>

    <build>
        <plugins>
            <plugin>
                <groupId>org.apache.maven.plugins</groupId>
                <artifactId>maven-shade-plugin</artifactId>
                <version>3.2.4</version>
                <executions>
                    <execution>
                        <phase>package</phase>
                        <goals>
                            <goal>shade</goal>
                        </goals>
                        <configuration>
                            <transformers>
                                <transformer

implementation="org.apache.maven.plugins.shade.resource.ManifestResourceTransform
er">
                                    <mainClass>hello.HelloWorld</mainClass>
                                </transformer>
                            </transformers>
                        </configuration>
                    </execution>
                </executions>
            </plugin>
        </plugins>
    </build>
</project>
```

Data format

INSA

- Tools use configuration files

- Specified format
  - XML (eXtensible Markup Language)
  - JSON (JavaScript Object Notation)
  - YAML (Yet Another Markup Language then Yaml Ain’t Markup Language)

INSA

XML (eXtensible Markup Language)

XML

INSA

- File with extension .xml
- Text oriented data format
- Relies on **markups**
- Tree of different nodes: element, attribute, text

```
<?xml version ="1.0" encoding="iso-8859-1" ?>

<laboratory type="recherche">

    <name> LAAS </name>

    <town> Toulouse </town>

    <specialties>
        <specialty> Computer science </specialty>
        <specialty> Automatic </specialty>
    <specialties>

</laboratory>
```

XML

INSA

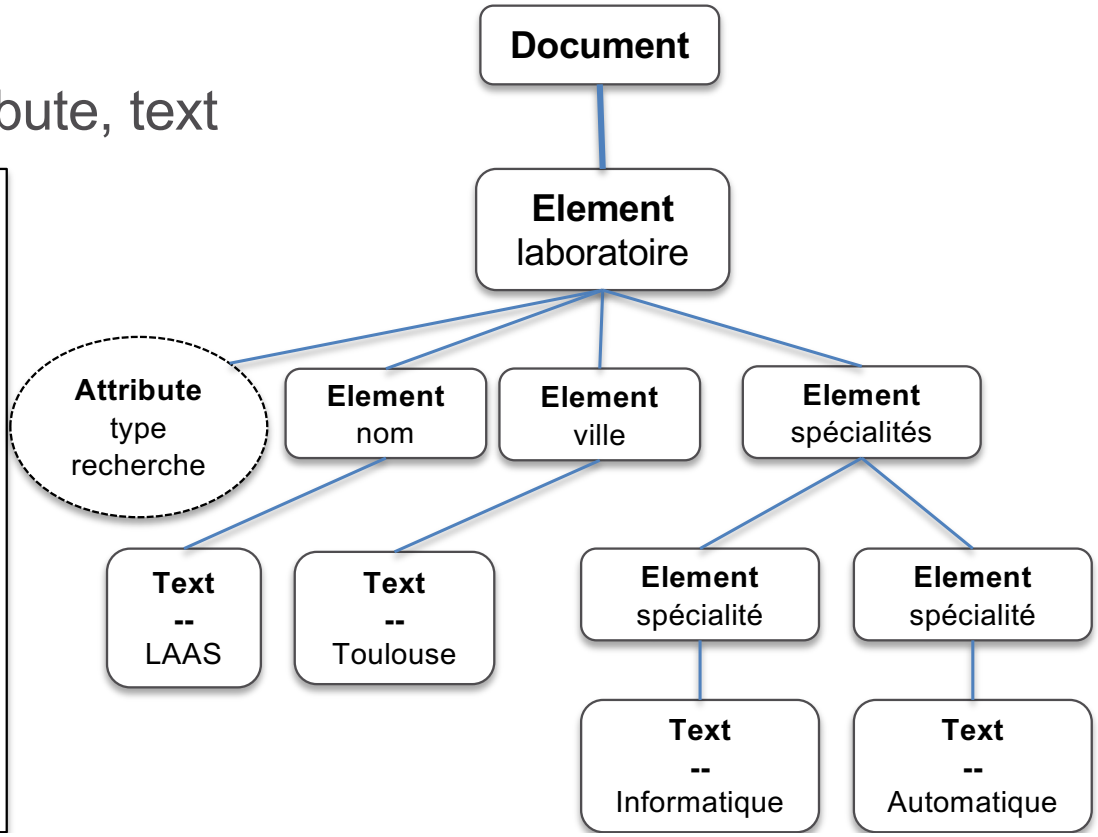
- File with extension .xml
- Text oriented data format
- Relies on **markups**
- Tree of different nodes: element, attribute, text

