Definition of done:

Working software in production

Thomas Sundberg

Think Code AB
http://www.thinkcode.se
@thomassundberg
thomas@thinkcode.se

Continuous deployment

MYNDIGHETEN FÖR
TILLGÄNGLIGA MEDIER

Why

do we want the software in production?

Write down as many reasons you can in 2 minutes

Customers want the service

We want feedback

Was this what you needed?

How

do you prefer to install software?

Write down as many good ways you can in 2 minutes

Packages for the OS

We had Red Hat

RPMs were a good fit

```
yum -y install package-name
```

Java web app

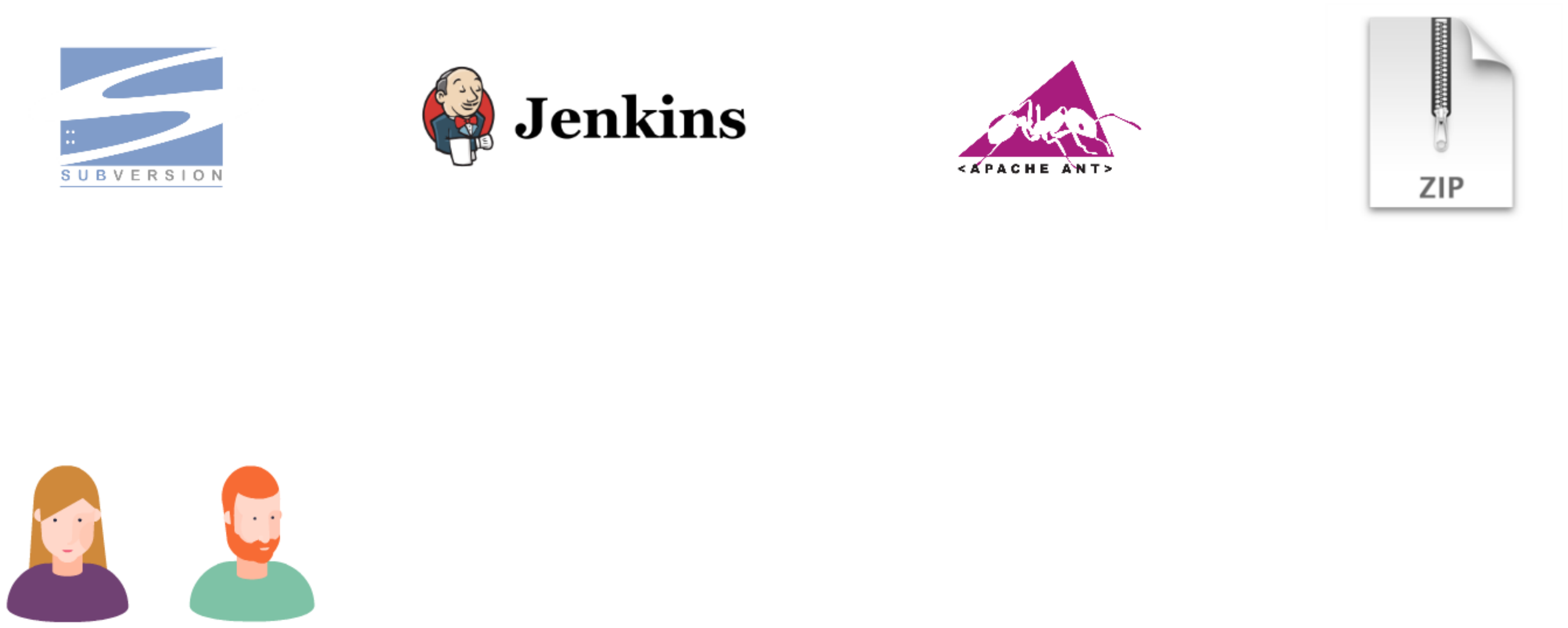
package a Java app as an RPM

Gradle

gradle-ospackage-plugin

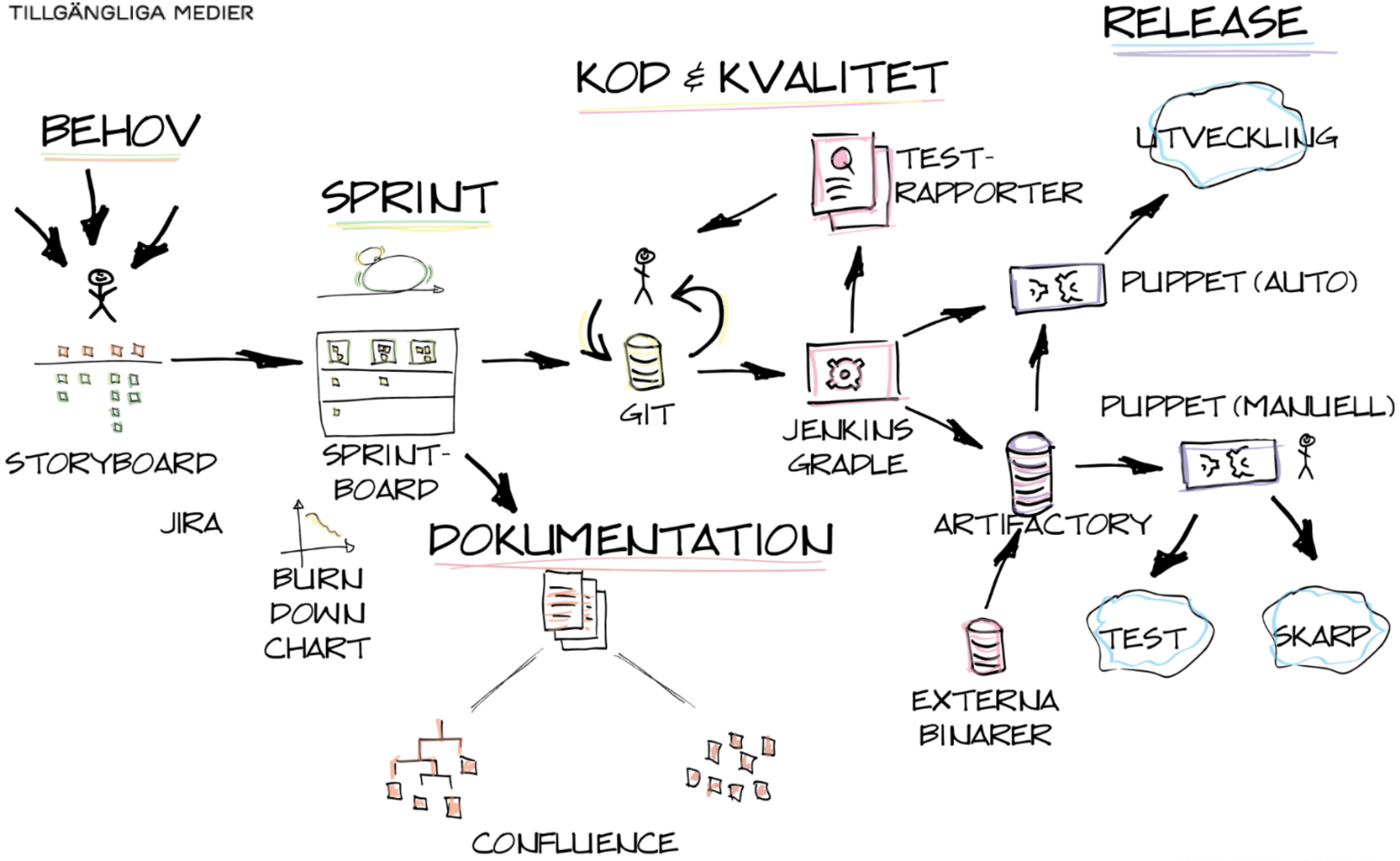
A Netflix project

Where did we start?

