

Software Engineering Tools

SYSC3020

The JOEL test

1. Do you use source control?
2. Can you make a build in one step?
3. Do you make daily builds?
4. Do you have a bug database?
5. Do you fix bugs before writing new code?
6. Do you have an up-to-date schedule?
7. Do you have a spec?
8. Do programmers have quiet working conditions?
9. Do you use the best tools money can buy?
10. Do you have testers?
11. Do new candidates write code during their interview?
12. Do you do hallway usability testing?

The JOEL test

TOOLS

1. **Do you use source control?**
2. **Can you make a build in one step?**
3. Do you make daily builds?
4. **Do you have a bug database?**
5. Do you fix bugs before writing new code?
6. Do you have an up-to-date schedule?
7. Do you have a spec?
8. Do programmers have quiet working conditions?
9. **Do you use the best tools money can buy?**
10. Do you have testers?
11. Do new candidates write code during their interview?
12. Do you do hallway usability testing?

Software Engineering Tools

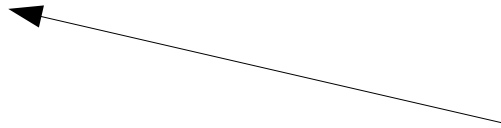
- Modeling tools
- IDEs
- Version Control
- Build scripts
- Testing tools
- Code analysis
- Bug tracking
- The cloud
- Command line

Software Engineering Tools

- Modeling tools
- IDEs
- Version Control
- Build scripts
- Testing tools
- Code analysis
- Bug tracking
- The cloud
- Command line

AUTOMATION

Generate diagrams, code,
documentation



Software Engineering Tools

- Modeling tools
- IDEs
- Version Control
- Build scripts
- Testing tools
- Code analysis
- Bug tracking
- The cloud
- Command line

AUTOMATION

Compile as you type,
Interface with other tools



Software Engineering Tools

- Modeling tools

- IDEs

- Version Control

- Build scripts

- Testing tools

- Code analysis

- Bug tracking

- The cloud

- Command line

AUTOMATION

Save, automatic rollback

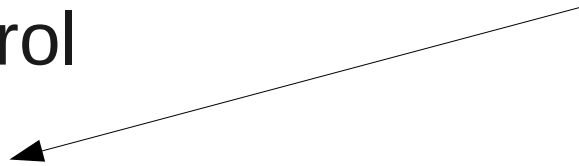


Software Engineering Tools

- Modeling tools
- IDEs
- Version Control
- Build scripts
- Testing tools
- Code analysis
- Bug tracking
- The cloud
- Command line

AUTOMATION

Checkout from VC, compile,
deploy, build final product



Software Engineering Tools

- Modeling tools
- IDEs
- Version Control
- Build scripts
- Testing tools
- Code analysis
- Bug tracking
- The cloud
- Command line

AUTOMATION

Checkout from VC, compile,
deploy, build final product

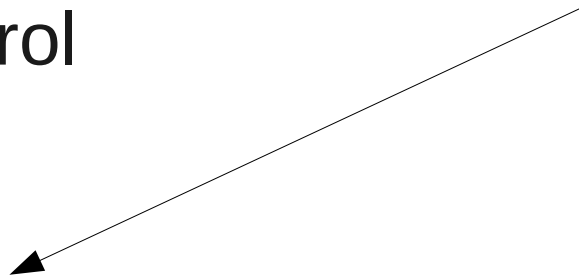


Software Engineering Tools

- Modeling tools
- IDEs
- Version Control
- Build scripts
- Testing tools
- Code analysis
- Bug tracking
- The cloud
- Command line

AUTOMATION

Automatically run a series of tests, generate report

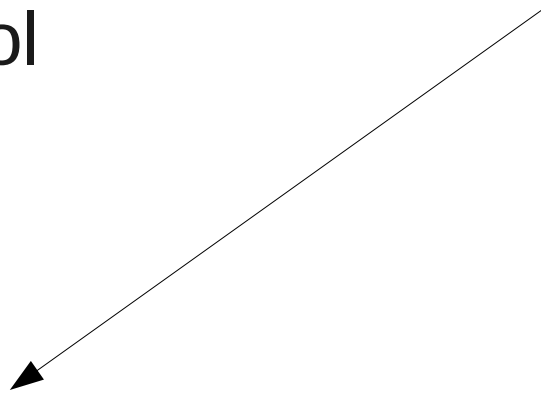


Software Engineering Tools

- Modeling tools
- IDEs
- Version Control
- Build scripts
- Testing tools
- Code analysis
- Bug tracking
- The cloud
- Command line

AUTOMATION

Automate code reviews

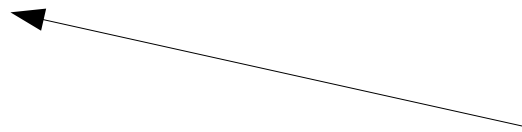


Software Engineering Tools

- Modeling tools
- IDEs
- Version Control
- Build scripts
- Testing tools
- Code analysis
- Bug tracking
- The cloud
- Command line

TRACKING

Never lose code



Software Engineering Tools

- Modeling tools
- IDEs
- Version Control
- Build scripts
- Testing tools
- Code analysis
- Bug tracking
- The cloud
- Command line

TRACKING

Know what the current bugs are



Software Engineering Tools

- Modeling tools
- IDEs
- Version Control
- Build scripts
- Testing tools
- Code analysis
- Bug tracking
- The cloud
- Command line

EFFICIENCY

Infrastructure as a service

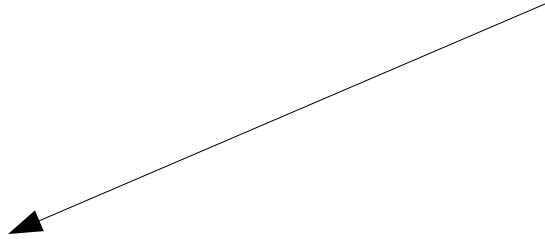


Software Engineering Tools

- Modeling tools
- IDEs
- Version Control
- Build scripts
- Testing tools
- Code analysis
- Bug tracking
- The cloud
- Command line

EFFICIENCY

Your grandma uses GUI.
Experts use command line.