```
<?xml version ="1.0" encoding="iso-8859-1" ?>

<laboratory type="recherche">

    <name> LAAS </name>

    <town> Toulouse </town>

    <specialties>
        <specialty> Computer science </specialty>
        <specialty> Automatic </specialty>
    <specialties>

</laboratory>
```

XML

- File with extension .xml
- Text oriented data format
- Relies on **markups**
- Tree of different nodes: element, attribute, text

```
<?xml version ="1.0" encoding="iso-8859-1" ?>

<laboratory type="recherche">

    <name> LAAS </name>

    <town> Toulouse </town>

    <specialties>
        <specialty> Computer science </specialty>
        <specialty> Automatic </specialty>
    <specialties>

</laboratory>
```



XML

INSA

```
<?xml version ="1.0" encoding="iso-8859-1" ?>

<laboratory type="recherche">

<!--Laboratoire du CNRS-->

    <name> LAAS </name>

    <town> Toulouse </town>

    <specialties>
        <specialty> Computer science </specialty>
        <specialty> Automatic </specialty>
        <specialty/>
    <specialties>

</laboratory>
```

XML : namespace

```
<laboratory>
    <name> LAAS </name >
    <town> Toulouse </town >
    <specialties>
        <specialty> Computer science </specialty>
        <specialty> Automatic </specialty >
    </specialties>
</laboratory >
```

```
<projects>
  <project>
      <name> SoSCP </name>
      <funding> CIMI </funding>
      <partner> LAAS  </partner>
      <partner> IRIT </partner>
  </project>
</projects>
```

XML : namespace

```
<laboratory>
    <name> LAAS </ name >
    <town> Toulouse </town >
    <specialties>
        <specialty> Computer science </specialty>
        <specialty> Automatic </specialty >
    </specialties>
</laboratory >
```

```
<projects>
  <project>
      <name> SoSCP </name>
      <funding> CIMI </funding>
      <partner> LAAS  </partner>
      <partner> IRIT </partner>
  </project>
</projects>
```

Need to merge the documents into one document

```
<?xml version ="1.0" encoding="iso-8859-1" ?>
<collaboration>
  <laboratory>
      …
  </laboratory>

  <projects>
    <project>
      …
    </project>
  </projects>
</collaboration>
```

XML : namespace

```
<laboratory>
   <name> LAAS </ name >
      <town> Toulouse </town >
      <specialties>
            <specialty> Computer science </specialty>
            <specialty> Automatic </specialty >
      </specialties>
</laboratory >
```

```
<projects>
   <project>
      <name> SoSCP </name>
         <funding> CIMI </funding>
         <partner> LAAS  </partner>
         <partner> IRIT </partner>
   </project>
</projects>
```

Need to merge the documents into one document

Conflict in the element ***name***

```
<?xml version ="1.0" encoding="iso-8859-1" ?>
<collaboration>
   <laboratory>
      <name> LAAS </name>
      …
   </laboratory>

   <projects>
      <project>
         <name> SoSCP </name>
         …
      </project>
   </projects>
</collaboration>
```

XML : namespace

INSA

```
<l:laboratory>
    <l:name> LAAS </l:name >
    <l:town> Toulouse </l:town >
    <l:specialties>
        <l:specialty> Computer science </l:specialty>
        <l:specialty> Automatic </l:specialty >
    </l:specialties>
</l:laboratory>
```

