



Maven the Beautiful City

Healthy, Viable, and Productive
Build Infrastructures



What is Maven?

- Build tool
 - Similar to Ant but fundamentally different which we will discuss later
- Dependency management tool
 - Similar to Ivy, but not bolted on as an afterthought
- Site management tool

Maven provides sound build infrastructures

- Build infrastructures consist of:
 - Building
 - Testing
 - Documentation
 - Continuous Integration
 - Releasing
 - Provisioning

Infrastructure, Portland, Oregon and Maven

- Intentional infrastructure
 - You have to invest in your infrastructure to yield returns
 - It must be carefully planned, infrastructures don't just happen
- Long-term sustainability
 - The key is keeping people involved
 - sustainability by virtue versus sustainability by force
- Healthy growth in mind
 - Some constraints are necessary
- Promotion of community
 - Create opportunities for people to interact

Questions about infrastructure

- Do you think your infrastructure is wildly different than anyone else's?
- Is your infrastructure a competitive advantage?
- Would you like help with the most difficult infrastructure problems you might face?

Inherent utility

- The infrastructure should provide a lot to make complicated tasks easier
- When complicated tasks are made easy, there is more time for the bigger picture
- What's in the bigger picture?

Maven & Social Capital

- Term coined by Jane Jacobs (The Death and Life of Great American Cities)
- People's ability to work together in groups
- Creation of real communities using the idea of patterns and conscious planning
- Concrete example of the Maven community working to improve project metadata

Maven's Objectives

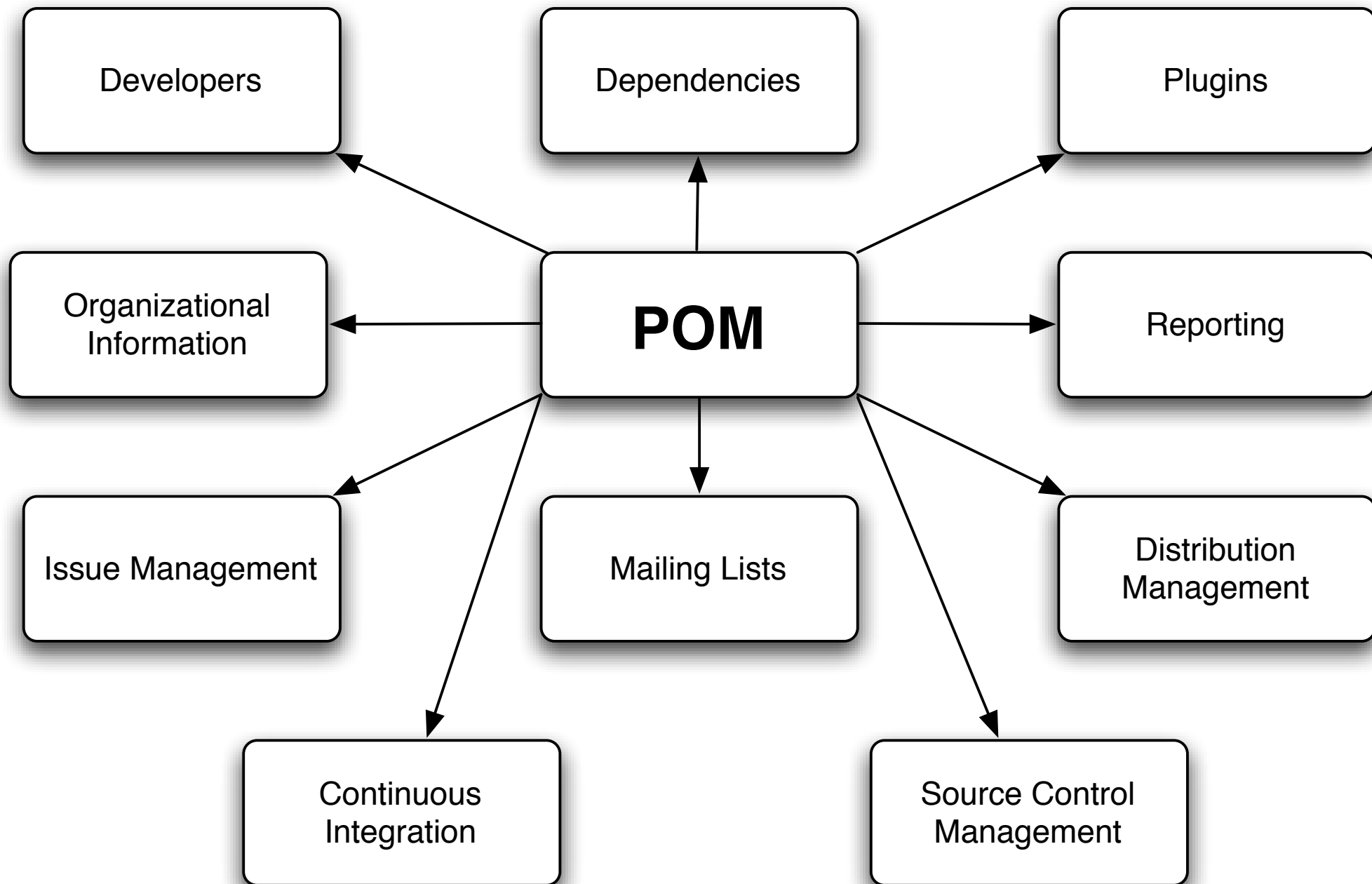
- Standards and Best Practices for build infrastructures
 - We use standard APIs and best practices when developing applications: Maven pushes this practice down to the level of infrastructure
- Provide a shared language for build infrastructure management
 - Patterns for build infrastructures in the spirit of “A Pattern Language” by Christopher Alexander
- Create healthy and robust build infrastructures that hold up to high degrees of flux