Modeling Tools

UML editors: Rational Rose...

- Forward Engineering: Generate code from UML
- Documentation: populates template specification document
- Reverse Engineering : code to models
 - class diagram
 - sequence diagram

Modeling Tools

UML editors: Rational Rose...

- Forward Engineering: Generate code from UML
- Documentation: populates template specification document
- Reverse Engineering : code to models
 - class diagram
 - sequence diagram

Note: **code-level**, not **analysis** diagrams.

IDEs

- Language specific:
 - Java: Eclipse, Netbeans, blueJ...
- Text editor + ... :
 - Syntax highlighting
 - Automate tasks: refactoring...
- Interface with other tools:
 - run code
 - debug
 - compiler
 - version control
 - unit tests
 - ... (plugins?)

IDEs

- Language specific:
 - Java: Eclipse, Netbeans, blueJ...
- Interface with other tools:
 - run code
 - debug
 - compiler
 - version control
 - unit tests
 - ... (plugins?)
- Built-in features
 - Compiler
 - Refactoring
 - ...

Version Control

- Maintain history
- Manage multiple versions
- Easy revert changes
- Team collaboration

Version Control

- Centralized
 - CVS, SVN...
 - centralized repository (server)
- Decentralized
 - Git, Mercurial, Bazaar...
 - Each user has complete repository
- Online Code “forges”:
 - Sourceforge, Google code, Github, Gitorious...

Version Control

- Main tasks:

Version Control

- Main tasks:
 - Create repository



Initial
repository

Version Control

- Main tasks:
 - Create repository
 - Define access control



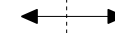
Bob ✓



Initial repository

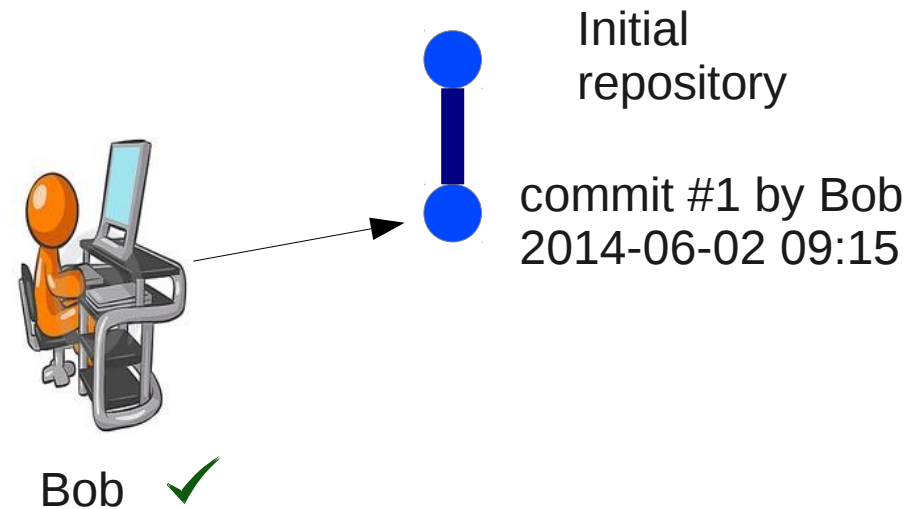
Bob's local workspace

Version Control repository



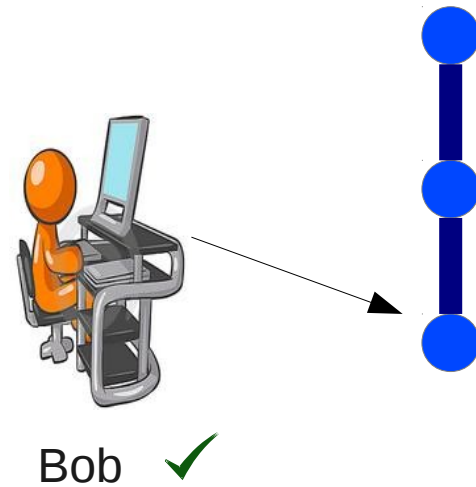
Version Control

- Main tasks:
 - Create repository
 - Define access control
 - Commit (to repo)



Version Control

- Main tasks:
 - Create repository
 - Define access control
 - Commit (to repo)

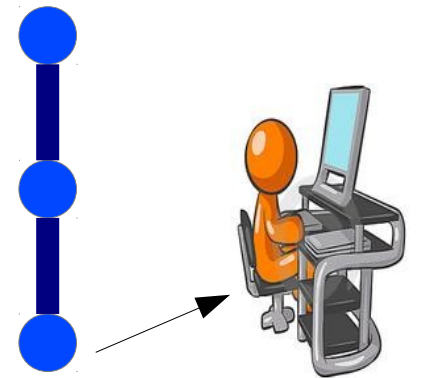


Version Control

- Main tasks:
 - Create repository
 - Define access control
 - Commit (to repo)
 - Update (from repo)



Bob ✓



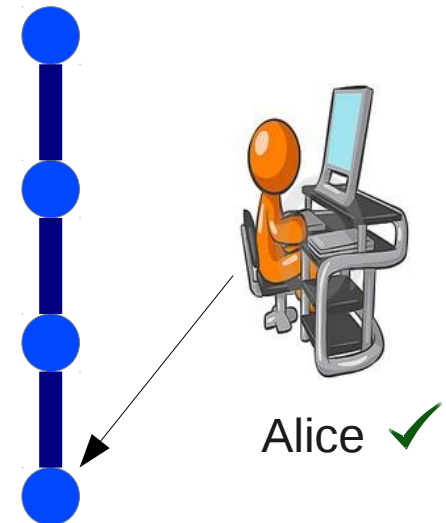
Alice ✓

Version Control

- Main tasks:
 - Create repository
 - Define access control
 - Commit (to repo)
 - Update (from repo)