```
<p:projects>
  <p:project>
      <p:name> SoSCP </p:name>
      <p:funding> CIMI </p:funding>
      <p:partner> LAAS  </p:partner>
      <p:partner> IRIT </p:partner>
  </p:project>
</p:projects>
```

```
<?xml version ="1.0" encoding="iso-8859-1" ?>
<collaboration>
  <laboratory>
     <name> LAAS </name>
     …
  </laboratory>

  <projects>
    <project>
      <name> SoSCP </name>
      …
    </project>
  </projects>
</collaboration>
```

XML : namespace

INSA

```
<l:laboratory xmlns:l="http://www.laas.fr">
    <l:name> LAAS </l:name >
    <l:town> Toulouse </l:town >
    <l:specialties>
        <l:specialty> Computer science </l:specialty>
        <l:specialty> Automatic </l:specialty >
    </l:specialties>
</l:laboratory>
```

```
<p:projects xmlns:p="http://www.project.fr/FrProjects">
  <p:project>
      <p:name> SoSCP </p:name>
      <p:funding> CIMI </p:funding>
      <p:partner> LAAS  </p:partner>
      <p:partner> IRIT </p:partner>
  </p:project>
</p:projects>
```

```
<?xml version ="1.0" encoding="iso-8859-1" ?>
<collaboration>
  <laboratory>
     <name> LAAS </name>
     …
  </laboratory>

  <projects>
    <project>
      <name> SoSCP </name>
      …
    </project>
  </projects>
</collaboration>
```

XML : namespace

INSA

```
<l:laboratory xmlns:l="http://www.laas.fr">
    <l:name> LAAS </l:name >
    <l:town> Toulouse </l:town >
    <l:specialties>
        <l:specialty> Computer science </l:specialty>
        <l:specialty> Automatic </l:specialty >
    </l:specialties>
</l:laboratory>
```

```
<p:projects xmlns:p="http://www.project.fr/FrProjects">
  <p:project>
      <p:name> SoSCP </p:name>
      <p:funding> CIMI </p:funding>
      <p:partner> LAAS  </p:partner>
      <p:partner> IRIT </p:partner>
  </p:project>
</p:projects>
```

```
<?xml version ="1.0" encoding="iso-8859-1" ?>
<collaboration>
  <l:laboratory xmlns:l="http://www.laas.fr">
    <l:name> LAAS </l:name>
    …
  </l:laboratory>

  <p:projects>
    <p:projects xmlns:l="http://www.project.fr/FrProjects">
      <p:name> SoSCP </p:name>
      ….
    </p:project>
</p:projects>
</collaboration>
```

XML : namespace

INSA

- A namespace is used to prevent name element conflicts

```
<?xml version ="1.0" encoding="iso-8859-1" ?>
<collaboration>
  <l:laboratory xmlns:l="http://www.laas.fr">
    <l:name> LAAS </l:name>
    …
  </l:laboratory>

  <p:projects>
    <p:projects xmlns:p="http://www.project.fr/FrProjects">
      <p:name> SoSCP </p:name>
      ….
    </p:project>
</p:projects>
</collaboration>
```

XML : namespace

- A namespace is used to prevent name element conflicts

```
<?xml version ="1.0" encoding="iso-8859-1" ?>
<collaboration xmlns:l="http://www.laas.fr"
                    xmlns:p="http://www.project.fr/FrProjects">
  <l:laboratory>
    <l:name> LAAS </l:name>
    …
  </l:laboratory>

  <p:projects>
    <p:projects>
      <p:name> SoSCP </p:name>
      ….
    </p:project>
</p:projects>
</collaboration>
```

Well formed XML : rules to respect

INSA

- File extension is **.xml**
- XML declaration
- A root element
- Each element has an opening and end markup
- For each sub-element, respect the open and close orders
- Attributes are inside the open markup