Maven's Principles

- Model driven development
- Convention over configuration
- Reuse and encapsulation of build logic
- Coherent organization of dependencies

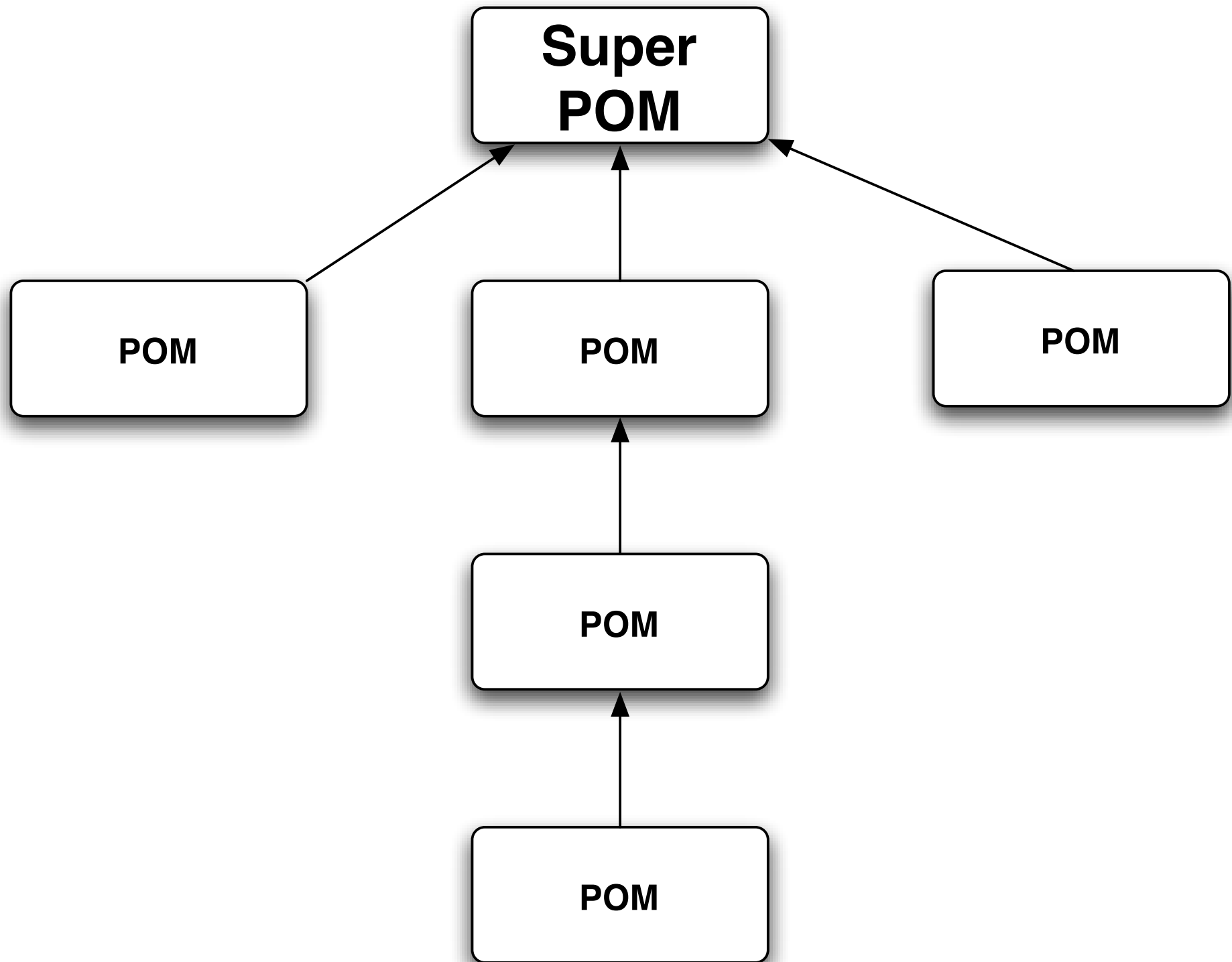
Model driven development

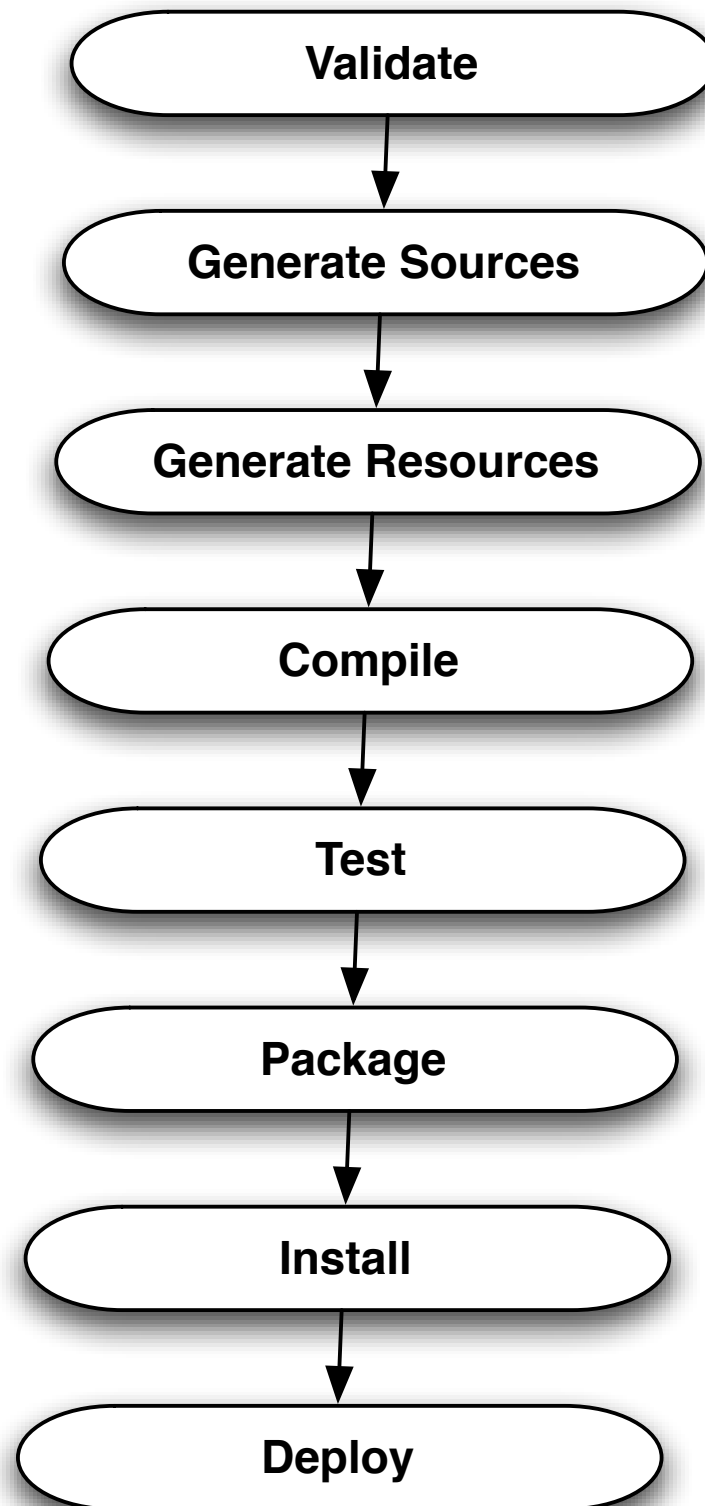
- Maven's project object model (POM)
- Maven's build life cycle