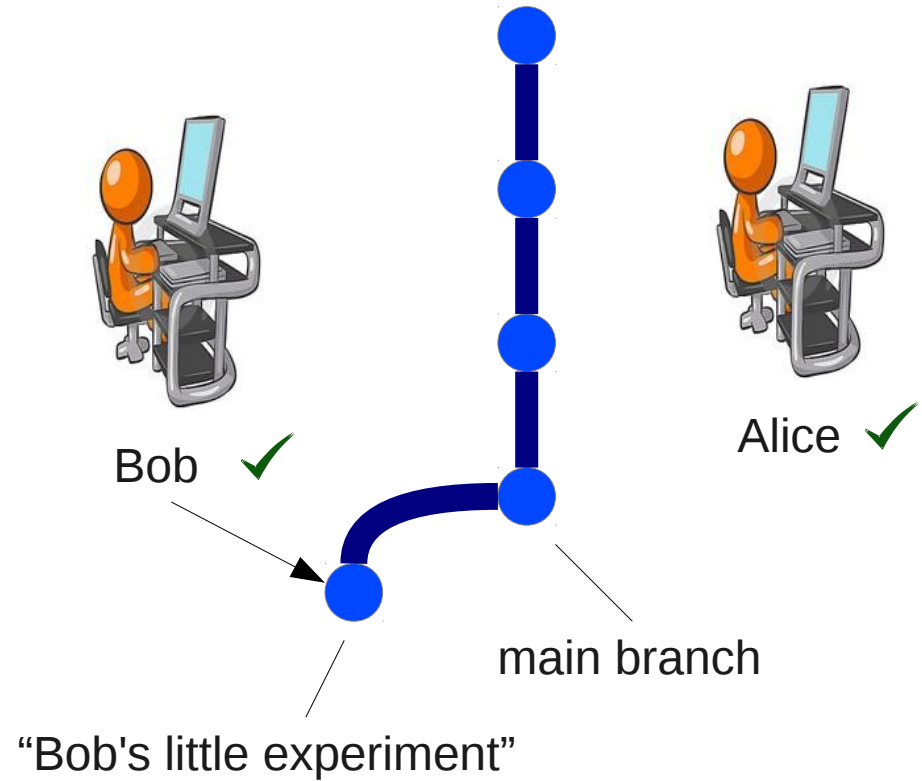
Bob ✓



Alice ✓

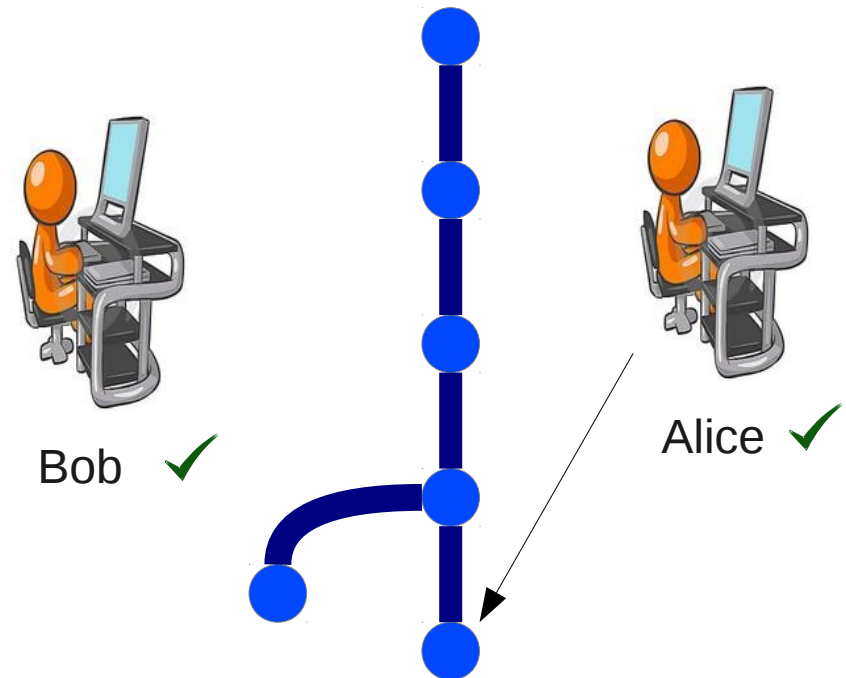
Version Control

- Main tasks:
 - Create repository
 - Define access control
 - Commit (to repo)
 - Update (from repo)
 - Branch



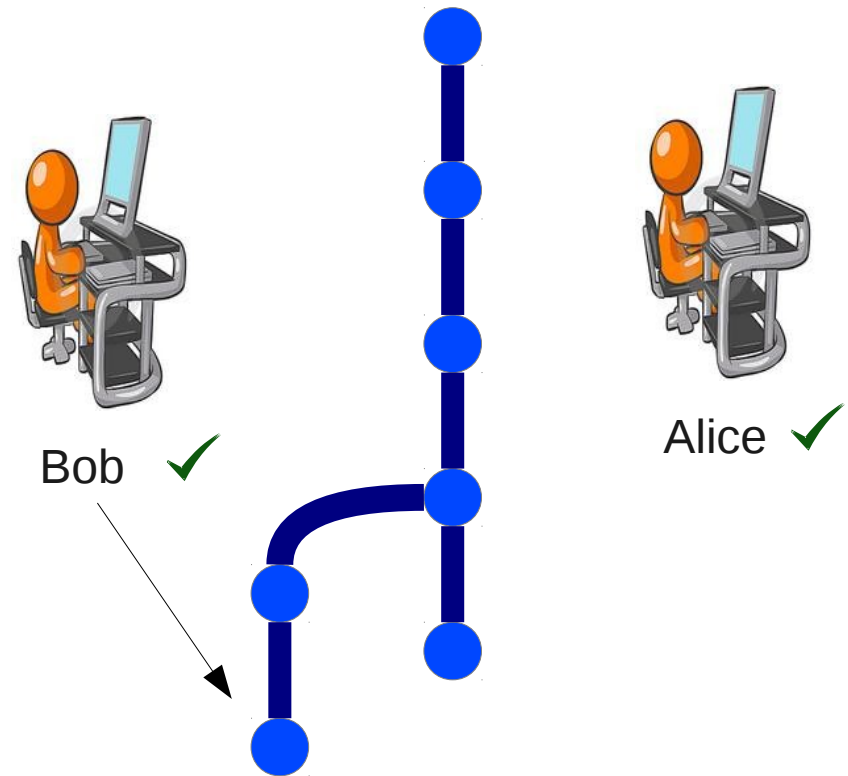
Version Control

- Main tasks:
 - Create repository
 - Define access control
 - Commit (to repo)
 - Update (from repo)
 - Branch