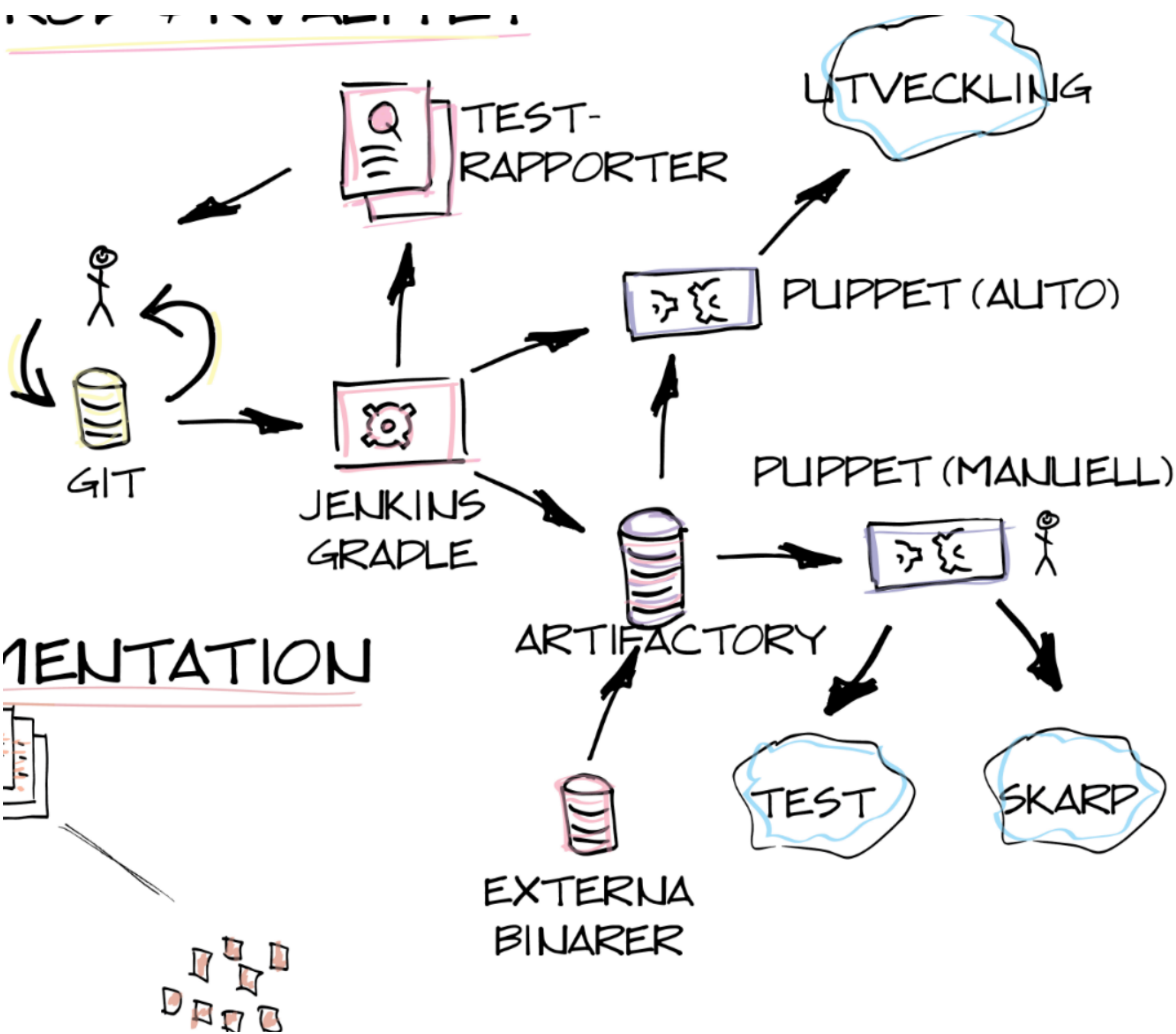

SUBVERSION
Jenkins
<APACHE ANT>
ZIP

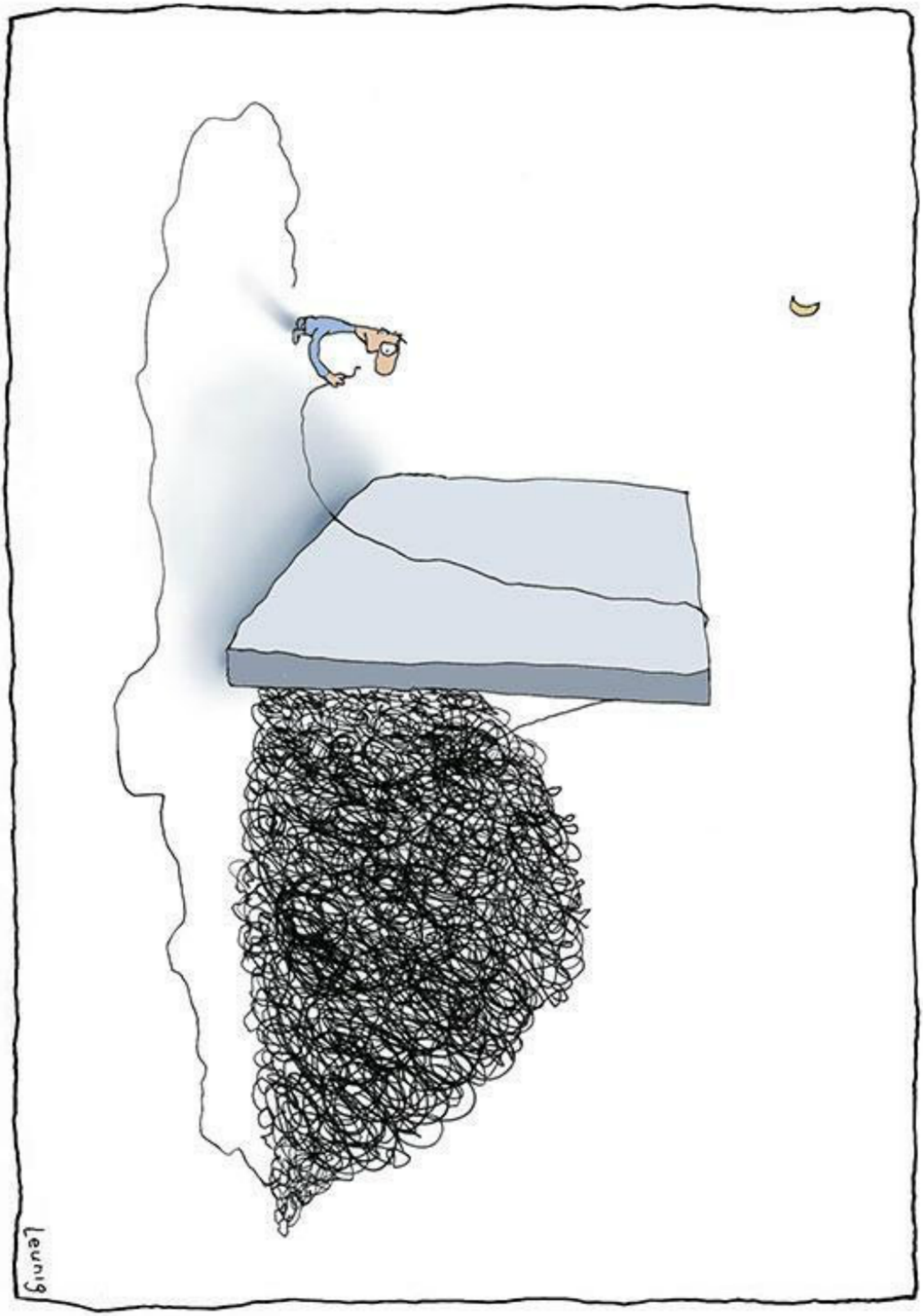
MYNDIGHETEN FÖR
TILLGÄNGLIGA MEDIER

UTVECKLING