Simple Project Object Model

```
<project>
  <modelVersion>4.0.0</modelVersion>
  <groupId>com.iona.superapp</groupId>
  <artifactId>superapp</artifactId>
  <version>1.0-SNAPSHOT</version>
  <name>Maven Quick Start Archetype</name>
  <dependencies>
    <dependency>
      <groupId>junit</groupId>
      <artifactId>junit</artifactId>
      <version>3.8.1</version>
      <scope>test</scope>
    </dependency>
  </dependencies>
</project>
```





Convention over configuration

- Standard directory layout
- One primary artifact per build
- Standard naming conventions

Encapsulation and reuse of build logic

- All build logic is encapsulated in plugins
- Maven is a plugin execution framework
- Plugins can be applied to all Maven projects

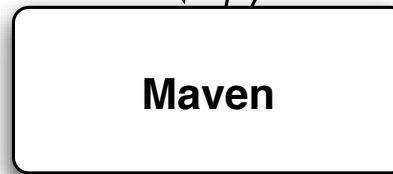
Coherent organization of dependencies

- Say what you need, not where or how to get it
 - Dependencies in Maven are requested in a declarative fashion
- Artifacts and Repositories
 - Remote repositories for the satisfaction of dependencies
 - Local repository is the developers personal analog to a set of remote repositories

Maven checks the local repository for the required dependencies



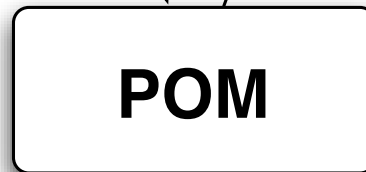
2



3

Maven retrieves the necessary dependencies from the remote repository

1



Maven gathers dependency information from the POM

What Maven Provides

- Coherence - An orderly, logical, and aesthetically consistent relation of parts
- Reusability - Reuse not only of software components but the best practices of an entire industry
- Agility - Maven allows easier integration and project straddling
- Maintainability - Maven projects are more maintainable as there are far fewer surprises

Benefits of Maven

- Relieves the burden of project and build maintenance
- Easy for new users to embrace Best Practices
- Focus on adding value to your applications
- Draw upon the community for solutions

Questions?

- Do have any questions about Maven?