```
<?xml version ="1.0" encoding="iso-8859-1" ?>

<laboratory>

    <name> LAAS </name>

    <town> Toulouse </town>

    <specialties>
        <specialty> Informatique </specialty>
        <specialty> Automatique </specialty>
    <specialties>

    <tutelle nom="CNRS">

</laboratory>
```

Exercise (1) : XML

INSA

We need to store configuration data for applications.

Each application has a name.

We store data on the development process : the used test library, the quality code, and the versioning tools.

Besides, we store information on the deployment server : the port number and the host URL.

Exercise (1) : XML solution

INSA

```
<?xml version ="1.0" encoding="iso-8859-1"?>
<!-- First solution-->
<configuration>
  <application>
    <name> Application1 </name>
    <development>
      <testLibrary>JUnit</testLibrary>
      <quality>SonareQube</quality>
      <versioning>Git</versioning>
    </development>
    <server>
      <port>8080</port>
      <host>http://localhost</host>
    </server>
  </application>
  <application>
    <name> Application2 </name>
    …
  </application>
</configuration>
```

Exercise (1) : XML solution bis

INSA

```
<?xml version ="1.0" encoding="iso-8859-1"?>
<!-- Second possible solution-->
<configuration>
  <applications>
    <application>
      <name> Application1 </name>
      <development>
        <testLibrary>JUnit</testLibrary>
        <quality>SonareQube</quality>
        <versioning>Git</versioning>
      </development>
      <server>
        <port>8080</port>
        <host>http://localhost</host>
      </server>
    <application>
  </applications>
</configuration>
```

For or against XML

INSA

- XML is a standard
- Several tools support XML
- Allows to well structure data
- Relies on strict structuration rules

For or against XML

INSA

- XML is a standard
- Several tools support XML
- Allows to well structure data
- Relies on strict structuration rules

- It is very heavy (verbose)
- Limited in restricted bandwidth environment
- Does not allow to distinguish data types (int, short, …etc)

Other languages more simpler have been proposed as an alternative

INSA

JSON (JavaScript Object Notation)

JSON

INSA

- Light data format
- Very simple to write and to read
- Supports data types
- Supported by all modern languages
- A file with the extension .json

JSON

- Data stored in a set of value-key pairs
- The key and the value must be within “ “

```
{
  “key”: “value”,
  “key”: “value”
}
```

JSON

- Data stored in a set of value-key pairs
- The key and the value must be within “ “
- Data types :
  - String

```
{
   “key”: “value”,
   “key”: “value”
}
```

```
{
   “name”: “LAAS”,
   “town”: “Toulouse”
}
```

JSON

- Data stored in a set of value-key pairs
- The key and the value must be within “ “
- Data types :
  - String
  - Number

```
{
   “key”: “value”,
   “key”: “value”
}
```

```
{
   “name”: “LAAS”,
   “town”: “Toulouse”,
    “effectif”:700
}
```

JSON

- Data stored in a set of value-key pairs
- The key and the value must be within “ “
- Data types :
  - String
  - Number
  - Boolean (true/false)

```
{
  “key”: “value”,
  “key”: “value”
}
```

```
{
  “name”: “LAAS”,
  “town”: “Toulouse”,
   “effectif”:700,
   ”isPublic”:true
}
```

JSON

INSA

- Data stored in a set of value-key pairs
- The key and the value must be within “ “
- Data types :
  - String
  - Number
  - Boolean(true/false)
  - Array

```
{
   “key”: “value”,
   “key”: “value”
}
```

```
{
   “name”: “LAAS”,
   “town”: “Toulouse”,
    “effectif”:700,
    ”isPublic”:true,
    “speciality”:[“computer science”, “automatic”]
}
```

JSON

- Data stored in a set of value-key pairs
- The key and the value must be within “ “
- Data types :
  - String
  - Number
  - Boolean(true/false)
  - Array
  - Object