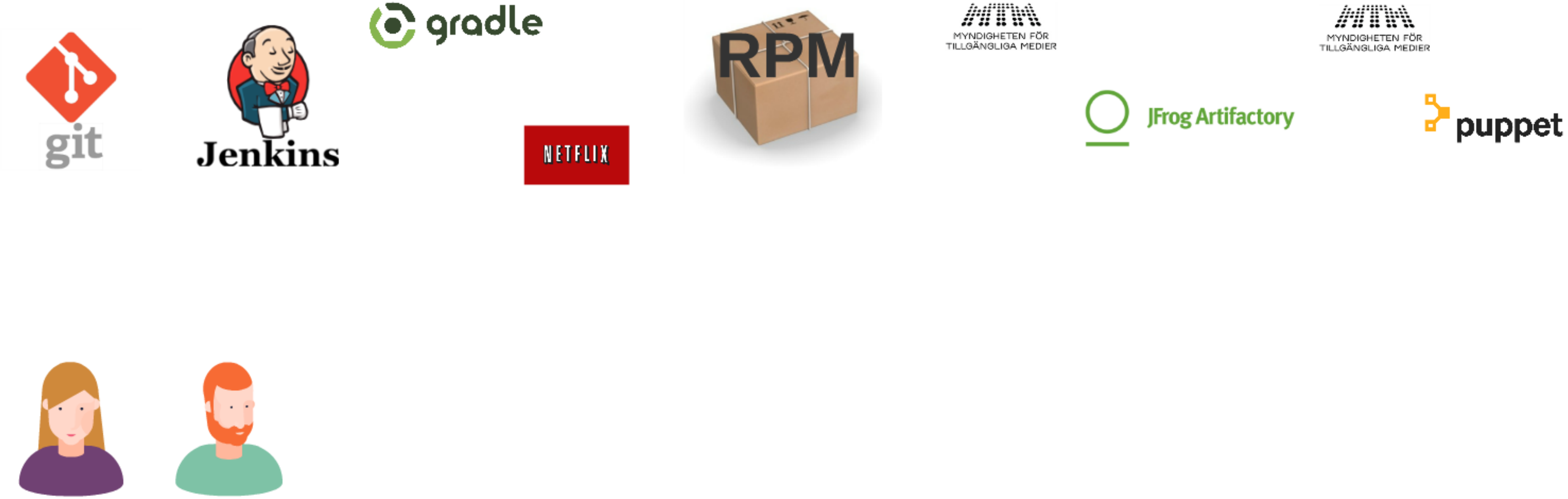
JACOB TARDELL 2014-09-25


TEST-
RAPPORTER
UTVECKLING
PUPPET (AUTO)
GIT
JENKINS
GRADLE
PUPPET (MANUELL)
MENTATION
ARTIFACTORY
TEST
SKARP
EXTERNA
BINARER