Features of Maven 2.x

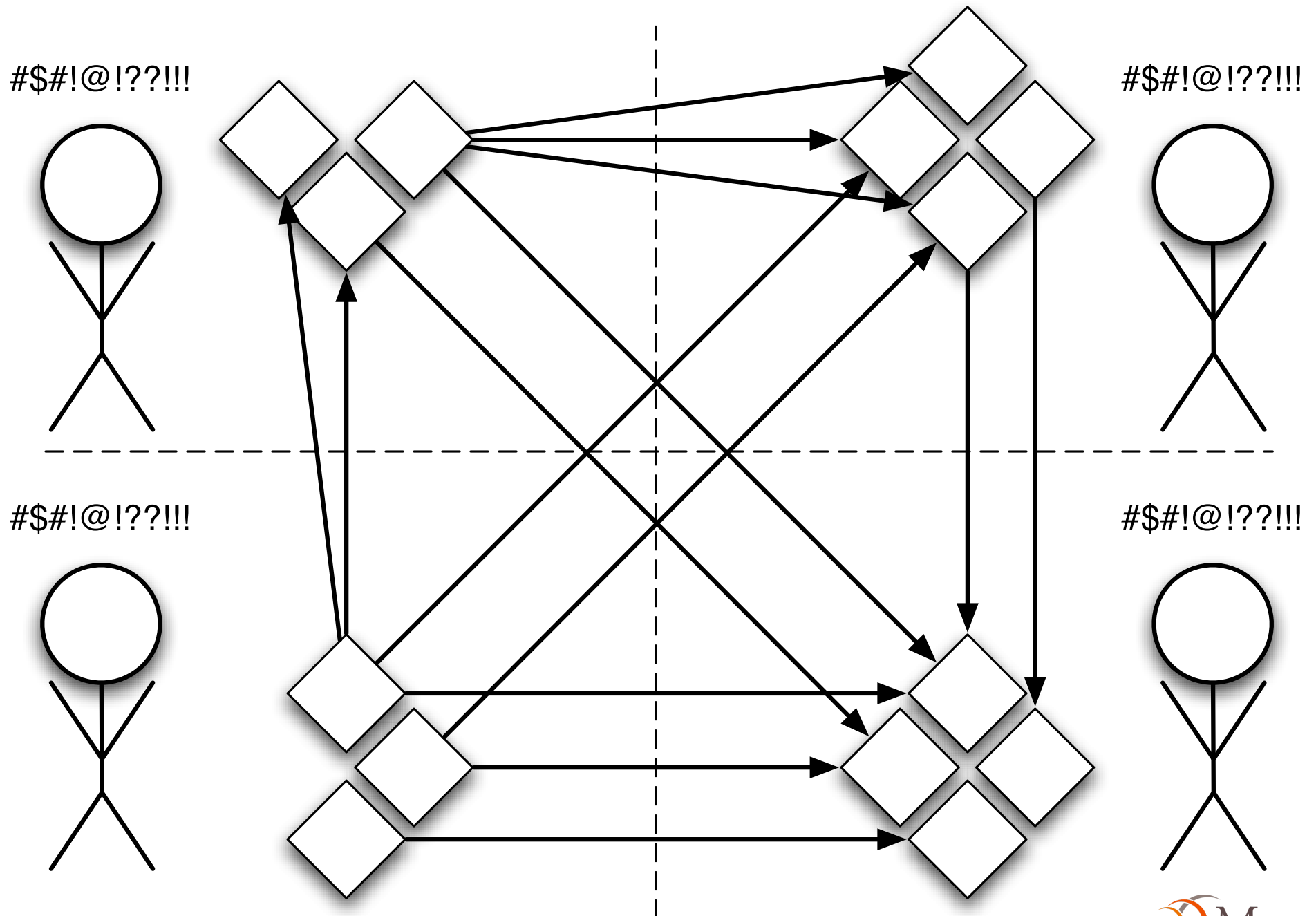
- Enhanced dependency support
 - Transitive dependencies
- Build Lifecycle
 - Default lifecycle which can be augmented
 - New lifecycle can be created when necessary
- Unified project file
 - Everything relating to a project is stored in one place
- Enhanced plugin support
 - Developing plugins in Java is easy
 - Developing plugins in scripting languages is also possible
- Improved site management
- Improved release management
- Archetypes