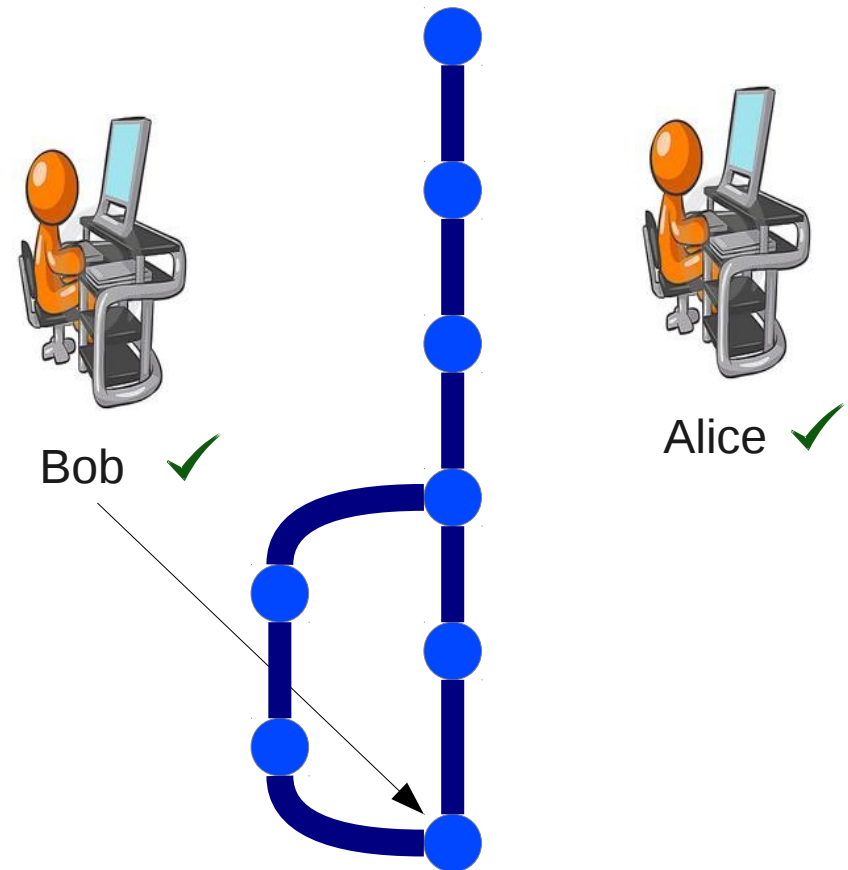
Version Control

- Main tasks:
 - Create repository
 - Define access control
 - Commit (to repo)
 - Update (from repo)
 - Branch



Version Control

- Main tasks:
 - Create repository
 - Define access control
 - Commit (to repo)
 - Update (from repo)
 - Branch
 - Merge

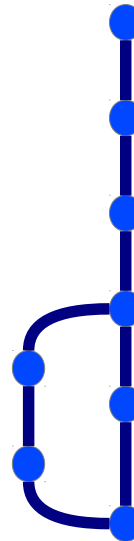


Version Control

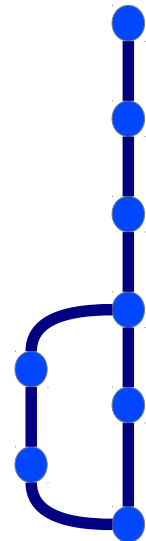
- Main tasks:
 - Create repository
 - Define access control
 - Commit (to repo)
 - Update (from repo)
 - Branch
 - Merge
- DVCS model:



Bob



Alice



Version Control

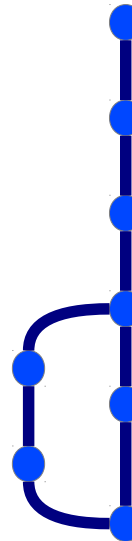
- Main tasks:
 - Create repository
 - Define access control
 - Commit (to repo)
 - Update (from repo)
 - Branch
 - Merge
- DVCS model:



Bob



Alice

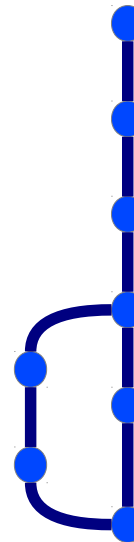


Version Control

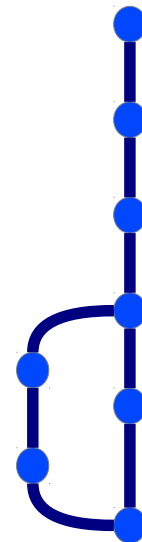
- Main tasks:
 - Create repository
 - Define access control
 - Commit (to repo)
 - Update (from repo)
 - Branch
 - Merge
- DVCS model:
 - Clone/fork



Bob



Alice



clone

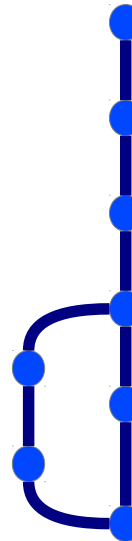


Version Control

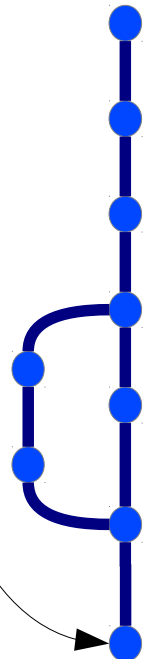
- Main tasks:
 - Create repository
 - Define access control
 - Commit (to repo)
 - Update (from repo)
 - Branch
 - Merge
- DVCS model:
 - Clone/fork
 - Local Commits