```
{
  “key”: “value”,
  “key”: “value”
}
```

```
{
  “name”: “LAAS”,
  “town”: “Toulouse”,
   “effectif”:700,
   ”isPublic”:true,
   “speciality”:[“computer science”, “automatic”],
    “departments”: [
        {
          “acronyme”: ”RISC”,
          ”name” : ”Reseau, Informatique
        },
        {
          “acronyme”: ”DO”,
          ”name” : ”Decision et Informatique”
        }
    ]
   }
```

JSON

- Data stored in a set of value-key pairs
- The key and the value must be within “ “
- Data types :
  - String
  - Number
  - Boolean(true/false)
  - Array
  - Object
  - null

```
{
  “key”: “value”,
  “key”: “value”
}
```

```
{
  “name”: “LAAS”,
  “town”: “Toulouse”,
   “effectif”:700,
   ”isPublic”:true,
   “speciality”:[“computer science”, “automatic”],
    “departments”: [
        {
          “acronyme”: ”RISC”,
          ”name” : ”Reseaux, Informatique”
        },
        {
          “acronyme”: ”DO”,
          ”name” : ” Décision et Optimisation
        }
    ],
    mainPartner: null
    }
```

Exercise (2) : JSON

INSA

Let as consider the first exercise. Give a JSON document that structure the following data.

We need to store configuration data for applications.

Each application has a name.

We store data on the development process : the used test library, the quality, and the versioning tools.

Besides, we store information on the deployment server : the port number and the host URL.

Solution: JSON

INSA

```
{
   "Configuration":[
    {"application":[
       {
          "name": "Application1",
          "development":[
           {
            "testLibrary":"Junit",
            "quality":"SonareQube",
            "versionning":"git"
           }],
           "server":[{
            "port":8080,
            "host":"http://localhost"
           }]
       }]
    }]
}
```

```
<?xml version ="1.0" encoding="UTF-8" ?>
<!-- First solution-->
<configuration>
  <application>
     <name> Application1 </name>
    <development>
       <testLibrary>JUnit</testLibrary>
       <quality>SonareQube</quality>
       <versioning>Git</versioning>
    </development>
     <server>
       <port>8080</port>
       <host>http://localhost</host>
     </server>
  </application>
</configuration>
```

Solution: JSON bis

INSA

```
{
  "Configuration":[
   {"application":[
     {
       "name": "Application1",
       "development":[
        {
         "testLibrary":"Junit",
         "quality":"SonareQube",
         "versionning":"git"
        }],
        "server":[{
         "port":8080,
         "host":"http://localhost"
        }]
     }]
   }]
}
```

```
{
  "Configurations":[
    {
      "name": "Application1",
      "development":
        {
          "testLibrary":"Junit",
          "quality":"SonareQube",
          "versionning":"git"
        },
        "server":
        {
          "port":"8080",
          "host": "http://localhost"
        }
    }
  ]
}
```

```
{
  "Configurations":[
    {
      "name": "Application1",
      "development":
        {
          "testLibrary":"Junit",
          "quality":"SonareQube",
          "versionning":"git"
        },
        "server":
         {
           "port":8080,
           "host": "http://localhost"
         }
    },
    {
      "name": "Application2",
      "development":
        {
          "testLibrary":"Junit",
          "quality":"SonareQube",
          "versionning":"git"
        },
        "server":
         {
           "port":8088,
           "host": "http://localhost"
         }
    }
  ]
}
```

Yaml (Yet Another Markup Language/ Yaml Ain’t Markup Language)

Yaml

INSA

- YAML is a data serialization language
- Widely used language for configuration files
- Very simple to read and to write
- Supports several data types
- A file with extension .yaml

Yaml

INSA

- Stored data as pairs of key, value
- Uses line separation and **indentation** (mandatory)

```
---
  nom: LAAS
  ville: Toulouse
```

Yaml

INSA

- Stored data as pairs of key, value
- Uses line separation and indentation (mandatory)
- Considers several data type (as JSON)
  - String (“ “, ’ ‘, or without brackets)