Maven Apache Subprojects

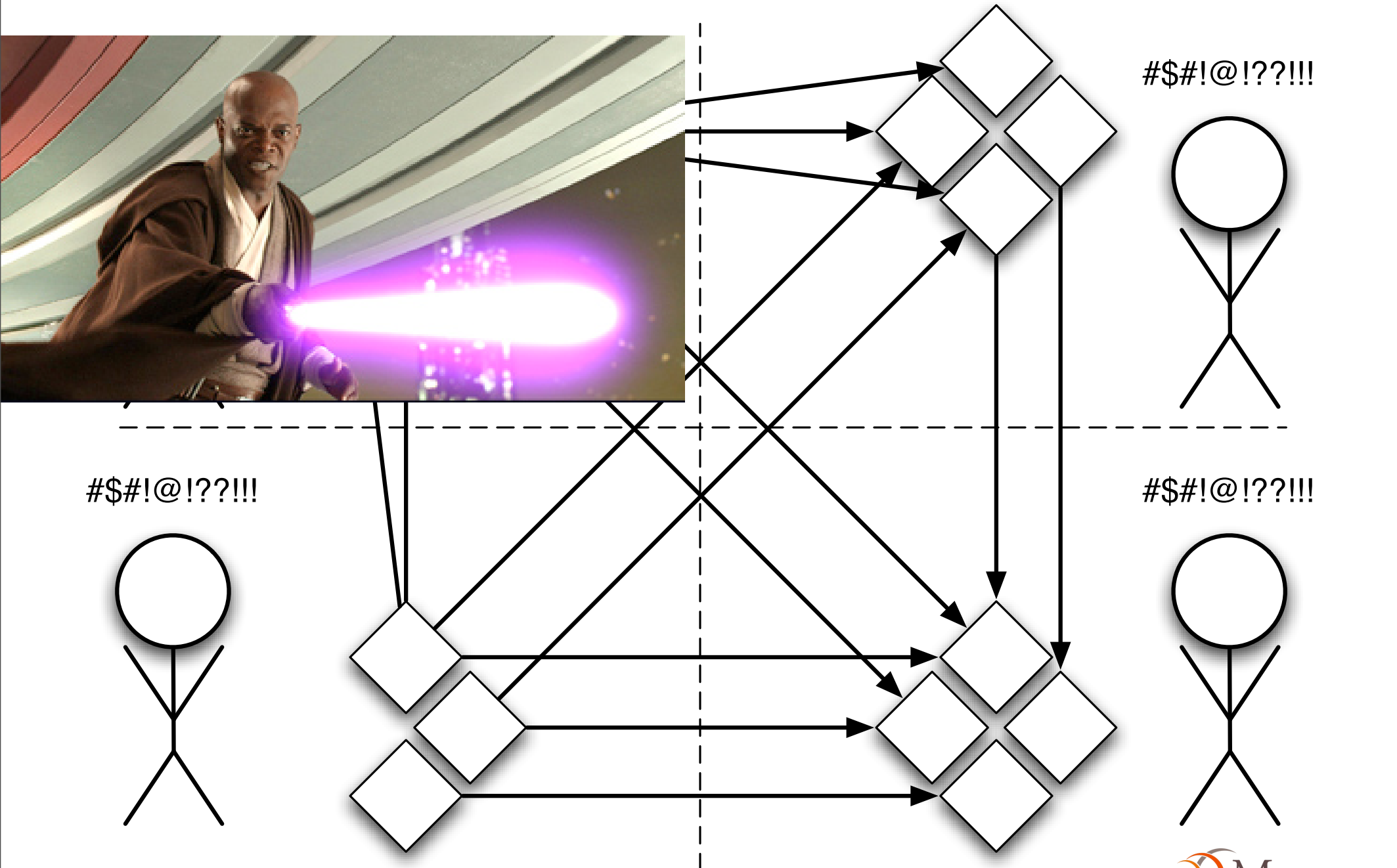
- MavenSCM
- Maven Wagon
- Continuum
- Repository Manager