Bob



Alice



Version Control

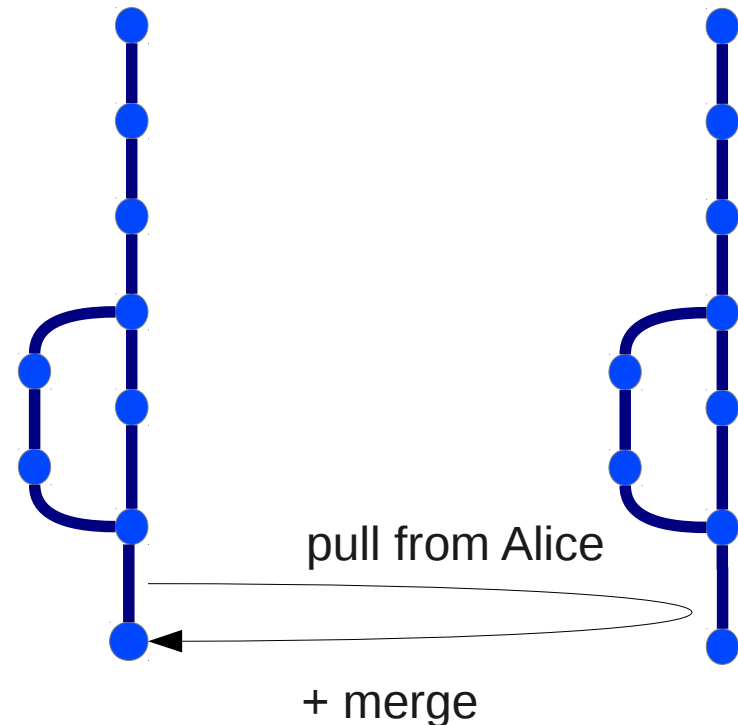
- Main tasks:
 - Create repository
 - Define access control
 - Commit (to repo)
 - Update (from repo)
 - Branch
 - Merge
- DVCS model:
 - Clone/fork
 - Local Commits
 - Pull



Bob



Alice

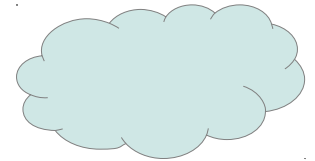
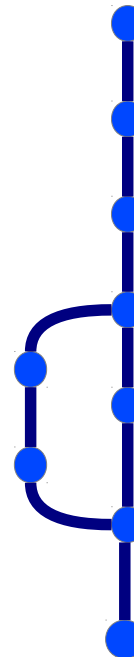


Version Control

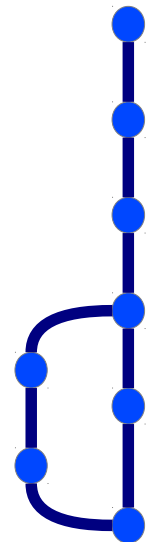
- Main tasks:
 - Create repository
 - Define access control
 - Commit (to repo)
 - Update (from repo)
 - Branch
 - Merge
- DVCS model:
 - Clone/fork
 - Local Commits
 - Pull
 - Push to remote repo



Bob



Bob's
github

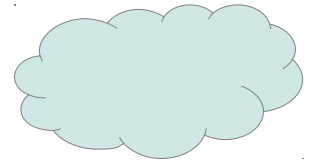


Version Control

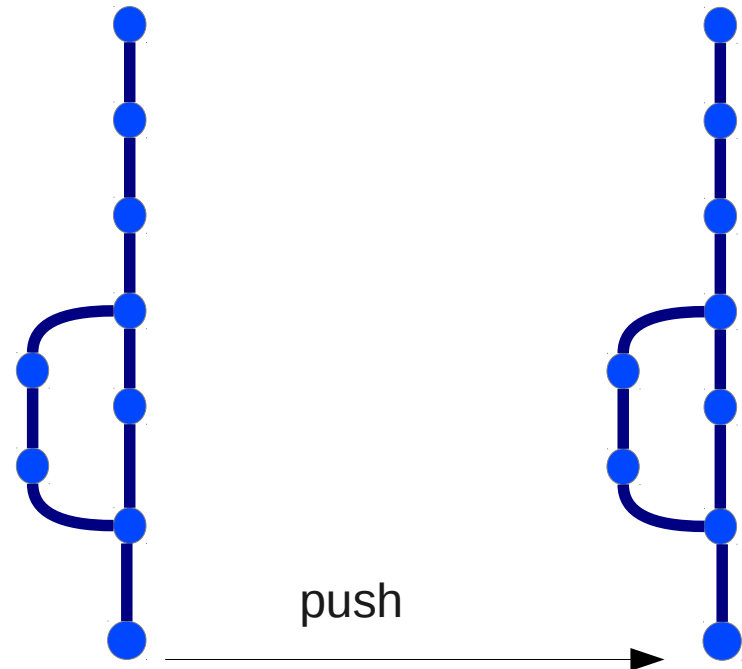
- Main tasks:
 - Create repository
 - Define access control
 - Commit (to repo)
 - Update (from repo)
 - Branch
 - Merge
- DVCS model:
 - Clone/fork
 - Local Commits
 - Pull
 - Push to remote repo