```
---
  nom: LAAS
  ville: Toulouse
```

Yaml

INSA

- Stored data as pairs of key, value
- Uses line separation and indentation (mandatory)
- Considers several data type (as JSON)
  - String (“ “, ’ ‘, or without brackets)
  - Boolean (true/false, yes/no, on/off)

```
---
  nom: LAAS
  ville: Toulouse
  isLaboratory: true
```

Yaml

INSA

- Stored data as pairs of key, value
- Uses line separation and indentation (mandatory)
- Considers several data type (as JSON)
  - String (“ “, ’ ‘, or without brackets)
  - Boolean (true/false, yes/no, on/off)
  - Number

```
---
  nom: LAAS
  ville: Toulouse
  isLaboratory: true
  year: 1959
```

Yaml

- Stored data as pairs of key, value
- Uses line separation and indentation (mandatory)
- Considers several data type (as JSON)
  - String (“ “, ’ ‘, or without brackets)
  - Boolean (true/false, yes/no, on/off)
  - Number
  - List

```
---
  nom: LAAS
  ville: Toulouse
  isLaboratory : true
  year : 1959
  specialities:
    - computer science
    - automatic
```

```
---
  nom: LAAS
  ville: Toulouse
  isLaboratory : true
  year : 1959
  specialities: [computer science, automatic]
```

```
---
  nom: LAAS
  ville: Toulouse
  isLaboratory : true
  year : 1959
  specialities: computer science, automatic
```

Yaml

INSA

- Stored data as pairs of key, value
- Uses line separation and indentation (mandatory)
- Considers several data type (as JSON)
  - String (“ “, ’ ‘, or without brackets)
  - Boolean (true/false, yes/no, on/off)
  - Number
  - List
  - Object

```
---
  nom: LAAS
  ville: Toulouse
  isLaboratory : TRUE
  year : 1959
  specialities:
     - computer science
     - automatic
  departments :
     - acronyme : RISC
       name : Reseau Informatique
     - acronyme : DO
       name: Decision et optimisation
```

YAML vs JSON vs XML

INSA

YAML

```
---
laboratories:
  - nom: LAAS
    ville: Toulouse
    specialities:
     - comp. Science
     - automatic
  - nom: IRIT
    ville: Toulouse
    specialities:
      - Comp. Science
```

JSON

```
{
   "laboratories" :[
    {
      "name": "LAAS",
      "town": "Toulouse",
      "speciality": [
          "computer science",
          "automatic"
       ]
    },
    {
      "name": "IRIT",
      "town": "Toulouse",
      "speciality": [
          "computer science"
       ]
     }
  ]
}
```

XML

```
<?xml version =" 1.0 " encoding="UTF-8"?>
<laboratories>
   <laboratory>
     <name> LAAS </name>
     <town> Toulouse </town>
     <specialties>
          <specialty> Informatique </specialty>
          <specialty> Automatique </specialty>
     </specialties>
  </laboratory>
<laboratory>
     <name> IRIT </name>
     <town> Toulouse </ town>
     <specialties>
          <specialty> Informatique </specialty>
     </specialties>
  </laboratory>
</laboratories>
```

Tools

INSA

- XML validator : https://www.xmlvalidation.com/
- JSON validator : https://jsonformatter.curiousconcept.com/#
- Yaml validator : onlineyamltools.com/edit-yaml

Exercise (3) : YAML

INSA

Let as consider again the previous exercise. Give a YAML document that structure the following data.

We need to store configuration data for applications.

Each application has a name.

We store data on the development process : the used test library, the quality, and the versioning tools.

Besides, we store information on the deployment server : the port number and the host URL.

Solution: YAML

INSA

```
---

Configurations:
  - name: appli1
    developement:
        testLibrary: Junit
        quality: SonareQube
        versionning: git
    server:
        port : 8080
        host : http://localhost
  - name: appli2
    developement:
        testLibrary: Junit
        quality: SonareQube
        versionning: git
    server:
        port : 8082
        host : http://localhost
```


INSA
INSTITUT NATIONAL
DES SCIENCES
APPLIQUÉES
TOULOUSE