The Issues Facing Teams

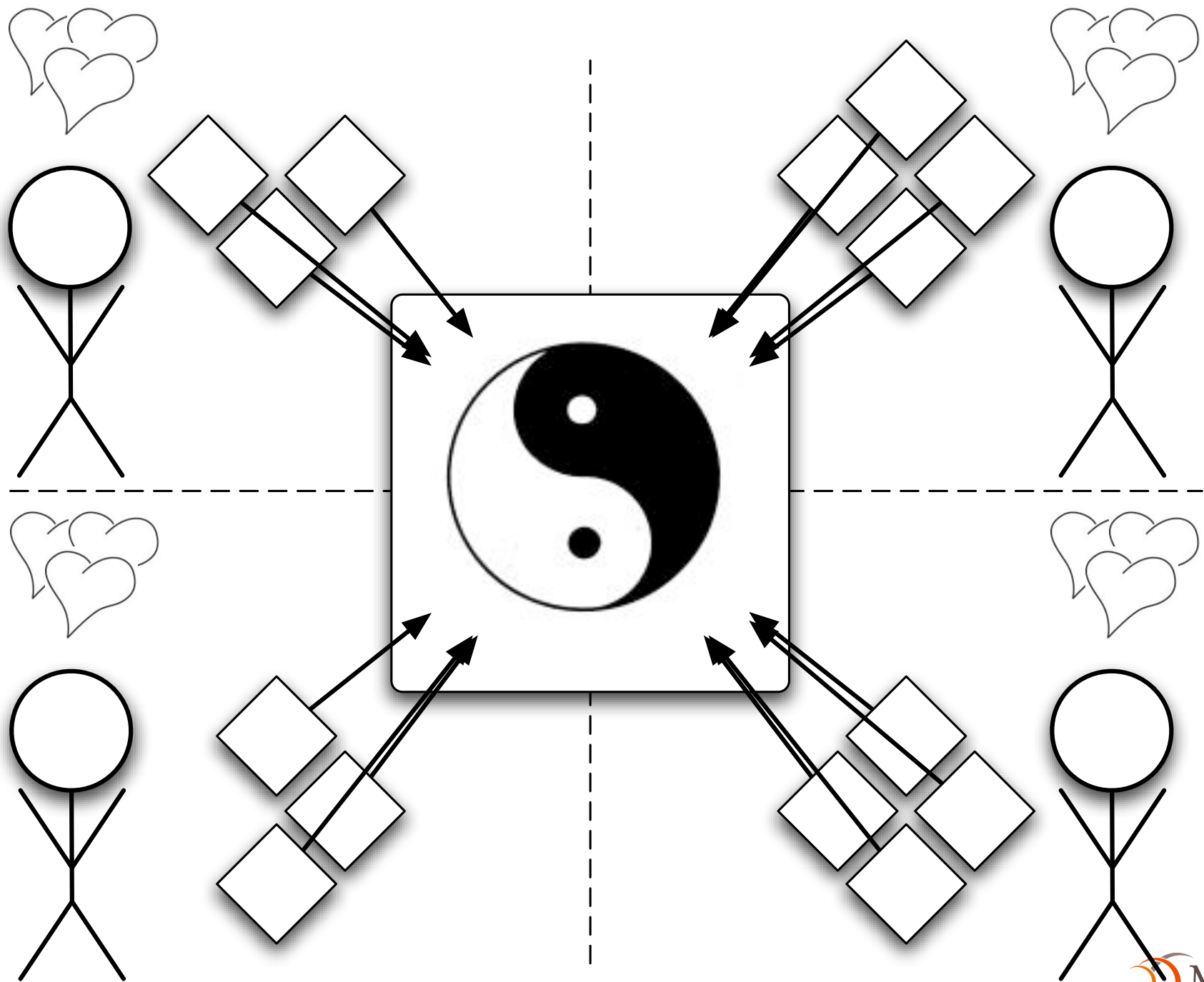
Working without Maven



Working without Maven



The Maven Way



The Maven Way