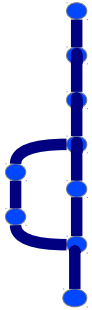
Bob



Bob's
github



Build automation

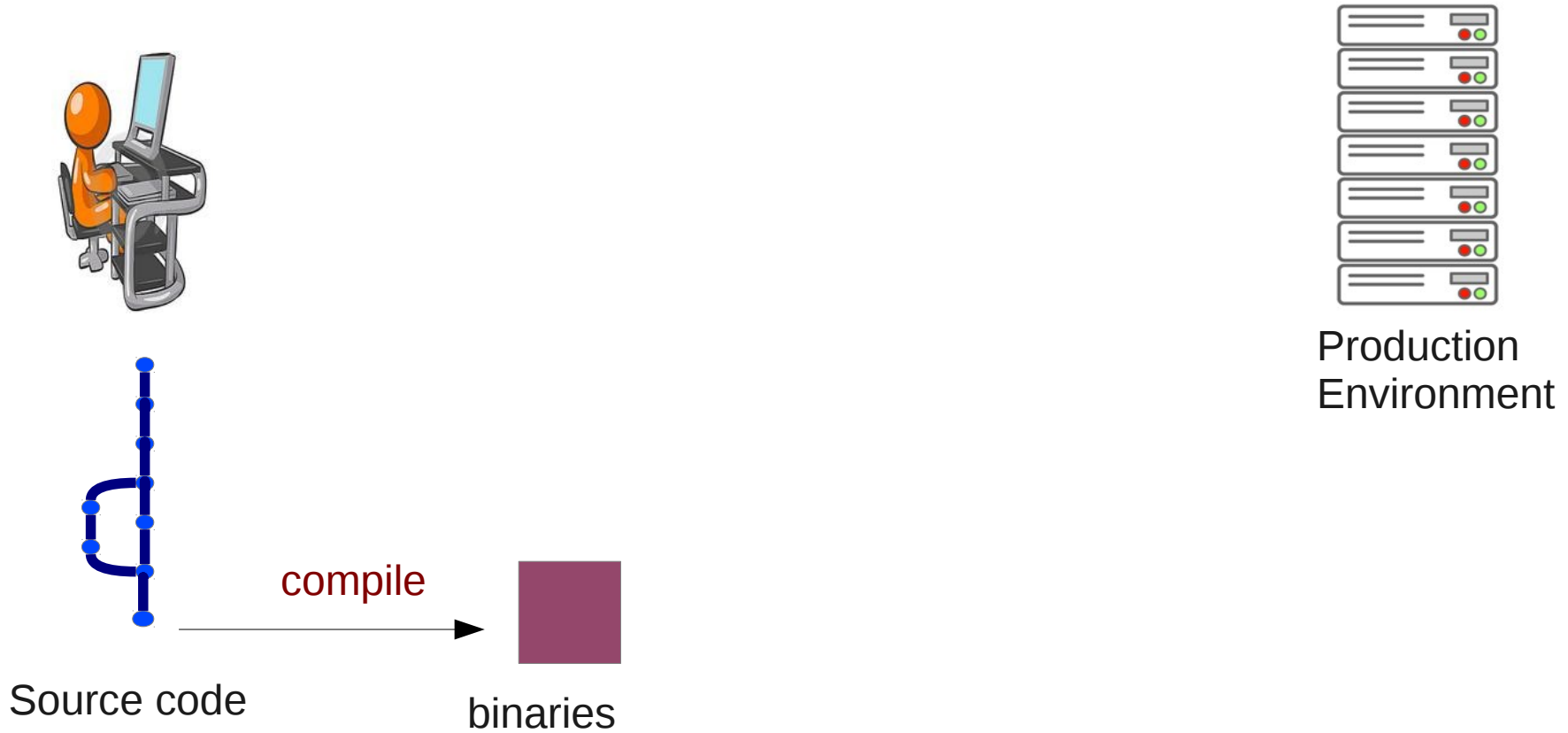


Source code

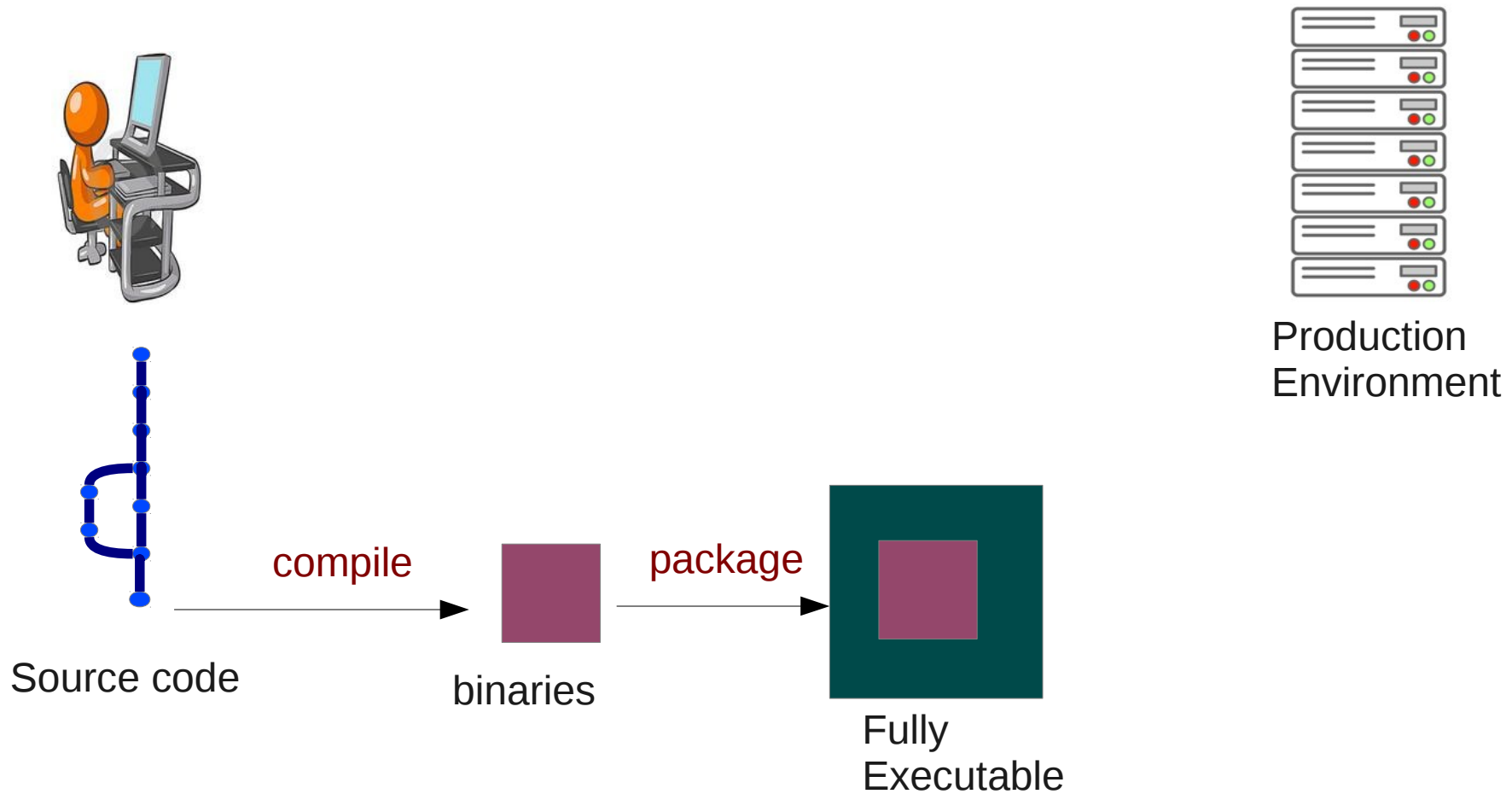


Production
Environment

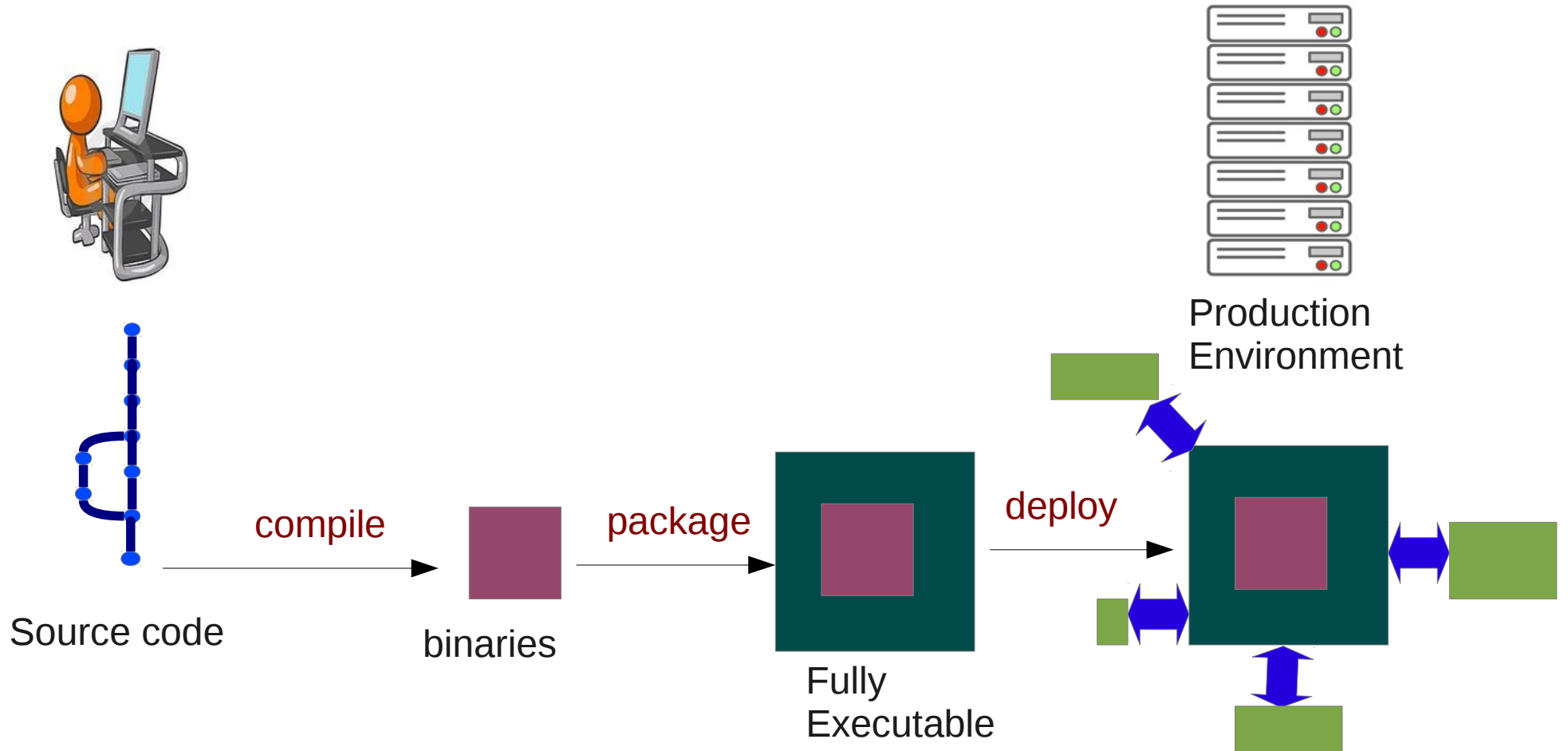
Build automation



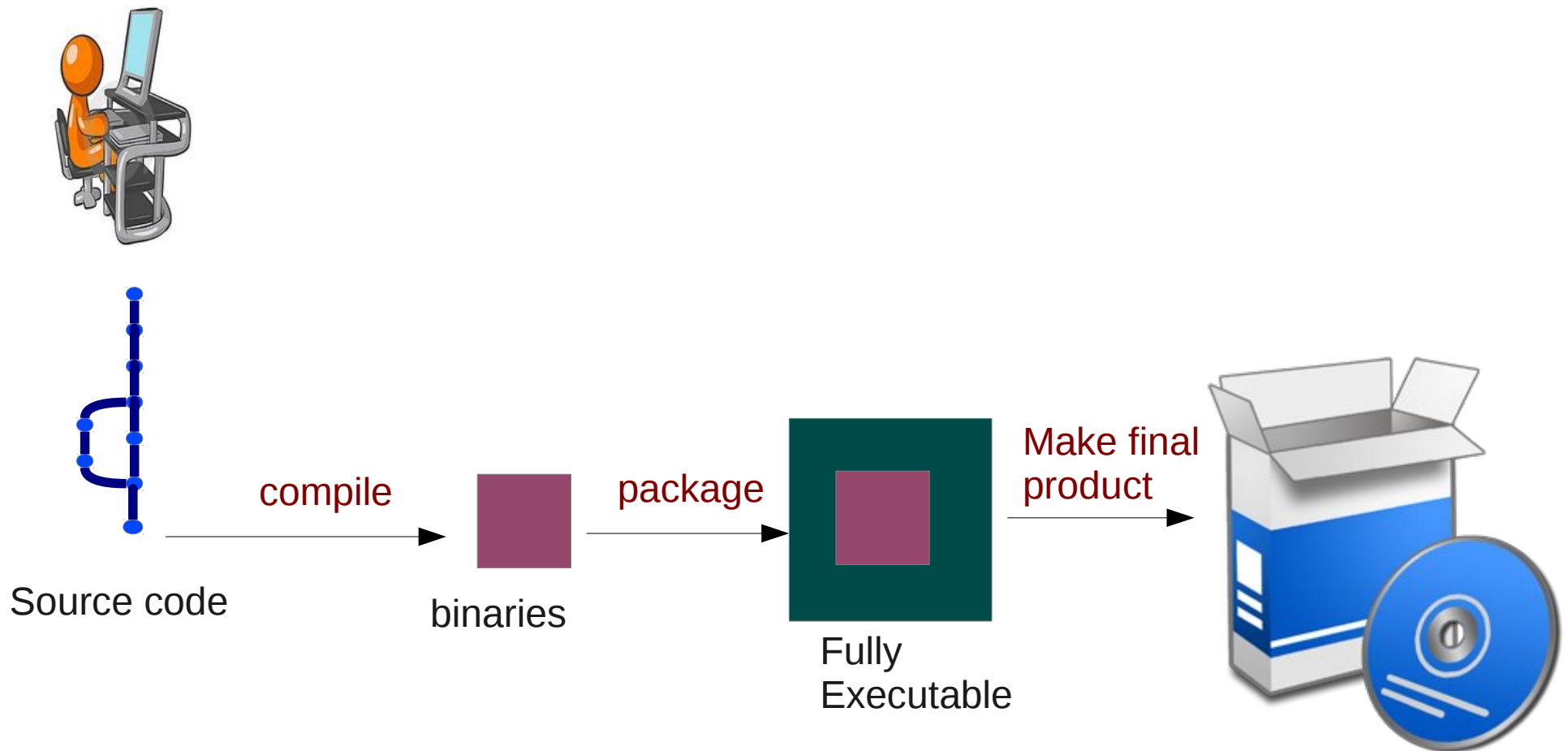
Build automation



Build automation



Build automation



Build automation

1. Do you use source control?
- 2. Can you make a build in one step?**
3. Do you make daily builds?
4. Do you have a bug database?
5. Do you fix bugs before writing new code?
6. Do you have an up-to-date schedule?
7. Do you have a spec?
8. Do programmers have quiet working conditions?
9. Do you use the best tools money can buy?
10. Do you have testers?
11. Do new candidates write code during their interview?
12. Do you do hallway usability testing?

Build automation

- Apache ant
- Apache Maven
- ...

```
<project name="MyProject" default="dist" basedir=". ">
  <description> simple example build file </description>
  <!-- set global properties for this build -->
  <property name="src" location="src"/>