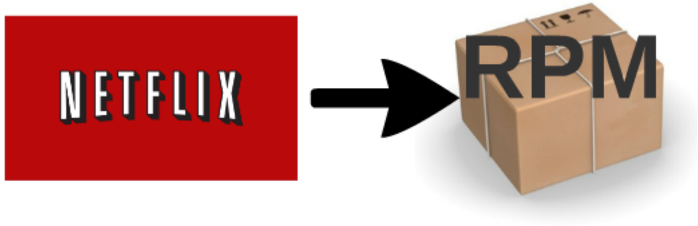

Leunig

Homak
professional

Build pipeline


git
Jenkins
gradle
NETFLIX
RPM
MYNDIGHETEN FÖR TILLGÄNGLIGA MEDIER
JFrog Artifactory
MYNDIGHETEN FÖR TILLGÄNGLIGA MEDIER
puppet

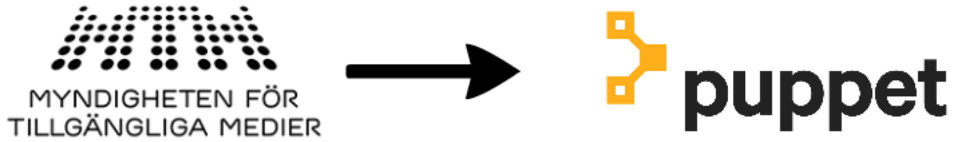
gradle



```
ospackage {
    packageName = project.name
    version = rpmVersion
    release = buildNumber
    os = LINUX
    user = "root"

    installUtils file('src/main/rpm/utils.sh')
    preInstall file('src/main/rpm/preInstall.sh')
    preUninstall file('src/main/rpm/preUninstall.sh')
    postInstall file('src/main/rpm/postInstall.sh')
    postUninstall file('src/main/rpm/postUninstall.sh')

    into '/opt/mtm-status'

    from('./build/libs') {
        into('/opt/mtm-status/lib')
        include '*all.jar'

        rename { String fileName ->
            fileName.replace('-' + version + '-', '')
        }

        rename { String fileName ->
            fileName.replace('SNAPSHOT', '')
        }

        fileMode = 0555
    }

    from('./src/main/scripts') {
        into('/opt/mtm-status/scripts')
        fileMode = 0755
    }

    link('/etc/init.d/mtm-status', '/opt/mtm-status/scripts/mtm-status.sh')
}
```

MYNDIGHETEN FÖR
TILLGÄNGLIGA MEDIER
JFrog Artifactory

```
artifactoryPublish {
    dependsOn jar, sourceJar, javadocJar
}
```



```
gradlePuppet {
    utvHosts = [’statusutv1.mtm.se’]
    testHosts = [’statustest1.mtm.se’]
    prodHosts = [’status1.mtm.se’]
}
```

Problems

No manager buy in

SNAPSHOT versions

Too many requests to Red Hat

SNAPSHOT not allowed in the RPM version number

Branches for the Puppet manifests

Tips and trix

Start small

Every improvement counts

$1.01^{365} = 37.8$

$1.0.1^{200} = 7.3$

Not bad while keeping a sustainable pace...

You will make mistakes

Learn or succeed!

Version control everything

The build server should be possible to replace in a few minutes.

Automate your testing

- Fast
- Reliable
  - False positives creates distrust

Practice

First in test

Then in production

Amateurs practice until they can make it, professionals until they cant fail.

A journey

Be prepared

Get a map

Set a direction

You will never
be done...

Perfect is not a place,
it is a direction!

Conclusion

Continuous deployment

Not a technical problem

People problem

- Changes are scary
- Managment doesn't see the benefit

References

Blog post:
http://www.thinkcode.se/blog/2015/12/23/continuous-deployment-a-case-study

Gradle plugins:
https://github.com/nebula-plugins/nebula-ospackage-plugin
https://github.com/mtmse/gradle-artifactory-rpm-plugin
https://github.com/mtmse/gradle-puppet-plugin

Definition of done:

Working software in production

Thomas Sundberg

Think Code AB
http://www.thinkcode.se
@thomassundberg
thomas@thinkcode.se