  <property name="build" location="build"/>
  <property name="dist" location="dist"/>

  <target name="compile" description="compile the source " >
    <!-- Compile the java code from ${src} into ${build} -->
    <javac srcdir="${src}" destdir="${build}"/>
  </target>

  <target name="dist" depends="compile" description="generate the distribution" >
    <!-- Create the distribution directory -->
    <mkdir dir="${dist}/lib"/>
    <!-- Put everything in ${build} into the MyProject-${DSTAMP}.jar file -->
    <jar jarfile="${dist}/lib/MyProject-${DSTAMP}.jar" basedir="${build}"/>
  </target>
</project>
```

Software Engineering Tools

- Modeling tools
- IDEs
- Version Control
- Build scripts
- **Testing tools**
- **Code analysis**
- **Bug tracking**
- The cloud
- Command line

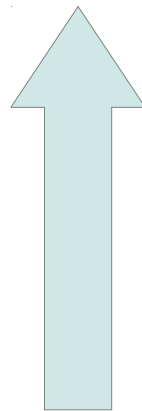


later

Cloud Computing

- x-AAS: x As A Service:


- Software
- Platform
- Infrastructure



high level

Low level

Cloud Computing

- IAAS: Infrastructure As A Service:
 - Software
 - Platform
 - Infrastructure  rent server space
and CPU
- Amazon Web Services
- ...

Cloud Computing

- PAAS: Platform As A Service:
- Software
- Platform ————— Deploy your application
to a fully functional server stack
- Infrastructure
Heroku, AWS...

Cloud Computing

- SAAS: Software As A Service:
- Software ————— Use online applications:
Basecamp
- Platform Google docs
- Infrastructure ...

The command line

Original Unix tool: Bourne Shell (sh)

Variations: bash, ksh, tcsh...

Unix, linux, Mac OS => sh

Windows => MS-DOS command line

- Faster than GUI
- Maintainable: easy to add commands
- Good for remote access/ embedded hardware
- **Automate** common tasks!

Web Site Architecture



Client

The diagram illustrates a two-tier web site architecture. It consists of two main components: a 'Client' layer at the top and a 'Server stack' layer below it. The 'Client' layer is represented by a light blue rectangular box. The 'Server stack' layer is represented by a larger light blue rectangular box. Both boxes have a thin black border. The text 'Client' is centered within the top box, and 'Server "stack"' is centered within the bottom box.

Server "stack"

Web Application Architecture

